AN INVESTIGATION OF TECHNOLOGY DESIGN FEATURES FOR SUPPORTING REAL-TIME COLLABORATIVE PROGRAMMING IN AN EDUCATIONAL ENVIRONMENT

A Thesis in
Information Sciences and Technology

by
Jin Zhang

© 2018 Jin Zhang

Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science

May 2018
The thesis of Jin Zhang was reviewed and approved* by the following:

Mary Beth Rosson  
Professor of Information Sciences & Technology  
Associate Dean, College of Information Sciences and Technology  
Thesis Advisor  
Head of Graduate Program

Benjamin Hanrahan  
Assistant Professor of Information Sciences & Technology

Rosalie Ocker  
Associate Professor of Information Sciences & Technology

*Signatures are on file in the Graduate School.
ABSTRACT

Courses that introduce concepts and skills for object-oriented programming (OOP) are very common across educational establishments, even in middle and high schools. In many such courses, collaboration in small teams on OOP projects is a necessary and important element.

Team programming allows educators to assign more challenging problems, while at the same time providing practice on programming collaboration that is common in industry. In these projects, real-time (synchronous) collaboration is often critical, as it gives students the opportunity to view, build and check on each other’s code as it is being developed. However, despite the value in such real-time collaboration, it is currently only partly supported by tools, with students often relying on independent tools for different aspects of their work (e.g., code editors, compilers, online chat). There are no technologies that would be most appropriate for students to use while completing OOP project collaboratively in real-time. Moreover, educators also use a mix of methods when grading OOP assignments. There is no effort to analyze technologies that would be most appropriate for their grading method.

My thesis studies this gap in tool support using two primary methods: 1) working from a scenario-based analysis of a state-of-the-art shared programming tool (Cloud9), I designed and mocked-up enhancements to Cloud9, so I could explore how it might support common team programming scenarios; and 2) I conducted interviews of OOP students and instructors to learn about their background and expectations for team projects, as well as react to my design ideas.

In this thesis, I report my methods for design analysis and empirical study, and interpret my results to propose new design directions for tools that might better support real-time collaborative programming in an educational environment.
# Table of Contents

LIST OF FIGURES ......................................................................................................................... vi

LIST OF TABLES .............................................................................................................................. vii

ACKNOWLEDGEMENTS ................................................................................................................... viii

Chapter 1 ........................................................................................................................................ 1
  Introduction ................................................................................................................................. 1
  Contributions ............................................................................................................................... 4
  Structure of Thesis ....................................................................................................................... 4

Chapter 2 ........................................................................................................................................ 5
  Literature Review .......................................................................................................................... 5
    Synchronous Collaborative Programming ................................................................................ 5
    Real-Time Collaborative Editing ............................................................................................... 6
    Collaborative Learning ............................................................................................................. 7
    Pair Programming .................................................................................................................... 8
    Computer Mediated Communication ......................................................................................... 9
    Feature Analysis of Existing Tools for Synchronous Programming ......................................... 11
    Summary and Research Question ............................................................................................. 14

Chapter 3 ........................................................................................................................................ 15
  Research Methodology ................................................................................................................ 15
    Chapter Overview ..................................................................................................................... 15
    Participants and Recruitment .................................................................................................... 16
    Data Collection .......................................................................................................................... 19
    Data Analysis ............................................................................................................................. 31

Chapter 4 ........................................................................................................................................ 32
  Results and Findings .................................................................................................................... 32
    Chapter Overview ..................................................................................................................... 32
    Student Experiences in Learning OOP ..................................................................................... 32
    Student Reactions to the Design Mock-up ............................................................................... 43
    Instructor Experiences with Teaching OOP ............................................................................. 57
    Instructor Reactions to the Tool Mock-up ............................................................................... 62
    Summary ................................................................................................................................. 66

Chapter 5 ........................................................................................................................................ 68
  Discussion .................................................................................................................................... 68
    Chapter Overview ..................................................................................................................... 68
  Limitations .................................................................................................................................. 75

Chapter 6 ........................................................................................................................................ 77
  Conclusions ................................................................................................................................. 77
  Future Work ................................................................................................................................. 77
References .......................................................................................................................... 79
Appendix A ................................................................................................................................ 84
Summary Explanation of Exempt Research ........................................................................... 84
Consent for Exempt Research ............................................................................................... 84
Appendix B ............................................................................................................................ 85
Part I Student Interview Questions .................................................................................... 85
Appendix C ............................................................................................................................ 86
Part II Student Interview ..................................................................................................... 86
Appendix D ............................................................................................................................ 93
Part I Instructor Interview Questions ................................................................................. 93
Appendix E ............................................................................................................................ 94
Part II Instructor Interview ................................................................................................. 94
Appendix F ............................................................................................................................ 99
Instructor I Interview Questions and Responses ............................................................... 99
Appendix G ........................................................................................................................... 104
Instructor II Interview Questions and Responses ............................................................ 104
LIST OF FIGURES

Figure 1. Scenario-based design framework ................................................................. 21

Figure 2. Framework for the Interview Sessions .......................................................... 29

Figure 3. Create/New function ..................................................................................... 72

Figure 4. Function ........................................................................................................ 72

Figure 5. Test ............................................................................................................... 73
LIST OF TABLES

Table 1. Development tools supporting real-time programming........................................ 12
Table 2. Demographic of the participants.............................................................................. 17
Table 3. Student participants’ general information and programming background...................... 18
Table 4. Instructor participants’ general information and teaching background......................... 19
Table 5. Summary of each scenario’s design features and subtask concept................................. 23
Table 6. Task goal and design feature in each screen (student)................................................ 24
Table 7. Task goal and design feature in each screen (instructor)............................................. 26
ACKNOWLEDGEMENTS

First of all, I would like to give special thanks to my advisor, Professor Mary Beth Rosson, for her encouragements and expert advice. She has always been exceptionally supportive from the day I joined her research lab as an undergraduate student. Despite the fact that she has extremely busy schedules, she always finds time and patiently guides me through my graduate studies. Next, I would like to thank my committee members, Ben Hanrahan and Rosalie Ocker, for agreeing to partake in my thesis defense. They also have provided me with valuable advice that helped my thesis along.

I would also like to thank my friends from the College of Information Science and Technology for their assistance and encouragement. Lastly, I would like to thank my family members, Ruiqi Zhang, Lijia Yu, Yuan Zhang for loving me and giving me supports. Thank you and I love you all.
Chapter 1

Introduction

As computer and information science (CIS) degrees have grown in popularity, virtually all institutions of higher education offer courses in object-oriented programming (OOP) that introduce and provide practice on this foundational software development paradigm that is prevalent in both science and industry. Often in these courses a student will be expected to interact with one or more peers to solve a programming task. Many educators assign these team projects to challenge students with more complex problems that are only achievable with group effort, while at the same time providing practice with collaborative programming that is so common in the CIS workplace.

Pedagogically, the process of team programming is intended to support collaborative learning. As Balasundaram and colleagues state, “Collaborative learning is a kind of pedagogical model where students learn concepts and topics through interactions and discussions” (Balasundaram et al., 2006, p. 1). Learning in a collaborative setting therefore can become significant factor in students’ learning achievement because it enables active learning (Suwantarathip & Wichadee, 2014). An active learning environment is considered to be important because it can inspire students to be more engaged in solving problems, sharing ideas, giving feedback, and teaching each other (Dori & Belcher, 2005; Johnson, Johnson, & Smith, 1998). According to Gokhale (1995), actively exchanging ideas within small groups can bring significant impact to learning such as promoting critical thinking skills. In fact, some educators
have argued that in active learning situations, the word ‘collaborative’ is more important than the word ‘learning’ (Dillenbourg, 1999).

Nowadays, there is an accelerating trend for remote interaction, for example because of telework, distance education, or other forms of globally distributed organizations that make collaborative learning and subsequent interaction necessary. With respect to educational settings, Collaborative learning is known to provide many benefits to learning: it develops social interaction skill; helps to develop learning communities; builds more positive heterogeneous relationships; encourages diversity understanding; develops critical thinking skill/higher level thinking skill; and promotes communication skill, etc. (Laal & Ghodsi, 2012). However, the actual learning experiences and outcomes of collaborative activities can be varied due to the quality of the shared learning process. In the case of collaborative programming, this implies a critical need to maximize the efficiency and effectiveness of collaborative programming itself, as this will increase the quality of students’ learning processes and the quality of their shared understandings (Borge, 2015). Unfortunately, despite the importance for supporting collaboration in programming (whether for learning or professional activities), team programming is currently only partly supported by tools, with programmers usually recruiting and integrating a wide array of independent tools for different aspects of their work.

Given the state-of-art options in tools support, team programmers in OOP courses tend to collaborate using one of two strategies: one is to work side by side (i.e., in-person) on a single copy of the code as it is designed and developed, using a single integrated development environment (IDE); another is to work in parallel on separate copies, possibly on different portions of the code that is later integrated. Neither approach is ideal for tightly coupled
synchronous collaboration, where all programmers in the same team actively contribute in real-time to the same programming project.

The first strategy only allows one programmer (whoever is currently controlling the keyboard and mouse, sometimes called “the driver”) to scroll around within and generate or edit the code; the collaborating team members are left to watch the process and make suggestions until it is their turn to “take the wheel”. One likely consequence is that the stronger programmers will write the code, with the unfortunate result that the weaker programmers gain less from the process. The second strategy requires the programmers to manually upload and download code changes; this process must be constantly monitored to avoid conflicts (Goldman, et al. 2011). My thesis project was grounded in the clear implication that a synchronous collaborative programming tool could be of tremendous help in these team programming projects. If well designed, such a tool could help programmers to avoid the obstacles and inconveniences of making shared edits, while also supporting more directly the benefits of collaborative learning as they work on shared goals.

My research is exploratory in nature. I am seeking to understand the collaboration process for OOP team projects as a whole, but with specific focus on tools and design features that can better support real-time collaboration. I am also interested in instructors’ opinions of these learning processes and supporting tools, to understand whether and how they could use them to assist students in learning OOP. My research will contribute to a better understanding of team OOP projects; it also provides an analysis and proposed enhancement of a state-of-the-art shared programming tool (Cloud9). Using interviews with students and instructors, I offer a better understanding of how students and instructors would use such tools, and suggestions for how to create a more appropriate tool for real-time collaborative programming.
Contributions

The contributions of this thesis are as follows. First, I hope to advance the existing research knowledge base in regards to real-time collaborative programming, aiming primarily at an educational setting for such projects. Second, I hope to offer new design directions that make some contributions toward future tool design for the support of real-time collaborative programming. The third contribution is to document current pedagogical goals for teaching and learning team programming in OOP.

Structure of Thesis

This thesis consists of five chapters, structured as follows: The first chapter examines my research interests and introduces the research motivation. The second chapter provides a literature review on related topics. The third chapter presents a detailed discussion of the research methodology I have employed. The fourth chapter presents the findings from my empirical study. The fifth chapter discusses the implications of my findings and the limitations in this research study. Lastly, the sixth summarizes my conclusions and future work.
Chapter 2

Literature Review

Synchronous Collaborative Programming

Synchronous collaborative programming (hence abbreviated “synchronous programming”) is an essential activity whose importance has grown as the internet has enabled pervasive connectivity and the virtues of teamwork have become better understood (Benveniste & Berry, 1991). Synchronous programming implies that team members in distributed geographical locations are working on the same program or software at the same time and toward the same goal (Bravo et al., 2013). It can be used in industrial settings to improve the efficiency of team projects, as well as in academia to enhance students’ learning experiences.

Software developers who are collaborating synchronously should be provided a development environment designed specifically for their collaboration, which is generally not the same one designed for solo programmers (Goldman, et al. 2011). Boyer et al. (2008) pointed out that enabling students to work in programming assignments collaboratively in different locations can provide significant benefits compared to working individually. Synchronous programming tools can provide geographically distributed students the environment to work concurrently in the same programming task, and it also allows the students to design, code, debug, test, and document collaboratively (Shen & Sun, 2000).

Despite the fact that collaborative learning in the classroom has been shown to be greatly beneficial for students studying computer and information sciences, the existing barriers
in today’s academic technology infrastructure often interfere with successful implementation of collaborative learning methods (Boyer et al., 2008). Fortunately, modern technology offers considerable promise as a solution path, at least partly because pervasive network connectivity and a growing number of real-time collaborative tools are generally available to all internet users (e.g., the pioneering and popular tool suite of Google.com). The time is ripe for the application of these internet technologies to team programming projects, as many CIS students and researchers have spoken of the benefits that such tools would bring to CIS learning activities (Goldman et al., 2011).

**Real-Time Collaborative Editing**

Real-time collaborative editing enables two or more users who may be geographically separated to work together at the same time via a computerized environment (Greenberg & Marwood, 1994). Another potential setting could be a sort of “programming studio” where a team sits together in a shared space but works in a distributed fashion using separate networked computers. The key supporting feature is a real-time shared editing tool that allows multiple users to access and simultaneously edit a document including text, picture, drawing, or any other content relevant to the shared task (Minor & Magnusson, 1993). As argued in the classic analysis of Ellis et al. (1991), collaborative editing brings many benefits to the users including: it can increase information access; encouraging parallel work within a group; it can be efficient and can help prevent information loss, leading to a tangible group produce; and also can make learning a natural aspect of tool use (Ellis et al., 1991).

In a collaborative learning environment, students are reported to have higher levels of learning performance and have been shown to make higher quality decisions; to complete higher quality reports; to participate in the learning process more equally; and to engage in discussions
that are more complex and challenging than when working alone (Ellis et al., 1991). However, Ellis et al. (1991) also mentioned that collaborative editing can bring downsides such as making discussion and group focus more difficult; and cutting down on social interaction (Ellis et al., 1991).

**Collaborative Learning**

The term *collaborative learning* has been defined in varying ways over the years; and has been recruited as a concept within many fields of study. This makes the concept difficult to delineate in a distinctive way that is tied to specific purposes, goals, and perspectives (Kreijns et al., 2003). Taking a macro perspective, collaboration can take place simply through conversation or other interactions among a group’s members with varying impacts on learning: for example, negotiation has stronger collaborative consequences for learning than the relatively more unidirectional giving of one peer giving instructions to another (Dillenbourg, 1999). Roschelle and Teasley (1995) offer a widely used definition of collaboration: “*shared understanding through interaction with others, where the participants are committed to or engaged in shared goals and problem solving*” (Roschelle and Teasley, 1995, p.77).

Borge (2016, p.324) offers a more elaborate definition for collaborative learning that considers in more detail what the shared learning processes might entail: “*Collaboration refers to the synchronous activity that occurs as individuals engage in collective thought processes to synthesize and negotiate collective information in order to create shared meaning, make joint decisions, and create new knowledge*” (see also Borge et al., 2012; Hakkarainin et al, 2013; Stahl, 2006). Generally, students and instructors are involved in a collaborative learning environment as they obtain and exchange knowledge and experience (Suwantarathip & Wichadee, 2014). In collaboration, students can enhance their own understanding by
explaining concepts to less advanced students. They can also benefit themselves by receiving help from other more advanced students (Carroll et al., 2015). Banerjee suggests that students engaged in collaborative learning come to feel as if they ‘own’ the material, because each student explains his or her understanding and shares thoughts with all teammates (Banerjee, 2000). Collaborative learning has been defined to involve three specific criteria: interactivity, synchronicity and negotiability (Dillenbourg, 1999). This leads to a view of OOP student teams as a small project group (Dillenbourg et al., 1996; Rogoff., 1995; Stahl., 2006) that operates through its interactions and negotiations as a knowledge-creating community, in the shared space afforded by a synchronous programming tool.

**Pair Programming**

*Pair programming* is a style of programming where two programmers work side-by-side at one computer throughout the process of completing the code collaboratively (Williams et al., 2000). This style of work has become popular in CIS education, particularly in introductory courses (Mcdoell et al., 2003). Students have reported that pair programming gives them a glimpse into a type of collaboration that they may experience in the workplace (e.g., advice from a peer or mentor); and that the perspective of a partner can help broaden one’s knowledge (Rodríguez, et al. 2017).

In pair programming, one person is called the “driver” and the other is the “navigator”. The driver types at the computer, while the navigator observes the work of the driver, makes suggestions, and monitors for defects such as logical errors or typos. Previous research has indicated that pair programming produces better results than individual programming when the pairs are physically collocated (Baheti, et al. 2002). Because my research is aimed at educational settings, I also considered the benefits that pair programming might bring to students and
instructors. A study of pair programming in an introductory course setting found that the pairing bolstered course passing rates and contributed to a greater persistence in CIS majors. Moreover, students who were previously paired were more likely to pass the subsequent programming courses, which strongly indicates that students have succeeded in the paired without taking a ‘free ride’; that is, they truly learned programming skills (McDowell, et al. 2006).

Computer Mediated Communication

*Computer mediated communication* (CMC) was introduced with the first electronic digital computer in the early 1960s. It refers to the process of human communication that takes place through (mediated by) digital devices/systems (Thurlow et al., 2004). The integration of computer and communication have created new forms of group work that rely on CMC (Baltes et al., 2002). For example, on educational environment, it has become common for university professors to orchestrate Internet-based discussion forums that leverage general access to the internet and associated tools (Guzdial & Turns, 2000). When collaborators are not co-located and must rely entirely on CMC, they often miss out on a great deal of informal communication, including nonverbal cues like attention, facial expression, minor comments like “hmm”, and so on. This implies that the team members must do extra work to maintain common ground and conversation coherence (Clark, 1996; Convertino et al., 2008). These concerns have led to considerable research in the area of “media richness”, namely investigations of what sorts of channels (face-to-face, video, audio, simple text, virtual reality, etc.) provide the best support for collaboration at a distance (Dennis & Kinney, 1998).

At the same time, CMC may provide important benefits that are not available in face-to-face conversations, as argued in the classic paper “Beyond Being There” (Hollan & Stornetta, 1992). Some of these features include: as a side effect of computer-mediated conversation, there
can be a digital record of the communication, which means you can operate on the communication record in various ways (e.g. search, classify, visualize); the computer mediation increases the perceived “social distance” between collaborators (how connected you feel to each other), which can reduce the effect of social stereotypes or social inhibition (perhaps helping “less powerful” group members contribute more). Another technology-related implication of CMC is that the communication aspects of a shared task can be integrated directly in other task-supporting tools. This final benefit is a feature of great interest in my research because of its obvious implications for synchronous programming (as in the chat and commenting tools that are provided by Google Docs).

Hollan and Stornetta (1992) propose that some task-based interactions require immediate, synchronous feedback and that these settings need special attention by CMC designers. Examples of interactions that seem to depend greatly on synchronous feedback include real-time decision making (where collaborators rely on a variety of cues to evaluate each other’s positions), brainstorming, community building (where feelings of closeness are essential). Finally, in an educational setting, researchers have suggested that virtual office hours held by instructors would benefit greatly from immediate feedback, presumably because students having problems may not be able to express these in precise or articulate language (Branon & Essex, 2001).

For the context of my project, these findings and arguments emphasize that some form of real-time communication may be critical when team programmers are encountering a particularly difficult or challenging programming problem or negotiating an approach. At the same time, it raises a number of interesting options for integrating the “digital record” of the shared work (i.e.
the code itself) with online discussion and recommendations about the work, both as it is in progress and as a historical record for later review.

In addition, researchers have also recognized some downsides of CMC, including how technology can interrupt concentration. Many studies have indicated that in a collaboration environment, users often experience interruptions during the activity process (O'Conaill & Frohlich, 1995). It usually takes about 25 minutes for users to resume the primary task after interruption (González & Mark, 2005), and it does impact the quality of the work (Kalgotra et al., 2016). Frequent interruptions can actually bring significant impact on the increase of exhaustion, annoyance and anxiety (Lin et al., 2013). Therefore, when to interrupt users have become a significant concern in the field of HCI and CSCL.

**Feature Analysis of Existing Tools for Synchronous Programming**

Before entering in to design thinking about how best to support synchronous programming, I first examined the strengths and weakness of existing tools. I identified and reviewed several existing code editing tools or integrated development environments (IDE) that have been designed and developed for commercial uses; I examined the features they offer in support of CMC and collaborative code development. Looking across these tools, several classes of features became salient. Specifically, I identified seven collaboration-related features that are illustrated across these tools and seem important in the design of a real-time collaborative programming that supports team communication, code development, and real-time (synchronous) collaborative programming. The six tools I analyzed were: Coderpad, Collabedit, Etherpad, Codebunk, Codeshare, and Cloud9. The seven features were video chat, text chat, syntax highlighting, history, integrated compiler, real-time editing and more generally the
provision of an OOP IDE. In the following, I give a brief introduction to each tool and summarize their coverage of these seven features (Table 1).

Coderpad (coderpad.io) allows users to code collaboratively in real time. Specifically, a user may open a video chat with his/her team member(s). This video can then be saved as a playback for any user to review for future reference. This tool also provides syntax highlighting that guides successful completion of programming language “sentences”. Users are also able to run and test the code after they have completed the editing process. There is no chat box in this tool, so people cannot be able to send each other text messages for communication.

Table 1. Development tools supporting real-time programming

<table>
<thead>
<tr>
<th>Development Tools</th>
<th>Video Chat</th>
<th>Chat Box</th>
<th>Syntax Highlighting</th>
<th>History</th>
<th>Compiler</th>
<th>Real-time Editing</th>
<th>Project Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coderpad</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Collabedit</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Etherpad</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Codebunk</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Codeshare</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Cloud 9</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Collabedit (collabedit.com) enables features such as real-time editing, history checking, real-time editing and text chatting. However, the interface is poorly designed, and it does not offer syntax highlighting; nor does it offer a compiler as its focuses is on editing only. These two functionalities are considerably important for programmers, especially for programming beginners like college students.
Etherpad (etherpad.org) allows users to edit in real-time. It also allows users to communicate via a chatbox. But it does not have the other functionalities such as video chatting, history checking, syntax highlighting, and most importantly, it does not have a compiler.

I considered Codebunk (codebunk.com) to be a more useful tool than the ones described thus far. It provides all the functionalities such as video chatting, history checking, syntax highlighting; it also includes a chat box and a compiler. Nonetheless, it does not have the function of synchronous editing that can support two or more users working together at the same time on the same body of code.

CodeShare (codeshare.io) enables real-time code editing and video chatting. However, it does not include a IDE. It also does not offer syntax highlighting and does not include a compiler. It is more like a Google Doc for working on shared text that happens to be code, rather than a real-time collaborative programming tool.

All the tools I have mentioned above have a range of limitations and restrictions. First of all, they have mainly been designed and used to conduct remote interviews as part of an industrial candidate review and hiring process; that is, they have been specialized for the use case of observing one or more candidates write or edit code. Secondly, they are all limited to development of a single class of code, which does not support real-time coding in an actual OOP environment that includes with multiple classes and typically entire libraries of reusable code.

After considerable further research, I found the Cloud9 tool (aws.amazon.com/cloud9/), which supports all the main functionalities except video chatting. It allows the team members to gain access to other classes and files within the same platform (e.g., a library that may have been pre-loaded by an instructor). Users who of Cloud9 are able to work with their teammates collaboratively on OOP projects, which offers an online integrated development environment.
Cloud9 supports hundreds of programming languages such as C, C++, Python, Java, Ruby, JavaScript, and more. Cloud9 allows users and their teammates to edit code together in real time (Cloud9 IDE). Cloud9 offers all of the features that I expect in a synchronous programming tool, except for video chat, which is debatable as an essential element and may have been provided primarily for the industry interview use case. As a result, Cloud9 became the starting point for my design explorations.

**Summary and Research Question**

A common thread drawn from my literature review is that real-time collaborative editing and CMC should enhance the learning expected in team OOP projects. As the number of students who are learning to program grows in size, attention must be given to supporting their team collaboration process on programming projects. After also conducting an analysis of existing collaborative programming tools, I am left with this question:

*RQ*: What features should appear in a tool designed to support small group, real-time collaborative programming in an educational setting?

To address this research question, I have conducted interviews using scenario-based design prototypes with students and instructors to investigate their current practices and explore my own design ideas. I used the findings retrieved from the interviews to make recommendations toward a feasible real-time collaborative programming tool.
Chapter 3

Research Methodology

Chapter Overview

Research methodologies are necessary to gain knowledge about a research problem (McGrath, 1982). My research was conducted using the methods of qualitative study and interaction design. In CIS fields, qualitative methods allow researchers to understand user needs and behaviors, and especially to evaluate situated use of technology. It also allows researchers to ask questions and develop more complexed, unexpected, and interesting answers (Blandford, et al. 2016). Because my research is aiming to discover ideas for technology design, and is looking for open-minded answers, there is no hypothesis testing included. The intent for this research instead is to explore the phenomenon of real-time collaboration in programming, and also to uncover the current pedagogical goals and related practices of both students and instructors, reflections on personal experiences relating to team programming projects, as well as challenges and opportunities for improved tool support.

There are four main sections in this chapter. In the first section, I describe how I recruited study participants. In the second section, I summarize how I designed an interview study following the guidance of scenario-based design (SBD). In the third section, I elaborate on the data collection process, in particular describing on how I conducted the interviews and gathered meaningful information. In the final section, I explain how I analyzed the participant comments collected during the interviews.
Participants and Recruitment

My participants were recruited from the College of Information Sciences and Technology (IST). The participants for this study included sixteen students and three instructors; in a semi-structured interview study it is typical to interact with a relatively modest number of individuals but to approach each interview with the goal of in-depth understanding. My intention was that all the student participants should have at least novice level understanding of OOP; all the instructors should have OOP teaching experience. As described below, I used a mix of convenience sampling and snowball sampling for student recruitment, and a more direct personal identification and request for instructor selection (i.e., sampling from a relatively small pool).

First, I sent students who are from the College of IST emails with the study goals and process included, asking whether they would like to participate in the study. Perhaps due to my own long-time participation in this population (and established personal network), a number of students replied quickly and I was able to quickly schedule ten students’ interviews (convenience sample). However, I still needed six more students to participate in this study. To accomplish this, I asked students who joined me for interviews to introduce me to their friends who know OOP (snowball). In this way, I was able to meet my goal of sixteen student participants.

The process of recruiting instructors was a bit more difficult, generally because they have tight schedules. The method I used for recruiting instructors was also convenience sampling. I sent emails, including the research goals and interview process to instructors from whom I had taken previous OOP classes, asking whether they are interested in this study and will be available for an interview. I sent emails to three instructors and one teaching assistant. All replied saying they are interested in the study, however, one of the instructors was not available due to scheduling conflicts. Therefore, I conducted an interview with two instructors and one teaching
assistant. There were no incentives included in this study, with all participants participating voluntarily.

Table 2 summarizes several demographic features of my participant pool. In Table 3, I offer a more specific record of each student participant, including their gender, discipline of study, years of school, description of programming skills/level, and courses they have completed. I also introduce the participant abbreviations (S1-S16) that I use when giving example quotes from interviews. In Table 4, I have put together a similar view of the instructors and teaching assistants including gender, discipline, and teaching experience.

Table 2. Demographic of the participants

<table>
<thead>
<tr>
<th>Demographic Data</th>
<th>Number of Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>15</td>
</tr>
<tr>
<td>Female</td>
<td>4</td>
</tr>
<tr>
<td>Undergraduate</td>
<td>14</td>
</tr>
<tr>
<td>Graduate</td>
<td>2</td>
</tr>
<tr>
<td>Instructor</td>
<td>3</td>
</tr>
<tr>
<td>American</td>
<td>7</td>
</tr>
<tr>
<td>Asian</td>
<td>12</td>
</tr>
</tbody>
</table>
Table 3. Student participants’ general information and programming background

<table>
<thead>
<tr>
<th>Participant ID &amp; Gender</th>
<th>Discipline &amp; Gender</th>
<th>Description</th>
<th>Completed Courses</th>
<th>Programming Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1: M IST &amp; Math</td>
<td>3 years of programming; Recent graduate student (class of 2017)</td>
<td>IST140, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S2: M IST</td>
<td>2 years of programming; Senior undergraduate student</td>
<td>CS105, IST140, IST220, IST240, IST242, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S3: M IST</td>
<td>8 years of programming; Second-year PhD</td>
<td>IST597 (AI, Deep learning)</td>
<td>Professional</td>
<td></td>
</tr>
<tr>
<td>S4: M IST &amp; SRA</td>
<td>3 years of programming; First-year master (class of 2017)</td>
<td>CS121, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S5: F IST &amp; SRA</td>
<td>4 years of programming; Recent graduate student (class of 2017)</td>
<td>IST140, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S6: F IST &amp; SRA</td>
<td>5 years of programming; Recent graduate student (class of 2017)</td>
<td>CS121, CS122, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S7: M IST</td>
<td>3 years of programming Recent graduate student (class of 2017)</td>
<td>IST140, IST242, IST311</td>
<td>Novice</td>
<td></td>
</tr>
<tr>
<td>S8: F IST &amp; Economics</td>
<td>4 years of programming; Recent graduate student (class of 2017)</td>
<td>CS101, IST240, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S9: M IST</td>
<td>2 years of programming; Senior undergraduate student</td>
<td>IST140, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S10: F IST &amp; Business</td>
<td>2 years of programming; IUG student</td>
<td>IST140, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S11: M IST</td>
<td>4 years of programming; Recent graduate student (class of 2017)</td>
<td>IST140, IST242, IST261, IST311, IST411</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S12: M IST &amp; SRA</td>
<td>4 years of programming; Senior undergraduate student</td>
<td>CS121, IST242, IST261, IST311, IST411</td>
<td>Professional</td>
<td></td>
</tr>
<tr>
<td>S13: M IST</td>
<td>4 years of programming; Senior undergraduate student</td>
<td>IST140, IST242, IST261, IST311</td>
<td>Professional</td>
<td></td>
</tr>
<tr>
<td>S14: M IST</td>
<td>10 years of programming; Fourth-year PhD</td>
<td>IST501</td>
<td>Professional</td>
<td></td>
</tr>
<tr>
<td>S15</td>
<td>4 years of programming; IST Alumni (class of 2015)</td>
<td>CS101, IST240, IST242</td>
<td>Intermediate</td>
<td></td>
</tr>
<tr>
<td>S16</td>
<td>3 years of programming; Senior undergraduate student</td>
<td>IST140, IST242, IST261, IST311, IST411, IST597</td>
<td>Intermediate</td>
<td></td>
</tr>
</tbody>
</table>
Table 4. Instructor participants’ general information and teaching background

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>Discipline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1: M</td>
<td>IST</td>
<td>Assistant teaching professor; Four years of teaching programming experience.</td>
</tr>
<tr>
<td>I2: M</td>
<td>IST</td>
<td>Assistant professor; More than six years of teaching programming experience.</td>
</tr>
<tr>
<td>I3: M</td>
<td>IST</td>
<td>Teaching assistant; two-year TA experience of OOP</td>
</tr>
</tbody>
</table>

**Data Collection**

In this research study, I used analytic methods to assess important features of existing programming tools and design enhancements to the Cloud9 IDE, and qualitative methods to collect and analyze my empirical data. The use of qualitative methods usually generates a large amount of data (e.g., interviews, videos or field notes), and allows researcher to explore a particular research phenomenon that may help to clarify a less-well-understood problem (Sutton & Austin, 2015).

First, working from scenario-based analysis, I considered likely usage scenarios for using a the Cloud9 programming tool in a team programming project, relying greatly on my own extensive experience; I have taken 6 programming courses as an IST/SRA major and worked on 4 team programming projects using Java during this time. This analysis helped to identify some of the limitations of Cloud9, which I addressed using novel technology concepts drawn from my literature review of synchronous editing and computer-mediated communication.

Next, I created a script to conduct semi-structured interviews, through which I shared the resulting design ideas with the participants so that I could collect their reactions and suggestions. I used interviews in this study because interviews are well suited for understanding participants’
perceptions and experience with technology (Blandford, et al. 2016). In the following, I will first introduce the SBD methods I used, specifically how I used this framework to design the novel features for the design prototypes. After that, I describe my interview process.

**Scenario-based Design**

In this study, scenario-based design (SBD) was used to explore the potential usages of an existing programming tool (Cloud9) and envision new features that might enhance this tool for synchronous programming tasks. These scenarios were then used to probe participants’ reaction to a “doctored” version of the tool with the new features (Blandford, et al. 2016).

SBD is a set of techniques in which designers envision the future use of a system that is still in its early design process (Carroll, 2000; Rosson & Carroll, 2009). SBD is also a framework for giving users more insight into the implications of proposed design features (i.e., it situates the new ideas in familiar task settings and user experiences; Blandford, et al. 2016). Figure 1 shows an overview of the SBD framework. This approach allows researchers to quickly demonstrate problems and to quickly capture human activities and specific actions which correspond to certain design features (Carroll, 2000).
My study makes eclectic use of the SBD framework; as an exploratory study of programmers’ needs, I have been more concerned with the early phases of the framework and never intended to build and iteratively enhance an actual programming tool. Thus, to begin, I investigated the strengths and weaknesses of state-of-the-art programming tools (see “Feature Analysis of Exiting Tools for Synchronous Programming” in Chapter 2). After identifying Cloud9 as an appropriate starting point, I generated design ideas, also based on theories and empirical studies of computer-mediated tools that I read in conducting my literature review.
process. Lastly, I applied these ideas to create an “enhanced” design of a prototype tool, presented as mock-ups created from Cloud9 screenshots.

The scenarios were designed to follow a logical sequence that might be part of a team OOP project. For example, the scenario implies that a student does one step before moving on to subsequent steps, comprising a sort of sub-task analysis of a hypothetical team project. Each scenario was created to highlight at least one new feature within the enhanced tool.

After generating the scenarios, I created design prototypes, working from the Cloud9 tool and user interface. A design prototype is used to represent a group of elements derived from overlapping design cases that provides the basis of the start and continuation from a design. It brings all the requisite knowledge which are appropriate to the design together in one schema (Gero, 1990).

In this study, I am mainly interested in six features. In Table 5, I have complied a list of design features along with the subtask concepts I wanted to explore in each piece of the OOP scenario. Among the six features, four of them were already developed by Cloud9, including the Run Button, the Error Message, the Share, and the Chat Box. Based on the combination of literature review and my own experience of tool use, I added two more features, which are the Comment Box and the History Tracking. At each step in the scenario, my general goal was to get the students’ awareness, reaction, and opinions of what they saw on the mocked-up screen. I wanted to understand if they can infer the use of each feature without prompting from me, and if so what they think of the importance of having such features available. More details are embedded in Table 5, Table 6, and Table 7.
Table 5. Summary of each scenario’s design features and subtask concept.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Design Feature</th>
<th>Scenario subtask goal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Screen 1</strong></td>
<td>Run Button</td>
<td>The importance of being able to run some code to test it at any time</td>
</tr>
<tr>
<td><strong>Screen 2</strong></td>
<td>Error Message</td>
<td>Getting useful but also code-specific error messages</td>
</tr>
<tr>
<td><strong>Screen 3</strong></td>
<td>Share</td>
<td>Inviting team members to join the same programming environment</td>
</tr>
<tr>
<td><strong>Screen 4</strong></td>
<td>Chat Box</td>
<td>Text-based communication “on the side”</td>
</tr>
<tr>
<td><strong>Screen 5</strong></td>
<td>Comment Box</td>
<td>Text-based communication integrated into the body of the code</td>
</tr>
<tr>
<td><strong>Screen 6</strong></td>
<td>History</td>
<td>Record of who did what and when, e.g. as viewed by an offline team member</td>
</tr>
</tbody>
</table>

In Table 6, I have put together the design features and task goals that I have used in the student interviews. Under each task goal, I have included the student interview questions I have used for this study.
Table 6. Task goal and design feature in each screen (student)

<table>
<thead>
<tr>
<th>Task goal</th>
<th>Design feature in each screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Button</td>
<td></td>
</tr>
<tr>
<td>Question:</td>
<td></td>
</tr>
<tr>
<td>What can you tell from the</td>
<td></td>
</tr>
<tr>
<td>interface below?</td>
<td></td>
</tr>
<tr>
<td>What functions within this</td>
<td></td>
</tr>
<tr>
<td>interface you think are</td>
<td></td>
</tr>
<tr>
<td>the most important to you?</td>
<td></td>
</tr>
<tr>
<td>What functions are the</td>
<td></td>
</tr>
<tr>
<td>least important?</td>
<td></td>
</tr>
<tr>
<td>Error Box and Share</td>
<td></td>
</tr>
<tr>
<td>Question:</td>
<td></td>
</tr>
<tr>
<td>Now, imagine that you have</td>
<td></td>
</tr>
<tr>
<td>clicked on the Run button</td>
<td></td>
</tr>
<tr>
<td>and you see an error</td>
<td></td>
</tr>
<tr>
<td>message at the bottom.</td>
<td></td>
</tr>
<tr>
<td>How important it is for</td>
<td></td>
</tr>
<tr>
<td>you and your teammates</td>
<td></td>
</tr>
<tr>
<td>to be able to see this</td>
<td></td>
</tr>
<tr>
<td>error message?</td>
<td></td>
</tr>
<tr>
<td>Now, imagine that you don’t</td>
<td></td>
</tr>
<tr>
<td>know how to solve this</td>
<td></td>
</tr>
<tr>
<td>problem and would like to</td>
<td></td>
</tr>
<tr>
<td>have a discussion with your</td>
<td></td>
</tr>
<tr>
<td>teammates. Based on this</td>
<td></td>
</tr>
<tr>
<td>interface and the given</td>
<td></td>
</tr>
<tr>
<td>functionalities.</td>
<td></td>
</tr>
</tbody>
</table>
What do you want to do next?

Share
Question: What does this screen suggest to you? Imagine you can see each other’s code in real-time. What will you want to do next? What functionalities can you use?

Chat Box
If you are offered with this chat box functionality, how likely will you use it? How important is it for you and your teammates to be able to have a conversation within this interface?

Comment Box
Now imagine Claude has helped you to determine the problem. What can you tell from the screen?
History

Now, imagine you are the only one online. You would like to see your teammates’ progress. What would you do based on the given functionalities?

In Table 7, I have put together the design features and task goals that I have used in the instructor interviews. Under each task goal, I have included the instructor interview questions I have used for this study. Unlike the student interviews, which are more focused on the use of each feature in detail, the instructor interviews are more focused on a higher level understanding of the tool use.

Table 7. Task goal and design feature in each screen (instructor)

<table>
<thead>
<tr>
<th>Task goal</th>
<th>Design feature in each screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>General reaction to the tool (Run, Error message, Chat box)</td>
<td></td>
</tr>
<tr>
<td>Question: What can you tell from the interface? Do you think these features are important for students to use?</td>
<td></td>
</tr>
</tbody>
</table>
General reaction to the tool (Comment box)

Question: What can you tell from the screen? Do you think the comment is important for students to have? Do you think this can help them in their learning process?

General reaction to the tool (History)

Question: This is an example of what you would see in the history. Would you use this feature to help the grading process? For example, to determine students’ contribution to the project.
Semi-Structured Interviews

I used one-on-one semi-structured interviews to create a rapport with each participant that would help to produce richer and more meaningful responses to my probes. Because my participants include both students and instructors, I designed two different scripts for the two types of participants, with appropriate variations according to educational role. Both interview scripts consisted of two parts containing different questions, detailed in the sections below.

All the interviews were audio recorded as iPhone audio files. The sessions lasted from 30 to 40 minutes for each interview, including both parts of the interviews. The interview sessions were guided by the step-based framework that is illustrated in Figure 3. Figure 3 provides a step-by-step description on how exactly I conducted the interviews. I have included the Part I interview scripts in Appendix B (Student) and Appendix D (Instructor). The prototypes presented during the interviews are included in Appendix C (Student) and Appendix E (Instructor).

Note that in order to create as rich a channel as possible during the interviews, some of the discussions with students were conducted in Chinese. Although this had not been planned in advance, Chinese is my native language, and several of my early interviewees (also Chinese students) suggested that by using Chinese they would be able to convey their thoughts more precisely. Nine of nineteen interviews were conducted in Chinese. Because these interviews were conducted in a foreign language, the transcription process had an additional level of complexity than transcribing from English audio to English text. I did the language translation “along the way”, that is, I listened to the Chinese question and response and typed up the response in its English translation. For these interviews, I have tried my best to preserve their original meaning.
while making this translation, but I realize that I may have injected some subjective content into my choice of English word translations.

Figure 2. Framework for the Interview Sessions.

**Student Interviews**

The first part of the student interview was designed to learn about the participant’s programming background; to recognize how s/he thinks about the role of collaboration when
completing OOP projects with teammates; and to understand the typical tools or software s/he uses for collaboration in team programming or other group projects.

The second part of the student interview is presented and probed as a “tool walkthrough” of the design concepts I had created using the cloud9 screenshots. In these cases, I began by simply getting their reactions to the scenario and proposed design, followed by additional prompts as needed to gather as much information about their reactions as possible. This part of the interview was aimed at probing the students’ awareness of the functionalities that were being offered. If they did not “see” or notice the feature of interest, I continued to prompt them to look more carefully and if needed, I pointed specifically at features for which I was seeking reactions. I also asked them more generally about the proposed features and whether/what further suggestions they might have based on the current design.

**Instructor Interview**

The first part of the instructor interview is aimed at grasping their OOP teaching background and teaching objectives. I also use this section to understand how they assign and grade OOP team projects. Finally, I probe their thoughts about current tools that their students may be using for collaboration.

For the second part of the instructor’s interview, I used a style similar to the students, with interview questions given along with the design prototypes. However, for the instructor’s interview, I was targeting how they think about certain functions within the tool, and in particular whether they think these functions would help their students in the collaboration process or not (i.e., not considering themselves as users). Moreover, I also asked whether they might want to use such a tool to help their students in solving problems (programming errors). Lastly, I explored whether and how they might use such a tool as a part of a grading process.
During the interview sessions, I made sure that my interviewees feel comfortable and relaxed. I also made sure that my interviewees understood the purpose of this research at the beginning when I provided my introduction and general instructions. When closing interviews, I also made sure that interviewees have enough time to add any additional thoughts they want to convey. At the end of each interview, I thank each participant for their time and effort, and told them what will happen next with the data they have provided (Blandford, et al. 2016).

**Data Analysis**

Due to the nature of this study, the first step of the analysis for this study was to transcribe the audio recordings into text. This could be done at different levels of detail depending on what the researcher needs. For this study, when the interviews are conducted in English, I have done full transcription of all words that participants have mentioned. When the interviews are conducted in Chinese, I have tried my best to preserve the original meanings of Chinese into English.

In order to retrieve meaningful data from the transcription, after finishing the transcriptions, I have applied the coding technique that is often use in semi-structured qualitative studies. This involves identifying words, phrases, thoughts, actions, and so on, giving them useful labels or descriptors (Blandford, et al. 2016). I have followed the coding process that is introduced in Chapter 5 of the *Qualitative HCI Research* (Blandford, et al. 2016). Specifically, I have used Thematic Analysis for this study. My findings will be presented in Chapter 4.
Chapter 4

Results and Findings

Chapter Overview

This chapter presents empirical results from my study. The findings will be presented in four main sections: the prior experiences of my student interviewees when learning and using OOP; these interviewees’ reactions to the novel design features injected into the Cloud9 tool; instructor experiences with respect to their teaching of OOP; and instructor reactions to the tool mock-up.

Student Experiences in Learning OOP

As described in Chapter 3, the first part of the student interview was aimed at understanding each student’s programming background and educational (or work as relevant) experiences. An important sub-goal was to investigate how the student thinks about the role of collaboration when working on team OOP projects, and to investigate what tools and tool-related experiences they have had as part of their collaborative programming activities.

Frequency of OOP Application

Most students reported that they have spent considerable time learning about OOP in IST programming classes. Among the sixteen students I interviewed, 14 (87.5%) of them said that they had used OOP frequently in their education.

[S5] For every programming courses I have taken in IST; I have used some sort of OOP.
Only the first programming course in IST has not used OOP, all other courses are embedded with OOP.

Mostly in every programming course, I would say 80% of time.

Only two students said that they had relatively little experience with OOP. These students were both in what is known informally as the “Integration” option within the IST B.S. This option does require introductory programming instruction in Java, but by the third year the focus has shifted to topics such as organizational impacts, project management and frameworks for technology integration within modern industry:

Less than 40% of time. Because I have changed my major to integration option.

I have used OOP about 30% of the time while I was studying in the College of IST.

As I expected from my recruiting methods, most of the student interviewees are still in progress with respect to their degrees in the College of IST. A few had recently graduated and have already begun their career in industry. One of the students who now works as software developer mentioned that he uses OOP whenever he has to program for work [S11]. Another student who has had industry internship experience remarked that he appreciates well the benefits of using OOP, and that he uses it all the time [S13].

I use it whenever I have to program for work. I would say frequently, as a software developer.

I use OOP all the time. It’s a more manageable concept in CS, because now you are able to have multiple objects as far as inheritance.

Two of my interviewees are currently Ph.D. students who carry out programming-related work as part of their graduate research activities. They both indicated that they continue to use OOP quite often, but now it happens as part of their research projects:
[S3] Yes, I use OOP in my own research project. So I use it quite often.

[S14] I have learned it in college, I still use it now in grad school. I should say I use it quite often.

**Why OOP Is Important to Learn**

All of the student interviewees (16/16) reported that OOP is very important to learn and master. As indicated by the comments in the earlier section, many students were able to articulate a range of benefits for the OOP paradigm. Specifically, some students emphasized two often touted concepts associated with OOP: reuse and maintenance.

[S11] Yes, because OO is a very useful skill. I guess for solving problems, being able to create a programming solution in terms of objects. It also comes with reusability and maintenance.

[S14] OOP is better for the maintenance of the code. It is important to learn, even in industry.

[S13] Yes, important, because OO is the way that feature, that's why most programming languages are object-oriented, because then you will have an identity to an entity, you gonna identify an entity in a program that has name or some specific attribute that just have in different classes.

Some students proposed that OOP is important because it is helpful in teaching novice programmers about the conceptual relationships among the objects in a program, and to understand the logic behind the project. In the object-oriented design thinking that accompanies OOP, students must learn to assign coherent responsibilities to the objects that will implement the design, and create these objects by defining and inter-connecting classes. These classes exist within an inheritance hierarchy, so learning about class-specific behavior that must be created,
and existing behaviors that can be inherited and simply “called out” as needed is an important aspect of learning OOP (Wirfs-Brock and Wilkerson, 1989).

[S4] Yes, I personally think it is important. It is important especially for students who just begin to learn programming. This really teaches them how to think about the relationships between each object and to understand the structure and also the logic behind.

[S5] Yes, it is embedded in the programming. You do it without knowing the theory behind it, so you should learn it before you do it.

[S8] It is important because I feel like this is the fundamental knowledge of learning programming. Learning it is very beneficial for the future.

From some comments I have gathered, I can see that students who hope to become software developers have a sense of what they will encounter in the job market when they graduate (all IST majors are required to complete an internship and many gain industry experience in this way). Over one third of students have mentioned that mastering OOP is a key to winning a job as a software developer after they graduate.

[S2] It is very important for students to learn and master object-oriented programming. It is extremely important for students who are going to work in the software development field.

[S6] Yes, it is important. It is very helpful when you are looking for jobs.

[S7] Yes, it is important. It has been used a lot nowadays. It is also very helpful if you are looking for jobs in the future.

One student who is now working in the industry commented that OOP is often used in many companies, and it also ties to other software development frameworks and process, such as
those typical of agile development (Beck et al., 2002). In agile settings, teams of programmers often develop parallel objects that is shared and updated in short iterative cycles, sometimes known as “extreme programming”.

[S15] Yes, it is very important to learn, because many companies use agile development, which requires OOP skill since the program development breaks down into modules. It is an important skill to have but not mandatory.

In summary, students who have learned OOP seem to understand when and how to use it. One nuanced comment is listed below. The student mentioned the modularity OOP promotes, which can be helpful for efficient collaboration [S16].

[S16] It’s important. It’s professional that people use it in development programming field. People can write different parts for the project, so it’s more effective and economic.

**Challenges of Object-Oriented Design and Programming**

Some students think that the most challenging part of OOP is to understand its structure and the logic behind the project, especially when they are first building the project (design), where they are trying to understand the relationships between each type of entity. At a higher level, OOP is known to have some common challenges, for example grasping the purpose and workings of the Model-View-Controller framework (MVC), which is a common OOP design pattern used in building modular user interfaces.

[S5] I think in the beginning, when you are trying to understand the relationships for the design and to understand the structure of the project.

[S6] The biggest challenge is to understand the logic behind the project, also to make sure your ways of design will work.
[S8] I think the biggest challenge is when you are constructing the framework. Figuring out how many classes you will need to make the project simple and clear.

[S9] Biggest challenge is the logistic side, to make sense of the MVC. For me it is hard to start to thinking about the concepts when creating UML diagram.

**Challenges of Object-Oriented Design and Programming in Teams**

As more direct input to my research questions, two students discussed OOP challenges that arise from the perspective of working in teams. For example, they expressed that one challenge of OOP arises from people in the same team who are at different levels in programming. Because of this, the workload is often divided unevenly. Note though, that this comment is not necessarily specific to the OOP nature of the teamwork; in fact, one might expect that the inherent modularity of building a set of related classes is a helpful feature of OOP with respect to division of labor.

[S15] I think it is difficult, I am a technical person that likes to do more programming work, but some people like to do more documentation work, it becomes an issue when there is an uneven distribution of skill, it makes the dividing of work to be difficult.

[S16] The biggest challenge is that people in the same group have different standard, some people can only do the basic stuff, so the efforts put in are uneven.

Some students have mentioned that the biggest challenge of OOP is to understand their teammates’ code; to grasp each other’s coding style and logics. Again, this appears to be a more general comment, as it is not clear why learning about coding style and logical thinking would be specific to the use of OOP. It may simply be that OOP team projects are the only source of examples that these students have experiences.
[S1] Sometimes it gets hard in OOP projects if you and your teammates are having different thoughts or opinions.

[S3] I think to understand and use others’ code is the biggest challenge for me. When you are collaborating with others, you need to know how to use their methods and attributes.

[S12] I think to figure out my teammates’ code is the most challenging part. To understand their logic, and whether or not I can change the things they have written.

[S15] It requires programmers to have very good documentation skill and will need to read the parent class's documentation very carefully to fully understand the what abstract object that needs to be passed down and used by the child class.

**External Tool Usage**

Students reported using quite a variety of tools to support their team OOP projects. The main tools they use for developing code are NetBeans (netbeans.org) and Eclipse (eclipse.org). Groupme (groupme.com) and the chat function available as part of Google Doc (docs.google.com) were the most common tools used for team communication. Most students said that they use GitHub (github.com) and Google Drive (drive.google.com) to share their code contributions. For the most part, the students said they were happy with the tools they use (hardly any worried that they had poor support for collaboration); however, they did mention some limitations regarding these tools’ collaboration support.

Some students complained about GitHub, citing reasons of complexity and its asynchronous nature. GitHub is a code hosting platform that supports version control and collaboration among programmers. It is complex to use for beginners because in order to use Github fluently, students must watch rather tedious tutorials or read instructions, spending hours
just to become familiar with the basic elements. If novices do not spend this “up front” time they are likely to make mistakes.

[S9] Github is very confusing at first, but if you get used to it, it’s pretty easy to use, but sometimes people will still mess up when they commit something to Github.

[S15] For programming collaboration tool, we did not use any in the earlier, but in the late programming courses we started to use Github as the collaboration method since it has the version controls. We don’t use Github well, we often mess up with the push and pull process.

Other students said that they do not like the asynchronous nature of GitHub — when one team member is actively working on code, other team members will not be able to see the changes being made until the other person has finished and uploaded a new version of his module. Some interviewees mentioned that they would like to have a coding tool that can function like Google Doc where people within the same team would be able to write and view the code together in real time.

[S1] I use Github for sharing code with my teammates; use Netbeans to finish the coding assignments; and use Groupme for communication. These tools are mostly asynchronous tools, and they do not support collaboration well.

[S2] I think the collaboration process is difficult. Generally speaking, there is no tool to support programmers to write code together. You have to wait for one person to finish writing some parts of the code and push it to Github and others can then pull it.

[S4] For communication, I use Groupme. For sharing code, I use Github. I use NetBeans for coding. I don’t like using Github. It is hard for beginners to figure out how to use it
I wish there could be something like Google Doc so we can write and see each other’s code together, that will be much more convenient.

Many students use Google products to support their collaboration process. They tend to ascribe this preference to these tools being relatively easy to use, and also that Google has assembled a single suite of integrated tools.

[S11] At school, we have used Google Drive and Gmail to send the code around. We did not use Github or anything like that, and it worked. Pretty much what we did was sending the project around. Like I need this class, I would just send you the class and you can have it. Sending the project around. I definitely think we should use something else.

[S15] Most of the time we use Google products as collaboration tools, since it integrates all the tools it is easier than use a single tool.


**Team Collaboration During OOP Projects**

Some students enjoy working in teams to complete OOP projects. This seems to be because they believe this style of work will help them to learn other things, and it may allow them to reach out for help as needed from their teammates.

[S1] Compared to individual assignment, I like team collaboration more. Because when you are doing the project by yourself, if you encounter problems, you don’t have someone there to discuss with. Team collaboration works better because you can share your thoughts and come up with a solution when you encounter problems.
Despite the fact that some students like to work in teams, they report that the collaboration process can be challenging because there are no existing tools that would allow them to write and see each other’s code in real-time (i.e., synchronous programming). A few students specifically mentioned that being able to see their teammates’ code when needed is extremely important for collaboration.

[S4] In general, the collaboration process is hard. In current technologies, there is no existing tool to support people to write code in real-time. Say Github, only one person can go and change a class, one at a time. If two people are doing that at the same time, then there will be an error. If two people are uploading different versions, you, as the third person, don’t really know which one to use.

[S8] Most of times, we will choose which parts to work on in the beginning, however, for some cases, you need to use your teammates’ code. Then it gets complicated, because you need to get your teammates’ code. It is such as waste of time in waiting for them to send you their code.

[S11] In school, it was definitely more difficult for sharing, when you are working with other people, you don’t really know what kind of changes they are making, what exactly they are doing. They are working on their personal machine, and you can’t really see it until they send you, you can only see the final product. When you have question, you would be like, ‘Hey I am having trouble, can you help me with this?’.

**Team Communication**

All of the student interviewees indicated that they use Groupme for team communication. Some students also use tools such as those provided by Google, or Wechat (a mobile text and
voice messaging communication service) to support their communication. They consider communication to be one of the key enablers of successful collaboration.

[S1] I think if the communication in between team members works out fine, then the collaboration process should be very easy.

[S11] Groupme was the most dominant one we used for communication. I think it was very useful for communication. Usually, the first thing we do when we get into a group is to create a group on Groupme.

[S15] It is about 50/50%, sometimes we sit in IST and do the programming together, but if not everyone can be meeting, we normally use Google Hangout to communicate.

**Working Time and Location**

Students frequently work from different locations; they also sometimes work at the same time but in different locations [S6, S16]. When possible, if they have a really difficult problem to solve, they would like to address this using a face-to-face discussion. Among 16 students, all of them mentioned that they typically collaborate from different places.

[S4] We work from different locations. I usually work on team assignment from my apartment.

[S5] We work from different places, somebody online, somebody offline. Some people finish some initial parts first, others finish the rest.

[S6] We work in the same time, but in different locations. Different people write different classes and functions, we put them together in the end, and check for its operation.

[S16] We work at the same schedule, but we don’t work in the same place. People will upload their code to the Google Drive, others will then try to download the code and put them together.
One student mentioned that he usually works from different places with his teammates when completing OOP projects, but also added that he does not like this style of work because team members are not able to see each other’s code when necessary.

[S8] When first given the assignment, we decide what we want to do based on our strength and weakness. Then, we work from different places. However, this is not convenient when we want to see each other’s code, or to put all the code together.

In a few cases, the students mentioned that they collaborate from different locations, but said that they also work in class together sometimes. They will also occasionally meet up if there is a major issue that could benefit from their teammates’ help [S12].

[S9] I work both from home and in class. Sometime the group will hold discussions during the class time to talk about the process of each individual part. But most of the time, we are trying to finish our own part at home or in the library by ourselves, and we will have a group meeting each week to talk about the process and test the code of each one, I mostly do coding at home.

[S11] In college, I would say I use a mix of those, sitting at the same table, meeting up and we could finish some decent work. Sometimes, when I am developing, it’s easier to work on my own, from my apartment, occasionally meet up at school.

[S12] It depends. If people need help, then we will meet up. But I think we are mostly working remotely and then using Groupme to communicate.

**Student Reactions to the Design Mock-up**

In the second part of the student interview, I probed the students’ reactions to the mock-up creating an “enhanced” version of the Cloud9 tool. As documented in Chapter 3, I used my own experience in team programming, my review of the collaborative editing literature, and my
analysis of existing tools to identify functions that might be important and useful for synchronous programming. In some cases, I had deliberately left options quite open so as to invite additional brainstorming; for example, I asked the interviewees to generate their own ideas about how history browsing might work in this new design. In SBD, brainstorming is considered as a key concept for expanding the space of ideas (Rosson & Carroll, 2002).

Overall, students were quite happy with the design ideas shared through the scenarios and mocked-up screens, though they were still able to articulate ideas for improvements. These ideas will be discussed in the Design Suggestions in this section.

Co-Editing While Programming

When focused on synchronous programming, the student interviewees generally liked the idea of being able to write code in the same section and at the same time as other teammates. This is the essence of synchronous editing and as the earlier section indicated, many students are familiar with this from prior use of the Google Doc tools.

[S2] Looking at the screen, I think the most important is being able to edit and collaborate with my teammates in real-time. I also think the ‘Run’ is important because I would like to see if the code works or not.

Staying Aware of What is Happening

Beyond the basic support for synchronous editing, students shared reactions to two features that would enable them to either emphasize what they are doing (pushing updates) or catch up on what their team members are doing (receiving updates). With respect to sharing their own updates, they saw both sharing of error messages [S1] [S12] and posting of comments as important [S3] [S11] [S13].
In the case of error messages provided by the compiler (after using the “Run” button), the programmers want to be able to share their problems (i.e., as documented in the scenario’s error message) with their teammates who are also around and engaged in the project. Interestingly, the students also seemed to recognize that not all errors are problems that need to be seen by everyone all the time. Rather this should be a user-initiated action, a sort of implicit call for help:

[S1] I wish if there is an error that I can’t solve, my teammates can see it in real time. I wish there is a button that allows me to share the error message whenever I want my teammates to see it. If it is a smaller problem, I wish to see the error message first, then if I can’t solve it, I will choose to share.

[S12] When coding together in real-time, I hope that I can have the option to hide my error message from my teammates, and also have the option to share the error message with them when I wanted. Because I don’t want to interrupt them when not necessary.

Another option for sharing updates was seen in the commenting feature, for example conveying something they have just done, or what may still need to be edited – right at the time and place where the editing would be most useful [S3, S11]. Thirteen (81.25%) students called out the comment box as a very useful feature for coordinating synchronous work; I saw this as quite interesting, because commenting is more common as asynchronous communication, for example as when one leaves a Facebook comment in response to an interesting photo or post. In this synchronous editing context, commenting is similar in some ways to a shared error message, except that it is the programmer who decides to initiate the communication and the locus of the related code, not the system. In some cases, [S3] the comment might be as simple as a notification of some edits about to happen that may affect other team members’ work.
The comment box is very useful I think. You can instantly know who offered you help, there is also a time stamp you can tell when he commented. You can also accept or don’t accept his edit. To me, being able to comment and edit in real-time is very important. You can receive immediate feedback.

Ah... that will be cool!! It looks like that the k is crossed out there, and then it shows you the author, who has made this suggestion. Looks like when they made the suggestion. And then the comments I am assuming you can put why you need to remove it. Something like that, yeah, that seems cool, it looks like check mark, I guess that’s either accept the change or a X mark to ignore it. Oh yeah, definitely, I think that will be very useful. They can highlight it, and then they can either accept the change or not.

Yeah, that is cool! Time stamp is useful too. You can tell when people are working, that will be useful.

One interesting variation of the commenting reaction was related to code “ownership” or responsibility. I noted earlier that one concept driving OOP is that it enables teams to break up a problem into modular components and distribute tasks accordingly. But the implication of this is that team members might be working on inter-related pieces of a project in parallel, and these pieces might have implications for each other. The commenting is a technology-mediated feature the collaborators might use to integrate across these distributed responsibilities.

I think the comment box in real-time programming is really important. Mostly, me and my teammates won’t go into details to understand each other’s code. So if I am trying to make edits on the code, then it is important to ask if I can make that edits or not. Because it might bring effects on other classes or methods.
One of the students thought ahead to other scenarios, which is one of the goals of the SBD approach (i.e., it encourages “what if” thinking; see Rosson & Carroll, 2002). For example, this interviewee considered how much more beneficial the comment feature would be if the code being developed was long enough to require scrolling. In that case, simply being able to see where and what is being edited (and by whom) would be lost, but a comment could be used to highlight it for later review.

[S15] I think the Comment Box will be best to have, because it’s actually showing who made a change, what he made a change to, and the time. I will be able to see that my code is being changed. This scenario is only 17 lines long, if we talk about over hundred-line code, then it will be not visible to me that he made a change on which line.

Two students raised an important design issue for synchronous programming tools: when two team members are writing code at the same place, and trying to compile it at the same time, there may be conflicts that lead to errors. They suggested that the tool should build in something like a version controller, where you decide on and publish a version of code that can be compiled and used by your teammates, but at the same time, you can still let them see the work (edits) you are doing in real-time.

[S3] Having version control is important, your teammates should not be able to run the code which you are currently editing because it will bring errors.

[S4] You should set when you are writing the code, others cannot run your current version of code, or else, there will must be errors. You should add a feature so that when you are coding together, they can only run the last version that you have uploaded.

Real-Time Communication
The student interviewees were univocal in seeing instant communication as a key element in their collaboration activities. They described its role both when interacting with other online teammates, and when they are working on their own but encounter problems. This echoes the more general comments made earlier when describing their experiences with OOP team projects. The mocked-up tool prompted additional comments, for example presenting details relating to who is now online or not. Interestingly, at least one participant [S2] gave a higher ranking to the synchronous editing than to a real-time chat, emphasizing that the core support is editing together in real time.

[S1] On this screen, I can see that two of my teammates are online and one of them is offline by looking at the colors and the text. There is also a chat box at the bottom that I assume I can use to communicate. I would give the chat box a rating of 10. I think being able to communicate in real-time instantly when you encounter problems is extremely important. I would use the chat to give feedback and ask for help.

[S2] I can see who is online and who is offline. There is also a group chat we can use to communicate. I would rate it as an 8. I think most of the time, we will focus on the coding more, unless if there is a problem that I can’t solve, I would use the chat.

[S9] I would use the chat when my teammates are online, and when I can’t figure out what’s wrong with my code.

[S10] I would use it frequently. In this way, you don’t have to “quit” what you are doing, and open up another application to communicate, like what we have to do on Groupme.

Another participant suggested additional functionality for this particular scenario of shared editing, namely a way of linking pieces of the code to chat messages. This may have been prompted by the earlier exposure to the comment function, which similar to how things work in
Google Docs, can link a comment (perhaps added asynchronously) to a line or section of code. Why should it not be possible to have the same functionality in a chat, where you also have continuity of the ongoing (perhaps real-time) conversation?

[S3] I can see some people are online, some people are not. There are four people in total. The color and text helped me to determine whether they are online or not. There is also the ‘Group Chat’. I think if you can highlight the code where you have problem, and then use the chat box to communicate, it will be very convenient. I would rate this as a 10. If there is a problem I cannot solve, I will definitely use it.

Another participant [S4] noted in response to this scenario that one of his hypothetical teammates is not online. This led to other “what if” reasoning, where he imagined what he would need to do if all of his teammates were offline, for example how would he be in touch with them if he needed their help, or to share his status. He suggested that he might resort to an external tool (GroupMe), as is done now, to accomplish this. This raises a design option for the design of a synchronous programming tool, in that it is quite feasible that team members could be notified whenever they are not logged into the programming tool but there is an update to the group chat.

[S4] I think I will definitely use this Group Chat. The thing is if your teammates are offline. Then you will need to use external tool like Groupme to get instant help. But this is very useful if they are online.

**Collaborative Learning**

When responding to the code commenting scenario, students expressed that the commenting process will help them to learn about OOP. In fact, almost all (15, 93.75%) of the interviewees mentioned that the connections between code and commenting can be helpful for learning.
I think I really like this comment function. It is really easy for me to understand how to use it. This is important especially when you are writing a huge amount of code, being able to see where of the code got edited is important. It also helps you to learn what you have done wrong.

One participant was especially articulate about the semantic relationship between the code and the comments, explaining that the linking of comments to code forms a sort of perceptually intuitive “discussion”. Although he did not phrase it this way, the general notion is similar to the body of work in computer-supported collaborative learning, that uses the concept of an anchored discussion, meaning that the collaborators are at once sharing a piece of work (e.g., an image, a document) and their thoughts about it (Guzdial & Turns, 2000; Carroll et al., 2015). More broadly, communication was recognized as an important process for learning together.

Being able to hold a discussion online is very important, it is more perceptually intuitive. You can see each other’s thoughts and suggestions and learn from each other. I also think having the comment box tells you how your teammates fix your problems, you can also judge if this edit has helped you to solve the problem.

I think you should communicate before edit. Because you are collaborating, and you can learn stuff from collaboration. Instead of directly edit, I think you should have a conversation beforehand.

Reactions to Button Names and Icons

All 16 students were able to identify most of the “enhanced functionality” of the tool at first glance. Only two students did not understand what was being asked (e.g., what is the tool offering them) and needed some extra guidance. I provided this guidance by saying “If you look
to the left-hand side, right-hand side and the top, do you see the functions?” Those two students were able to focus on the inserted features after I provided them with that guidance.

[S1] I can easily understand the interface. It looks straightforward to me. It looks like the tool has got most of the functions I need. First time looking at it, I can realize functions such as ‘run’ ‘debug’ ‘share’ are embedded in the interface. I think the ‘Share’ button is used to share the environment with my teammates.

All 16 students reported that that the ‘Run Button’ is the most essential feature within the interface, suggesting that they were understanding the mocked-up version of Cloud9 as an IDE for their own use, which presumably involves considerable trial-and-effort activities (i.e., coding, testing, debugging, testing again). However, many students also mentioned that ‘Collaborate’ and ‘Share’ would be important in the context of the team programming scenarios.

[S4] The ‘debugger’ ‘run’ ‘collaborate’ are important at first glance. I would use ‘run’ in a very high frequency to understand to check error.

[S5] ‘Run’ is the most important function in this interface. I would rate it as 10.

Two students commented on the color differentiation in the code that had (presumably) been contributed by different teammates. The students who noticed this said they think it is better to have the color differentiation.

[S1] I think being able to see different colors for different teammates are important, you can immediately tell who is doing what, it is very straightforward.

Most students also understood the meaning of the Check Mark and Cross Mark embedded in the comment box (in these design scenarios, a Check Mark indicates accept changes; Cross Mark indicates reject/don’t accept changes). Only one student was confused, suggesting that perhaps the Cross Mark would cause the comment box to be closed. Even this
one confusion suggests that the specific icons used to convey different meanings need to be carefully considered.

[S12] Claude is suggesting me to delete the ‘k’, if I click on the check mark, then I am accepting the change. If I click on the X, then I am rejecting this change.

However, another exception to the general interpretability of the buttons and other mark-up was the “Share” and “Collaborate” buttons. Share was intended to convey inviting team members to join the real-time environment, while Collaborate was provided to convey starting a conversation with the people who are already in the same environment. Some participants reported that their meaning is rather similar, and that they could not guess at a difference at first glance. I noted that people did assign interpretations in general, but also acknowledge that these two words are in fact quite overlapping in meaning!

[S3] I would use ‘collaborate’ or ‘share’ to begin a conversation, not sure the difference between them. I think I would use ‘share’. Because ‘collaborate’ looks more general.

‘Share’ is more like sharing a link?

[S11] I would guess the collaborate, I guess the share obviously make the most sense.

The thing is when you talk about share, I don’t know if you are sharing error message, or the class itself. If you are trying to share a class, if you just click on the share button and then share the class, they can just get it. If that makes sense.

Responsibility and Contribution

Recall that I deliberately left open how the history browsing sub-task would work; the participants could only get the idea that it was possible to do this (e.g., when the currently offline team member came back online). In this, I expected many of the students to leverage their prior
experience with history in Google Docs shared editing and to a great extent this is what I heard from them.

[S10] I wish it could be like how Google Doc works. Where I can see the final version, and also the parts on what have been edited by team members. I expect to see different member’s edits in different colors, like using different colors to indicate different people’s work. In this way, I can easily tell each person’s work and contribution.

All of the student interviewees expected to see the editing history in the form of a timeline that they could check to see each team member’s contributions. Some [S10, S11] suggested a more specific approach, for example being able to find the contribution by checking the background color in the history. This might have been an inference tied to the use of colors to distinguish among authors in the shared code editor. Because these details were not part of the SBD process that I did in advance, they are particularly useful in starting to think about a history browser that could be part of a synchronous programming tool.

[S3] I would check who left the most comments and make the most edits.

[S7] I hope to see what each teammate has contributed by a timeline.

[S10] I would like to check the contribution by checking the history. If I am offered with this function, I would like to see different team member’s code with different background color, matching to their color when they are doing coding.

[S11] It depends on the view, if you are looking at the actual class, I guess you would like to see the code was edited by the team member’s color, maybe with time stamp on it as well. If this person does a lot of work in the beginning, and then like not doing anything else for the rest of the project, and then you can see that from the time stamp.
[S13] Time machine based, I can know which point of time that which person made a change, so it will keep record of a code every time it being changed. So if a person made a change in five minutes, if it takes a snapshot every five minute, you should able to see the time base.

General Comments about Expected Usability

In terms of the expectations about the usability of the mocked-up tool, all 16 students reported that because they could see and infer most of the functions from looking at the screenshots, the user interface seemed pretty straightforward. They felt that they could easily understand what the features would do and when they would want to use them.

[S3] The comment box is very intuitive. The comment suggests me what to do, I can either accept is or not. I can also see when the comment was made. This is quite important because if there is more than one edit, I want to know which one was made first. The reply box is also important, if I am offline, leaving a comment there is easier for me to go back and look. It is more direct.

Mental Model Provided by Google Doc Experiences

Not surprisingly (and as I have noted at various points), many students mentioned Google Docs as a source of ideas about how to interpret the mocked-up screens and as inspiration for other proposed functionality. For most of these participants, I assume that Google tools are a main source of real-time collaboration support in their daily life, whether for class projects or their own personal activities. The implication is that they are using their Google experiences as a sort of mental model that they are generalizing to the synchronous programming. The benefit is that they were better able to make sense of the novel features mocked up for these scenarios (Rosson & Carroll, 2002).
[S7] I think being able to see teammates’ name in different colors are important, it is more clear, very much like how Google Doc works.

[S10] The comment box function is like Google Doc’s. You can put comments on the side. This is very like combining the Google Doc functions with a IDE, I think it will work out very well.

[S12] I hope the chat will work like Google Doc, you can copy and share your error message, in the meantime, telling them where you are having problems with.

[S13] On the right-hand side, I can see that my team members are in the same environment, and can actively read and write. I think this is kind of like how Google Doc works? People can access the doc and make edits in real-time.

[S14] Sure, it is very like Google Doc and Slack, so keep it. Students should have mental model, so it is easy for them to understand.

**Design Suggestions**

There were some new design ideas suggested by the students. This was an intentional aspect of this SBD walkthrough process, in that I deliberately left many details of the possible interactions underspecified, relying only on a general storyline and some mocked-up screens. Two students who were focused on the coordination of code in a team project suggested that the tool could functions similar to what one sees in a version control tool like GitHub.

[S3] I would like to have version controls. For example, you can invite people to edit your code in real time, but they cannot use the current code that you are writing until you click upload or something. Before then, they can only use the version that you have uploaded last time.
As mentioned earlier, one student suggested the ability to connected a piece of highlighted code with a message in the chat tool.

[S3] I think if you can highlight the code where you have problem, and then use the chat box to communicate, it will be very convenient.

Another student [S2] was interested in providing a more elaborate “avatar” to represent his teammates who were working with him on the project. He suggested the addition of a user account photo (e.g., as part of a profile). This same user suggested a way to use chat as a sort of notification tool, for example to make it pop up or “flicker” when new messages are posted. Again, the reference was to “how Google Doc Chat” works. [S10] made a similar suggestion, using cursors in the shared editor (like in Google Doc) to show where someone is working.

[S2] Instead of using the letter in the profile picture, you can also consider to use a real-profile picture. It is even easier to distinguish who is working and giving feedback. One more suggestion, I think if the chat box would be better if it can be designed so that it has the pop-up effect. Like the Google Doc Chat, when people replies you, the Chat box will flick. Because sometimes when you are focusing on coding, you might not notice if people are asking questions in the Chat Box, and if it is an emergency issue, I wish it can get my attention or my teammates’ attention immediately.

[S10] Maybe adding a cursor, you know how Google Doc has the cursor, you can see who is doing what based on where they put their mouse on. It’s easier to recognize which class and where they are working on.

Other students had suggestions relating to conversations and communication in general. For example, [S14] felt that being able to go back and check the group chat history would be helpful, adding to the ideas about how history can be shared. [S15] was thinking about the
process of editing while also communicating about the edits, and he suggested that having a voice conversation function could help to reduce workload (the programmer could comment via voice while not disrupting his “typing” attention from the code). Lastly, [S16] noted that there might be icons for some common pieces of text, and that making such options available could speed the process of both creating and reviewing communications.

[S14] I think if you have a log to provide the chatting history will be good. Like Google Doc and Slack, every time when you restart, the chat history will be gone.

[S15] Like adding a voice conversation so you can choose to talk instead of type.

[S16] I think maybe some texts can be substituted with icons.

**Instructor Experiences with Teaching OOP**

In the method section, I mentioned that the first part of the instructor’s interview is trying to grasp their OOP teaching experience/background (recall that in this I am combining the interviews with the two instructors and the one teaching assistant; here I refer to all as “instructors”). I am also trying to understand their teaching objectives, specifically probing what they expect their students to learn and master. Moreover, I am exploring how they would assign grades for their OOP team assignments. Lastly, I am gathering their thoughts and opinions about current tools that their students might be using for collaboration.

Note that because these interviewees provided a great deal of interesting comments on topics that are of interest with respect to OOP teaching general, but not as relevant to team projects, my analysis here focuses on comments that are most related to my research questions. However, for readers interested in the general area of OOP education, the full set of interview questions and responses is provided in Appendix E and Appendix G). In what follows, I first
summarize their answers to my questions about OOP teaching practices, then shift to a more detailed analysis of comments about team projects and collaboration.

**Teaching Background and General Practices**

Both of the professors I interviewed have been teaching programming over a long period of time, and as expected, both have taught multiple courses covering OOP concepts and skills. Interestingly, one of the professors has been teaching an introductory Java course (IST 140), where OOP is not as much of a focus (general programming concepts like loops and variables are taught, leaving the design concepts of objects and classes for the later courses). The teaching assistant has two years of experience in this area of teaching.

As part of their normal teaching practice, the instructors estimate that they assign OOP tasks to their students at least weekly. This frequency might vary if they want students to try out something, but these other kinds of assignment then will not usually be graded [I2]. The main goal of these assignments is to give students a chance to practice what they have recently been exposed to in their readings or in class, and to address questions that have emerged in the course of trying out these new concepts.

For example, [I1] mentioned that an MVC (Model View Controller) assignment would be a typical assignment; he thinks that an important learning objective is for students to understand MVC application and use them in a regular and appropriate way. [I1] also said that in one course he teaches (IST 261), another key objective is to get students to be write OOP project proposals. [I2] offered a more general view, suggesting that a typical assignment would be to reflect on what the students have learned and are capable to apply. He described his teaching targeted on teaching programming principles instead of teaching a programming language. That is, his goal is for students to be able to understand the algorithm design of OO; to have an
appropriate level of computational thinking; and to conduct effective testing. He is also concerned that they can communicate in a professional way about their programming projects.

When asked about challenges in teaching OOP, the instructors mentioned the problem of students who have different levels of programming expertise. This becomes more of an issue for the later courses (e.g., IST 261) where the instructors would like to assume certain background knowledge (e.g., about MVC). Another challenge is student motivation, which of course is not specific of OOP education. Similarly, another general challenge is being able to explain complex OOP concepts in a way that students can grasp.

**Perspective on Student Collaboration**

When asked specifically about team projects, the instructors shared a general view of why it is important to learn how to interact with others when completing programming projects. The comments reflect the general arguments for collaborative learning covered earlier in Chapter 2, namely that students can learn from the different ways that they approach a problem. One instructor even made the connection to success in general, for example noting that even university professors are expected to be able to interact and learn from one another.

[I1] So one of the upsides is that for students who were seriously engaged and working on something they want to learn about, they can pretty much feed off each other, and use each other’s sounding boards and help that is, for instance right there in class or maybe they work together outside of class.

[I2] If you are interviewing some of my colleagues you would know that, we’ve had a lot of discussions about this, most of the faculty teaching programming here, but also other courses, we all think that it’s very important to learn how to interact with others, and
how to split work, how to define responsibilities, how to take seriously, so these set of skills are very important, they are important for people in life in general.

With respect to coding and other aspects of project sharing, the instructors were concerned about uneven distribution of work. This concern seemed to be tied to the earlier comment about teaching students who vary considerably in background and OOP expertise. That is, a “weak programmer” in a team may be able to just get away with not doing work.

[11] No, I don’t expect that. There was issue of people not pulling their weight on their teams which is kind of a known issue. Like if have 3 or 4 peoples in a team, sometimes one person “gets away with not doing a lot of work”.

[12] With that said, what we found was that especially in programming, especially in the situations where you only have a few people that we have a distribution of ability to reply us. For some people are very good at it, and many people aren’t at all. You might have team of four students, and you will see that actually only one student does all the programming.

Perceptions about Collaboration Tools Used by Students

For the most part, the instructors seemed to be satisfied with the tools their students are using to support their code development and when relevant, their team project collaborations. Not surprisingly, the tools they mentioned correspond well to those cited by students, including NetBeans, Eclipse, GitHub, Google Doc, and Groupme.

[11] Whatever they choose, I only have a few specifics. They could choose their platform or IDE, whether those platforms and IDEs have collaborative capabilities. The tools have been a freedom of choices of IDE, possibly Github, and some kind of mechanism for communication tool like Groupme.
IDE such as NetBeans or Eclipse, and other online collaboration tools for them to communicate and share code; and write document. Google Doc is a useful tool for students to write together. Share code they typically use GitHub; I think it works fine.

[I3] specifically mentioned that he was aware of a student team that used a tool that would support collaborative editing and coding, but this was not the usual case. Both instructors were quite open as to student discovery and selection of novel tools if they were available.

[I1] But I haven’t yet or used to seen any collaborative editing coding tool.

**Grading Methods for Team Projects**

The instructors have personalized approaches to grading team projects. [I1] uses a pre-defined rubric that scores team members’ contributions. Rather than grading a team effort, [I2] relies on grading of individual work and considers the team effort to be more formative. [S3] assigns grades using team members’ feedback that is provided through peer evaluation forms.

[I1] It would be graded on a rubric that’s on Canvas.

[I2] So we here have come to think about teamwork more as formative, while the evaluation happens in individual bases. The evaluation was your individual score based on how well did you do on that project.

[I3] We ask students to do peer evaluation to assess their teammates. To understand how much each team member have contributed to their project.

In general, there were no specific problems raised for team grading, although it was clear from earlier comments that there are issues of varying expertise and team contribution that might be operating behind the scenes in collaborative assignments. My guess is that the instructors have come to accept these as ongoing issues and created their own strategies for fair and consistent grading of assignments that involve student collaboration.
Instructor Reactions to the Tool Mock-up

Like the students, the instructors were guided through scenarios and mock-ups created from Cloud9 screens (Appendix E). However, in their case, I was asking them to think about scenarios that involve their interactions with OOP student teams, either in the process of the projects being completed, or afterwards when they were assessing team’s results. In general, these interviews were much less detailed, as most of the mocked-up functionality was aimed at supporting students and teams. Nonetheless the mock-ups evoked useful reactions.

Initial Reactions to Tool Features

In response to the initial browsing of the first prototype screen, instructors reported that the tool looks like a “normal” IDE but that it also includes information about the online status of students who are working in the environment. One interpretation is that for people who are quite familiar with programming tools, this status information would perhaps be the most novel aspect of the enhanced tool design.

[I1] It looks right now like a reasonably conventional IDE with some extra information up here on the upper right, that’s green and red indicating whose currently online.

[I2] So you have got different students on the right. They have read write access to this, one of them is offline, so one of them is missing. Assume this is a team I’m working with?

Real-time Communication

The instructors reported that a real-time chat would be useful, while at the same time offering suggestions for how to improve it. Echoing an earlier suggestion of a student (who wanted the chat tool to “pop up” just when a new message is added), [I1] suggested that when students need concentration (what we often term “heads down coding”), they want to turn the
chat off. Referring to the literature review (CMC), when to interrupt students is still a big concern that has not yet been solved by researchers. Again echoing one of the student’s suggestion (speech recognition based on voice comments), [I2] suggested that it there could be an audio-chat service, so that students can work on the code, see each other’s coding efforts in real-time, and discuss using voice.

[I1] It depends on the comfort level that people using it. I say that because sometimes if you are very engrossed in a section of your code, you want no interruptions, so depends on what mode you are working in. and here if we have got a really straightforward program that might have simple error, I would say you probably not in that intense deep focus kind of mode. If you are in that mode, I think it would be a distraction and I would want the programmer has the ability to turn it off or just go offline and like nobody can bother me. If it’s set up so that when I tell people I am online, then they can bother me, that’s fine, I just need to know it and to make a conscious choice.

[I2] The conversation box can be useful when they are working remotely. So they are not in the same room. I think it might be better to have this in a multi-modal fashion such that you have an audio-stream where they could talk, so they don’t have to write, because that’s the code, so they can write, they can write the code, but if they want to talk about while writing interacting… and later … the Chat Box might be too formal. I think it’s probably easier if you are able to mark something up, while talking about by an audio.

But it is certainly something good.

Grading

I was particularly interested in whether a synchronous editing tool of the sort I prototyped in these screens would be seen as helpful in the grading of team projects. My guess was that
having more detailed information about who did what (and when) could be quite enlightening when making decisions about what each team member should “earn” from a project. And indeed, all three instructors indicated that they would find such tool useful in their grading. [I2] offered a more detailed account of what he would like to see. Importantly, he assumes that students will still be using GitHub for their version control, so any synchronous programming tool would need to have some integration with this as an external service. This contrasts to a comment made by a student who was imagining that the new tool could be extended to have its own version control and code integration functions.

[I1] it possibly could. It looks maybe analogous like Google Doc, some tracking about who made which changes, to that extent that it reflects that.

[I2] Who did what? I might want to drill down into this more if there are issues with one student is not doing anything, and other students in a team comment on that, I don’t drill down what they did. I would probably want to see something like that integrated with the Git, so I would know Git check-ins. You can automate that of course. The other thing I’d like to see there is some statistic, overall statistic, who changed how many lines of code.

Delving into a more in-depth analysis, [I2] started to envision other functions that might be provided by an educational IDE, that is directed primarily at instructors and teamwork assessment. In the comment below, he is concerned about the availability of useful code on the Internet; to combat that, he imagines a function that could compile and display some sort of statistics regarding what was actually written by his students, making clear what they did rather than what was found, copied and pasted in from the Internet. This is somewhat reminiscent of the TurnItIn tool (turnitin.com) that has become commonplace at Penn State as an academic integrity insurance against plagiarism. It is interesting to consider how TurnItIn or similar tools might be
specialized for the case of OOP code, and indeed how much an instructor would choose to “penalize” students for finding and adapting examples of library functions in use.

[I2] And also, there is a lot of programming nowadays happens by copy and paste, people copy and paste out of like something like Stack Overflow, then they match it to suit their needs, and with complicated APIs that’s often the case, very few people bother to understand API that they are using in Stack Overflow, and I don’t mind there, I have to get done with things. There is only so much capacity I have, so because of that, it’d be important to figure out whether somebody that actually wrote the code or whether they copy and paste in that. Then that will be a bit more difficult from a technical perspective, but that would be nice to have, some statistics how many things, what students actually did.

This same instructor also engaged in “what if” thinking in the sense of worrying about scale. If a tool provides detailed information about individual students’ contributions, then one possibility is that there is yet another record of work that instructors must review and assess to assign grades fairly. [I2] noted that he usually has many students in his class, so he would hope that a tool tracking individual contributions would include higher-level information about the nature of the overall contribution.

[I2] The serious thing I have 40 students in my class, I don’t have time to look at every single student’s code, or groups like individual who contributed a lot, they really understand in that, so I need higher level information about that. But this is still important, to show whose contribution and make sure that every student actually does and make their own contributions.
In the end, [I2] shared quite a positive response to the directions suggested by the mocked-up screens and scenarios, conveying that the topic is of interest to him and these ideas seem to be heading in a good direction.

[I2] This is an interesting, certainly interesting design on how do we collaborate on code.

It is a certainly an interesting space to see how code can be learned together and collaborated together, so to transfer some computational thinking.

**Summary**

Although the scenarios and mocked-up screens varied for the student and instructor interviews, there were a number of shared themes and comments. Not surprisingly, the students offered much more detail, given that they were stepped through a shared programming task and probed for reactions to the hypothetical task steps, screen by screen. In contrast the instructors were offered a higher-level view of the tool ideas and asked instead to think about grading.

Overall, the mocked-up design features appeared to be both well understood and well regarded. Both students and instructors reinforced the value of synchronous editing; the students in addition had a number of positive reactions to (and design suggestions for elaborating) the ideas for sharing and maintaining awareness of one another’s work (e.g., use of the error messages, comments, and a novel idea about how to make connections between chat messages and code). One student and one instructor also suggested the support of multi-modal communication, for example being able to speak while editing code, where the speech might be used just as such (i.e., as if you were co-located but working on different computers) or where it might also be converted through speech recognition into chat messages that could become part of the team’s working history.
With respect to history – and indeed with respect to many ideas offered by both the students and the instructors – there was a clear presence of “Google-ization”, namely the various sharing and coordination techniques that have become accepted and popular through the use of Google Docs and similar services. Of course, I have also been affected by my own use of Google team tools, so my mocked-up ideas about who’s online, color coded text, and so on were obviously influenced by that experience. One simple and general implication is that Google could help programming teams a lot by creating a specialized version of their tools that is tied to the structured and text-based work of software development.
Chapter 5

Discussion

Chapter Overview

Whereas the results and findings section in Chapter 4 reported on what I found, the discussion section is primarily focused on what I found that I judge to be most significant and interesting. In this chapter, I return to some of my most interesting findings and follow these with design implications for the ongoing development of synchronous programming tools.

Co-Editing While Programming

Most participants were positive about being able to edit and write code together with their team members in real-time. However, two participants gave critical comments relating to version control concerns, worrying that if two people were literally editing the same line of code, conflicts could occur. This is a classic and long-lived problem for synchronous editing, with many proposed solutions but no single accepted approach (Greenberg and Marwood, 1994; Sun and Chen, 2002). These participants offered what is often called a “locking” approach: when coding in real-time, team members might be granted access to view what others are writing in real-time, but when compiling, they can only be granted with access to run the last-published version instead of the one which is currently being developed. This might help to avoid prompting errors when compiling.

A few participants have mentioned that they would like to have the option to choose whether or not to share error messages generated from the compiler to their team members. They
think instantly sharing error message which everybody in the same environment can see will definitely bring disturbance. On the other hand, other participants seem to be pleased with this feature, and did not give any negative comment.

Real-time Communication

Two participants (including one instructor and one student) have mentioned that when coding with others collaboratively, text-based communication is less efficient than communicating over auditory channel. This is because a programmer might want to write code and talk at the same time about what needs to be fixed (sometimes called eyes- or hands-free communication). Having to type code and then switch to typing chat messages takes attention and effort away from the main task of coding. I noted in my literature review (Chapter 2, CMC) that auditory communication may be more effective in terms of workload (it also brings nonverbal richness such as voice tone and rhythm). However, an important shortcoming is that it is challenging to store and review analog data of this sort for a history log (unless it could be automatically converted into text). Auditory information is also more distracting because it is heard by anyone in range, so depending on whether and how the team members were co-located this might also be an issue.

Two participants mentioned that having a pop-up effect for the chat might be preferable. One participant gave as his reason the chat’s possible disruptive nature; he thinks that having the chat always open may be distracting, as it is scrolling whenever someone adds a message. He would like to have the option to turn off the chat while he is concentrating. Another participant made the counter argument, saying that when he is focused on writing the code, he might not notice the conversation going on. So, if there is something needs immediate attention, he would like the chat tool to have a pop-up effect.
Shared Responsibility for OOP Development

There are several areas of concern about the shared work of an OOP team project. One known issue that is common for student teams is the unequal levels of programming expertise. Some students have a strong programming background, whereas others have weak programming skills. This often leads to a corresponding unequal distribution of workload. Interestingly, one participant who has a strong programming background indicated that he does not care if he does more programming work than others in a team project. In fact, he often chooses to work on a team project entirely by himself. However, instructors were clear about their concern from a teaching perspective: Every student who is taking a programming class is there to learn something, they never want free-riders to get the credit as students who are working hard.

Students believe that in addition to the peer evaluation form (used to evaluate teammates), having the history function could be very helpful. This might deter “free riding” because the instructors or graders will have the access to check on who did what. Similarly, instructors also gave comments on the history function. They both indicated they would welcome such functionality to check students’ work if there is a problem brought up by the team.

Design Implications

In addition to the original design, there were many other interesting design ideas raised by the participants (both students and instructors). The main ones will be discussed here, including enabling different types of working status; having different colors’ cursor located on where people are working on; adding an auditory (voice) channel; and having the ability to check for plagiarism from the Internet.

One participant raised an idea about expansion to the status information (my prototype showed simply whether a team member is online or no). He mentioned that when he works in a
company, people share different status information in their work chat tool, such as ‘available’, ‘busy’, ‘do not disturb’. The idea is that other people can understand when is a proper time to communicate. While engaged in synchronous editing of code, a team member might switch to other websites to look for resources or example code. When this happens, the system could detect that status change (e.g., when the IDE is no longer in focus) and automatically convert the team member’s status to show as ‘busy’. When a programmer is concentrating on working on a problem and wants no disruptions, he or she should be able to indicate ‘do not disturb’.

Another participant elaborated the shared editing concept to be more like he used to do with Google Docs. For example, each person could have a labeled or colored cursor, and the system could display where it is as each person works. This would allow each teammate to understand where exactly others are working. The challenge in such extra information is to provide enough details but not so much that it becomes overwhelming.

One instructor participant suggested on adding a function that could compile and display some sort of statistics regarding what was actually written by his students actually did, distinct from what was found, copied and from the Internet. This seems a very interesting direction to go but implies considerable work on tracking and monitoring access to Internet content (see Hummel et al., 2008 for interesting work of this sort).

Going beyond the interview comments to a broader view of supporting collaboration among student programmers, I have anticipated two additional design implications:

- The students attempting to learn how to do team OOP projects are quite different from the experienced professional programmers where fundamental skills can be assumed. For example, students must learn how to structure a UML diagram (this was often mentioned in the interviews). However, this feature is not supported by
any current collaborative programming tools, which focus entirely on code-based representations. For an education-focused programming tool, design support such as UML diagrams and subsequent related code should be featured. In this way, students can learn how to properly structure a project with real-time discussion.

- None of the student participants mentioned the role of unit testing as part of a collaborative project. However, it is clearly one of the learning objectives that instructors would like the achieve, given the important role of testing in successful software development (whether OOP or some other paradigm). As shown in Figure 3, 4, 5, students might be provided with a testing environment that would help to learn about the role of unit testing with their teammates. For example, they could first create a function (Figure 3); next write code in the function (Figure 4); and finally run a test (Figure 5) before they actually publish the code to the environment. From a pedagogical perspective, students should learn how to properly run a unit test, and they also should learn to divide the work appropriately, and to document the results, so that everybody in a team learns how to practice this important concept.

```java
1: public class Classname{
2:     private String someVariable;
3: }
```

Figure 3. Create/New function

```java
1: public String function1(int param1, double param2){
2:     if(param1 == 1 & param2 == 1.5){
3:         return 'This worked';
4:     }
5: }
```

Figure 4. Function
Overall, I think many suggestions offered by participants were very useful to hear, though most of them have their own pros and cons. I will consider these in my next section where I make recommendations for how to best support real-time collaborative programming.

**Recommendations for Synchronous Programming Tool Design**

My study suggests that students will benefit from a range of new features to work on their OOP projects collaboratively in an educational setting. My intention was to find a way to incorporate all features that could be useful, so as to provide the recommendations for the designers of future tools.

Through the study, I have learned that the design of such tools should be simple and straightforward. Although I realize it will be challenging to do so, every function of tool should make enough sense for the first-time users; otherwise the more novice programming students will never take the time to learn the tool (e.g., the comments about GitHub). In the case of my proposed features, the participants seemed to like the basic functions, making me relatively confident about a first pass set of recommendations:

- The tool should provide a full-function compiler that provides error messages that can be tied as closely as possible to the problematic code.
- The programmer should have control over who and when error messages are shared.
• Team members in the same working environment should be able to view each other’s code as it is edited real-time; but they should only be able to run (i.e. compile and test it with their own updated code) the last-published version. Otherwise they run the risk of testing their own code on not-yet-finished versions of related code.

• Users must be able to communicate with each other. Despite the fact that two participants suggested auditory communication, this type of communication is not convenient for keeping the conversation history, so my recommendation is to use text chat. If possible, users should be able to create references between chat messages and code.

• The students appreciated the comment box as a great way for them to learn what they (or others) might have done wrong, helping them to avoid the same mistake in the future.

• The IDE should provide the history browser. The instructors indicated that they would use such function to support their grading process; the students think may motivate and push them to learn, because they know others can see how much they have contributed, including the instructors. In this way, free-riders will not take advantages on not doing anything and get a free grade.

• A chat log should be part of the history support, so that people can see not only what code has changed but what communication about the project has taken place in their absence.

• Create some sort of notification (e.g., a flash) when messages arrive in the chat. One participant mentioned that he does not like to be bothered when in a
concentrating mode, therefore, the IDE should provide the option to close the chat. It becomes an open issue for how to reach a “do not disturb” team member when an important problem emerges that needs his/her help. Perhaps there can be a special indication of urgency (e.g., an @ prefacing the comment) that can be interpreted as “so important that all team members online need to see now”. However, referring to the CMC section in the literature review, when to interrupt users is still an unknown issue that needs to be solved.

- Lastly, the tool should provide students the availability to construct UML diagrams. Students should also be able to run unit tests on the environment. These are some of the main learning objectives from the pedagogical perspective. They are principal learning concepts for the students. Thus, it is important to consider having them embedded in a real-time collaborative programming tool that is designed for educational uses.

Overall, the design limitations and implications provided by my participants have triggered a lot of new design ideas. I am confident that their constructive feedback, in combination with my original design features, can help to guide future design and development of a real-time collaborative programming tool for educational purposes.

**Limitations**

The most obvious limitation of my research study is the sample size. I recruited a total of 19 participants, including 16 students and 3 instructors. I cannot be certain, but it is possible that a larger sample size of the participants would generate an even greater variety and depth in my results and findings. A related limitation is the nature of the participants I have recruited. All
were from the College of Information Sciences and Technology at Penn State’s University Park campus. If I had recruited participants from other colleges/departments (e.g. Computer Science, Computer Engineering), or even from other campuses, I might have obtained more varied unexpected results.

Another possible limitation is the validity of my qualitative coding. Because I am the only researcher who has been studying at the interview responses and coding them into themes and categories, the results cannot be assessed for reliability. One way to establish reliability in coding is to have another researcher to code the same transcript and to discuss the similarities and differences in between those two sets of codes (Sutton & Austin, 2015). However, given the modest size and ambition of this project, in concert with its highly exploratory goals, my decision was to rely on my own interpretative skills.
Chapter 6

Conclusions

In conclusion, my design ideas for enhancing the support of real-time collaborative programming seem to have considerable potential for future design and tool development. Despite the fact that design ideas I shared would need further improvement, both students and instructors believe such tools can bring significant impact on students’ learning processes towards OOP based projects. Broadly, the initial design provided in the prototypes seemed to be easily understood by the participants. Most participants appeared to be happy with the design ideas, though many of them offered valuable suggestions on improving the current design.

In this thesis, I first reported the results and findings from the interview studies. Then, I used these findings as fuel for guidance in tool design directions, including my best recommendations useful and feasible design ideas. The main contributions in this thesis includes: 1) based on my analysis of current tools, I offered an early stage real-time collaborative programming system design; 2) I contributed an empirical study and its results to the existing literature in this field; 3) I documented some recommendations of designing and developing such tools; and 4) I investigated and reported student and instructor’s pedagogy and learning goals.

Future Work

If I were to conduct future research on this topic, I would continue to use SBD methods to refine my design ideas until the usability of the final product is considered has stabilized. Based on the final version of design, I would then create a fully functional system that includes all the new design features, and then make the system available for classroom experiments. In
parallel, I would also deploy the system to run experimental studies and collect useful log data on specific functions to check whether or not these functions could bring significant impacts to the users. Moreover, since this study has been conducted in a small scale and has targeted students who are majoring in IST, another meaningful step to take would be to run studies with students from other colleges such as Computer Science and Computer Engineering, where students also use OOP frequently. It would be useful to see how students from other learning contexts react to such design concepts, and to see whether or not they think such a tool would help them to produce high quality results towards their OOP assignments.
References


Goldman, M., Little, G., & Miller, R. C. (2011, October). Real-time collaborative coding in a
web IDE. In Proceedings of the 24th annual ACM symposium on User interface software and technology (pp. 155-164). ACM.


Sutton, J., & Austin, Z. (2015). Qualitative research: data collection, analysis, and
management. The Canadian journal of hospital pharmacy, 68(3), 226.


Appendix A

Summary Explanation of Exempt Research

Consent for Exempt Research

The Pennsylvania State University

Title of Project: An investigation of technology design features for supporting real-time collaborative programming in an educational environment

Principal Investigator: Jin Zhang
Telephone Number: 814-777-0616

Advisor: Professor Mary Beth Rosson
Advisor Telephone Number: 814-863-3450

You are being invited to volunteer to participate in a research study. This summary explains information about this research.

- You are invited to participate in a research project studying a real-time collaborative programming tool. Our goal is to investigate the technology design features for supporting this type of tools in an educational environment.
- You will respond to questions about your programming experience and collaboratively working experience. Importantly, we will ask you about some specific design features of a real-time collaborative programming tool. The tool walkthrough session of the experiment will take about 15-20 minutes; after this we will ask you to participate in a 15-20 minutes’ interview.
- The data will be saved either in the Box system protected by the PSU 2-step verification, or in a password-protected computer in our research lab (Westgate W311). The data will be deleted no more than 3 years after the close of the study.

If you have questions or concerns, you should contact Jin Zhang at 814-777-0616. If you have questions regarding your rights as a research subject or concerns regarding your privacy, you may contact the Office for Research Protections at 814-865-1775.

Your participation is voluntary and you may decide to stop at any time. You do not have to answer any questions that you do not want to answer.

Your participation implies your voluntary consent to participate in the research.
Appendix B

Part I Student Interview Questions

General (Programming Background)

1. What is your year of school?
2. How many years have you been programming?
3. At Penn State, which programming courses have you taken? How do you describe your programming skill compare to other students? (Novice, intermediate, professional, etc.)

OOP Experiences

4. Within these programming courses, how often do you use object-oriented programming? Are you an expert or familiar with multiple OO languages?
5. Do you think OOP is important to learn and master? If so, why?
6. What is the biggest challenge of using object-oriented programming?
7. Can you describe what kind of assignments you will receive when you are taking an object-oriented programming courses? How likely are those to be team assignments and is there a difference between what is assigned individually or with teams?

Collaboration Experiences

8. How do you think about collaboration with teammates when you are completing object-oriented projects together? Do you find the collaboration easy or difficult and why?
9. Think about a typical team assignment and describe it from start to end (Do you work in the same physical area, do you work from different places, do you work at the same, etc)? How are these team assignments documented, submitted and graded?
10. If at different places, what kind of tools or software do you use in order to collaborate (Github, Groupme, Skype, Netbeans, Eclipse)? [For 1-2 examples they share: ask what they think about these tools for team programming assignments].
Appendix C

Part II Student Interview

Problem Scenario Narratives

<table>
<thead>
<tr>
<th>Problem Scenario I</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are working with three teammates to complete an object-oriented programming project. Imagine that you have completed the UML diagram together and have each decided to complete certain tasks.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem Scenario II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Now, you are working on a real-time programming environment with your teammates. Two of your teammates are working online with you, one of them is offline due to sickness.</td>
</tr>
</tbody>
</table>

Scripting of the Interview Questions

1. What can you tell from the interface below? What functions within this interface you think are the most important to you? What functions are the least important?
2. Now, imagine that you have clicked on the Run button and you see an error message at the bottom. How important it is for you and your teammates to be able to see this error message? Rate from the scale of 1 to 10. With 10 being very useful and 1 being very not useful.
3. Imagine you are writing the “Transaction Runner” class and you have encountered this problem (Marked in the red box). You don’t know how to solve this problem and would like to have a discussion with your teammates. Based on this interface and the given functionalities. What do you want to do? What does the functionalities suggest?
4. What does this screen suggest to you? Imagine you can see each other’s code in real-time. What will you want to do next? What functionalities can you use?
5. If you are offered with this chat box functionality, how likely you will use it? Given scale from 1 to 10. How important it is for you and your teammates to be able to have a conversation within this interface? 1 being not important at all, 10 being very useful.
6. Now imagine Claude has helped you to determine the problem. What can you tell from the screen?
7. Now, imagine you are the only one online now. You would like to see your teammates’ progress. What would you do?
Prototypes
```java
import java.util.ArrayList;

public static void main(String[] args) {
    ArrayList<Account> accounts = new ArrayList<Account>(100);
    Account account = new Account(100, 1000.00);
    accounts.add(account);

    for(int k = 0; k < 10; k++)
        System.out.println("Account Balance for Acct "+ accounts.get(k).getNumber()
        + ";");
}
```
import java.util.ArrayList;
public class TransactionRunner
{
    public static void main(String[] args)
    {
        ArrayList<Account> accounts = new ArrayList<Account>(100);
        ArrayList<Transaction> transactions = new ArrayList<Transaction>(100);

        for(int k = 0; k < 10; k++)
        {
            accounts.add(new Account(100-k, 1000.00));
            System.out.println("Account Balance for Acc #" + accounts.get(k).getNumber() + ": " + accounts.get(k).getBalance());
        }
    }
}
public class TransactionRunner
{
    public static void main(String[] args)
    {
        ArrayList<Account> accounts = new ArrayList(Account(1000));
        ArrayList<Transaction> transactions = new ArrayList<Transaction>(1000);

        for(int k = 0; k < 13; k++)
        {
            accounts.add(new Account(id:1000 + k, balance: 1000.00));
            System.out.println("Account Balance for Acc ": accounts.get(k).getNumber());
        }
    }
}
```java
import java.util.ArrayList;

public class TransactionRunner {
    public static void main(String[] args) {
        ArrayList<Account> accounts = new ArrayList<Account>(100);
        ArrayList<Transaction> transactions = new ArrayList<Transaction>(100);

        for(int i = 0; i < 10; i++) {
            accounts.add(new Account(1000 - i, 1000 - i));
            System.out.println("Account Balance for Act ": accounts.get(i).getNumber() + " = accounts.get(i).getBalance()");
        }
    }
}
```
import java.util.ArrayList;

public class TransactionRunner

    public static void main(String[] args)
    {
        ArrayList<Account> accounts = new ArrayList<Account>(100);
        ArrayList<Transaction> transactions = new ArrayList<Transaction>(100);

        for(int k = 0; k < 15; k++)
        {
            accounts.add(new Account(200-k, 1000.00));
            System.out.println("Account Balance for Acct 
        "), accounts.get(k).getNumber(k)
            + "; ", accounts.get(k).getBalance(k));
        }
    }
}
Appendix D

Part I Instructor Interview Questions

General (Teaching background)

1. How many years have you been teaching programming?
2. Within these programming courses, have you taught any programming courses that involve object-oriented programming?

Pedagogical Goals and methods

3. If you have taught object-oriented programming courses, how often would you assign OOP assignments to your students? Can you describe a typical assignment? What learning objectives are you most interested in achieving?
4. What is the biggest challenge of teaching object-oriented programming?
5. How do you think about the role of collaboration when students are completing object-oriented projects together? How do you create teams for that?
6. Do you expect them do finish the work together with equally distributed workload?
7. What kind to tools or software do you expect them to use when they are completing the OOP assignments together? What have you noticed about these tools?
8. How do you grade team programming assignments? More specifically, how do you assess who did what and whether and how much each student contributed? What kind of method do you use?
Appendix E

Part II Instructor Interview

Scripting of Interview Questions

1. Imagine you have assigned a OOP project to a group of students. Now, they have encountered a problem which they cannot solve by themselves. They are asking you for help. What can you see from the given screen?
2. What do you think about this functionality?
3. What do you think about the history function? What can you see from this screen? Will you use it as a part of your grading method?
4. If students would like to use this tool to support their collaboration, would you allow them to do that?
import java.util.ArrayList;
public class TransactionRunner
{
    public static void main(String[] args)
    {
        ArrayList<Account> accounts = new ArrayList<Account>(100);
        ArrayList<Transaction> transactions = new ArrayList<Transaction>(200);
        for(int k = 0; k < 10; k++)
        {
            accounts.add(new Account(100-k, 1000-k));
            System.out.println("Account Balance for Acct ", + accounts.get(k).getNumber(k)
                + ": ", + accounts.get(k).getBalance(k));
        }
    }
}
```java
import java.util.ArrayList;

public class TransactionRunner
{
    public static void main(String[] args)
    {
        ArrayList<Account> accounts = new ArrayList<Account>(100);
        ArrayList<Transaction> transactions = new ArrayList<Transaction>(100);

        for (int k = 0; k < 10; k++)
        {
            accounts.add(new Account(k), 1000.00); // Add some accounts
            System.out.println("Account Balance for Acct "+ accounts.get(k).getNumber(k) + " : "+ accounts.get(k).getBalance(k));
        }
    }
}
```
Appendix F

Instructor I Interview Questions and Responses

1. How many years have you been teaching programming?

   Teaching OOP only has been a year and a half, since the Fall of 2016. Programming, I have taught the 140 course online for roughly three or four years now, then I have taught 140 here since the Spring 2017. For the 140, we don’t do the OOP, it is there behind the scene, but we don’t focus on it.

2. Within these programming courses, have you taught any programming courses that involve object-oriented programming?

   Yes, I have.

3. If you have taught object-oriented programming courses, how often would you assign OOP assignments to your students?

   Typically, weekly, although this term I included some others that were almost the next class period, but most often is weekly. I should say at least weekly. And in the beginning of this term, I gave them projects with some code completed, show them the completed version, and ask them to complete it by next class period or in class.

   Can you describe a typical assignment?

   A typical assignment in that class is a model view controller assignment. There are several reasons. One is that the course designer, and the person who wrote the notes on Canvas has two key assignments are model view controller and ideally the rest of their project deliverables should implement that design pattern, and in particularly, those
early assignments I said would be due either in the class or the next class period, they were steppingstone assignments to get people used to that framework, because some people depending on who they had for 242, they emphasized it more or less, so I gave them some steppingstone programs, so that would be a very typical assignment.

What learning objectives are you most interested in achieving?

Learning objectives, there are several. That students can write basic object-oriented programs that employ polymorphism, that students can write a simple MVC application; that students can write a more complicated MVC application; that the students can write programs that use persistence in a form of serialization. In our class, we have one particular form, that they could choose to use others such as text file, JSON file, and database. So that’s the kind of persistence we are dealing of right now. And they have the option to use any kind of, they need to, for the course to use that. Another objective is to get them to have the experience of writing proposals and programming to the aspects “promise to deliver” in the proposals.

4. What is the biggest challenge of teaching object-oriented programming?

I would say is the uneven background that people coming in to that course, it’s natural, it’s not something that bothers me, it’s just that somebody maybe for instance, if they had the same professor who wrote the course in Canvas for 242, they would be very familiar with MVC stuff, and then other people while maybe they came from a campus, and the campus taught in a different language maybe, might have taught python, in 242. I am not sure. I know when people coming to 242 often their first programming class won’t have been 140, it will be maybe a cmpsc class here or another class from a campus which could have been c++ or a python course. It’s trying to address what I considered to be the natural and unevenness, and make sure that people need more help on given topic get that help, not to have the other ones sit there and be bored.
5. How do you think about the role of collaboration when students are completing object-oriented projects together?

It has its upsides and downsides. So one of the upsides is that for students who were seriously engaged and working on something they want to learn about, they can pretty much feed off each other, and use each other’s sounding boards and help that is, for instance right there in class or maybe they work together outside of class. And that’s if the project assignment involves team work. And this term, maybe last term, there are individual projects, and students is interesting I heard two different comments about teams from students in 261, one is ‘I want to do the project by myself because we do teams in everything else’. The other was the issue of people not pulling their weight on their teams which is kind of a known issue. Like If have 3 or 4 peoples in a team, sometimes one person “gets away with not doing a lot of work”. I’m happy now in 261 to have individual project, students seem to be liking that more and I think they are able to go and into more depth by themselves.

So you think work individual actually works better?

In that class, at least for now. I had a class in 2016, team works extraordinary well, including one team with seven people, and It was great and they just all happened to click, so there was no slacker on the team, and they divided work appropriately. They were all learning something new about in order to complete the project, so it worked well there. I think the reason I changed was that it was recommended that I change and have people do individual projects, and that later on maybe in 311 or 361 going to be team projects, I don’t know for sure.

How do you create teams in general?

In that case in 261, I let them choose their own partners, I found that reasonably effective because often students will know somebody because somebody was in another class, and that student chooses somebody that turns out to be slacker then you have the
reasonability to dealing with that, so I think that works better than trying to randomly assign or assign in some other way. Sometimes I guess I have given thought to having a particularly strong student with a not-particularly strong student, but one of the downsides of that is the particularly strong student can feel limited in terms of how much he wants to do because he needs to help the other student to come up the speed, so I do right now let student select their own teams and I will continue to do that unless somebody gives me an argument.

What about the students who do not help their teams, like who do not have teammates chosen?

Actually I didn’t have that situation, like team created in the first week, but one or two students added the class in second week, in that case, I gave the students opportunity to join another team or to work by themselves.

6. Do you expect them do finish the work together with equally distributed workload?

If there are teams, no I don’t actually expect that, but what I hope for and for instance worked in that one I have mentioned that team from Fall2016 is that they pretty much each contribute according to their strengths and interests, so not necessarily equally in any sense of equally. But that there are no slackers.

What about some students are good at writing documents?

That gets close to the slacker, the thing is if the project goal is to do coding in groups and having “modules interoperable”, one wants to make sure that everyone does in coding. So if one person says “oh, I’m weak coder, I just want to do documentation”, that doesn’t suit too well with me.

7. What kind to tools or software do you expect them to use when they are completing the OOP assignments together?
Whatever they choose, I only have a few specifics. For 261 classes, they could choose their platform or IDE, whether those platforms and IDEs have collaborative capabilities. I don’t know, also those who wanted to or free to use Github for repository. And then I found some, just by anecdote, form of chat or something like Groupme was how they coordinated, but not necessarily coordinated well, you know two people are sitting there and typing code at the same time, just coordinate more in terms of schedule or in terms of communication. So different IDEs, different platforms, so thinking of Android and associated IDE in mac, Eclipse on mac, Xcode on IPhone, they could have a choice. Now everybody is required to use NetBeans, this is for simplicity and homogeneity in the class, so we don’t need to waste the time when we go through the code with not familiar platform. Backing up and trying to go down again to your question, the tools have been a freedom of choices of IDE, possibly Github, and some kind of mechanism for communication tool like Groupme, but I haven’t yet or used to seen any collaborative editing coding tool.

What have you noticed about these tools? Do you think they worked well?

I think they did fine; I would have been open that had students come to me “hey there is great collaborative tool when they working with a teams, can we use that”. I think I would have been open to that. I think the tools that they used works satisfactory for them.

8. How do you grade team programming assignments? More specifically, how do you assess who did what and whether and how much each student contributed? What kind of method do you use?

It would be graded on a rubric that’s on Canvas. And the ones that are individual assignment now have that, so they have essentially aspect assignment itself, then there is a rubric that goes with it, and they are closely tied together.
Appendix G

Instructor II Interview Questions and Responses

1. How many years have you been teaching programming?

   Here at Peen State for about six years, but I have taught programming classes before coming to Penn State as well. Over the years, so maybe about between 6-8 years.

2. Within these programming courses, have you taught any programming courses that involve object-oriented programming?

   Yes, I have.

3. If you have taught object-oriented programming courses, how often would you assign OOP assignments to your students?

   Depending on how you count for an assignment, there is graded assignment, also there is assignment like “hey try this out”. There are assignments in class, there are assignments people do on their own time. In my current programming class that I am teaching this semester, we are trying to have something practical almost every session that we meet, it’s pretty much never just me talking, it’s always something practical.

   Can you describe a typical assignment?

   So typical assignment is such a good assignment, will get the students to reflect on what they were just taught and applying to new situation or a new question, sometimes, I just ask them to retrace the same steps that I have shown them, before they then have to apply the same pattern like design patterns, or the same programming principles, or the same algorithmic tools that giving them there, to a different context, that they are either given,
or that they are happy to working on in a context of a large project. So in my class this semester, we started out that the program that presents the user a website to enter some details, like when you order something from Amazon, you have to enter your address at some point, and that program is to take the address and validate to store in the database and be able to bring up a list of address. We have developed the program of course that several steps, because people have to write their own server, not just write html with pages but write their own server, and communicates by HTTP. So in this context, I would give them assignment, for example, can we make that faster, we would show them how to make a parallel program using multiple threads, and discuss issues related to that such as synchronization. Then I ask them to think about how to make their web server that they just wrote over the past two or three weeks, how to make that multi-thread. And I would give them a test, hardness, a unit test essentially, that barbers the server. With hundreds of requests in a short period of time, so they could see how the server becomes faster if it uses multiple threads, because different processors and different cores of processors execute these. So they can see the successful that they are doing and they can test it and run. But they have to think about the concepts saying that percentage, in terms of multi-threading, and in terms of synchronization, etc. How they apply to their particular program, because they are still working on their own code then, so they have to figure out what is being multi-threaded, what’s being handled in parallel, what are some shared data structure between the threads that are not concurrent, and how to protect them. It might apply what they just learned, and they hopefully internalized a bit more.

What learning objectives are you most interested in achieving?

As you know, I hope to teach programming principles, as opposed to a programming language. Of course I need to teach the syntax of the language, but it is really more important leaning objectives to understand the principles of algorithm design of object-orientation; of design patterns, essentially computational thinking, proper testing, and perhaps also the ways to describe what you have done, being able to talk about it, communicate with other technical professionals. Perhaps understand the process for doing all of that, defining use cases functional requirements, defining test cases, and
running regression tests and so on, documenting the code, having a good process for
documenting changes of the code, that means margining the system, like Git. I think all of
these are extremely important in general tools that will survive the next few years when
they are on the job market when they are looking for, or to start a career. So these are
some of my objectives. And there is somewhat different here in the IST compare to CS
department, where you have a lot of learning objectives about fully mental of foundations
of computer science, and of algorithms, and scalability and so on, with much more
formal look at these things. Whereas on own end, they only care about people understand
the scalability and principle understanding. How to write algorithms very generally, so it
is a lot more applied.

4. **What is the biggest challenge of teaching object-oriented programming?**

I’m teaching broad range, student with the broad range of amplitudes. There are people
that all have their opinion that everyone can write a program, everyone should.
Certainly, I think most people can learn programming, so does it seems proper sense of
writing simple code for the Teredo that moves process a string, left right, up and down,
and so on. I can do like define some of the procedure. But as soon as we introduce even
the concepts such as the loops, for functions, when we introduce some level of
abstractions, this is when you start losing people, can’t put a number on it, but I would
say that only a fraction of those that start and want to work in IT actually have right mind
set. The right abstract computational thinking capability to really learn to program and
come up with their own algorithms and their own approaches. I think the main challenge
from teaching programming comes from these people, they still sitting in your class, still
want to or supposed to achieve certain minimum competency, so that’s one aspect. The
other challenge is motivation, I see myself is motivator, how can I motivate students to
work harder on this, because you have to work hard, this is not easy, this isn’t even easy
for those that have the right mind set to think computationally, so this is hard even for
best of times. So how do I motivate people to work hard on this, right? That’s the next
challenge. The third challenge is just the teaching challenge, how do I explain these
concepts or that are complex and that are new to people, in a way that as many as possible who understand and internalized, how do I create that learning environment.

5. **How do you think about the role of collaboration when students are completing object-oriented projects together?**

Within limitations. Here is the thing. I don’t know if you are aware of this, but if you are interviewing some of my colleagues you would know that, we’ve had a lot of discussions about this, most of the faculty teaching programming here, but also other courses, we all think that it’s very important to learn how to interact with others, and how to split work, how to define responsibilities, how to take seriously, so these set of skills are very important, they are important for people in life in general. With that said, what we found was that especially in programming, especially in the situations where you only have a few people that we have a distribution of ability to reply us. For some people are very good at it, and many people aren’t at all. You might have team of afforce students, and you will see that actually only one student does all the programming. So the learning objectives of the class is to be a better programmer, that one person might get something out of it, but three won’t get anything out of it, because they collaborate on it, they maybe see the inside of the process, but they don’t have same learning gain, and in the end, everyone wants to get an individual grade, so the grade just shared between everyone in a group, and that’s an unfair grade, it de-values the individual class of your transcript, and it de-values your whole degree. Because someone reading your transcript that someone seeing your degree, I don’t know if you are one of the three, or if you are that fourth that actually does the whole program. So we here have come to think about teamwork more as formative, while the evaluation happens in individual bases, so if you look at even the class that you took with me, you know the evaluation was your individual score, how well did you do in that project? Currently I’m teaching 411, distributed-objected computing, everyone is doing that project. Support in class, when we are doing class work and ask to pair in teams to teach each other and to work up together. So I emphasized individual work over group work in order to get people to talk, to teach, to learn. But the evaluation is individual, I think that’s quite important.
How do you create teams for that?

As I said the team work is sort of more a learning thing, not so much as an evaluation thing, and the team are usually just on the bases of a single class, not necessarily teams over a period of time, so my current class is usually teams of two and they are adhoc, whoever sitting together, so people would typically like each other enough to sit together. So there are some superficial criteria. I don’t really look at, making sure that there are different abilities in the class, so there are some super teams or something like that, I don’t pay that much attention because have too many students in the class.