CPSC 444

Team χ^2 - Milestone II

Collocated, Collaborative Diagramming

Alex Totolici
69229045

André Malan
58353061

Gabe Silk
93911014

Jré Sarenac
17191081

Piam Kiarostami
94358991

February 8, 2010

Evaluation Report

Goals of User Study

The high-level goal for our study is to understand what users need in order to achieve effective collaboration on diagramming tasks. We presuppose that tools are the biggest limiting factor when it comes to collaboration, and we aim to use this study to both validate our supposition and gain further insight into diagramming work-flows. Gaining insight into what types of diagrams are used most frequently also allows us to narrow our scope in terms of the diagramming activities we want our system to support.

The high level goals of the study were to identify:

1. How often users are **required** to do collaborative diagramming
2. What **types** of diagramming they do, and **why**
3. Whether they require **collocated** or **remote** collaboration
4. Whether they **prefer** doing collaborative diagramming
5. What they **like** and **dislike** about their current tools
6. What concrete tasks they perform when working collaboratively

Rationale for Methods Used

We chose to use a questionnaire and an interview. The questionnaire was chosen because it would quickly give us answers to the first five of our high-level questions (see "Goals of User Study"), and because it is an effective and rapid method for collecting quantitative data. Users are also more likely to participate in a questionnaire, as it only requires a small amount of time on their part.

In the interview, we were looking to get more detailed information about both the interviewees' questionnaire answers, and to gain a deeper understanding of what their diagramming work-flow, what concrete tasks they execute, etc. The semi-structured format was appropriate because we could not anticipate all the questions that should be asked, prior to conducting the study. The interview provided us with a chance to clearly identify what the major hurdles are in terms of collaborative, collocated diagramming.

Participant Pool

Questionnaires

The participant pool consisted of five user groups:

- 1st/2nd year computer science students
- 3rd/4th year computer science students
- MSc/PhD computer science candidates
- Individuals working in the software industry

We chose our participant pool to include a broad range of experience levels to assess the validity of the following hypothesis:

> *Experienced users create diagrams more often than novice users, and in general, they create diagrams collaboratively.*

Our intention was to correlate experience level with degree of collaboration on diagrams, and either disprove or lend support to our hypothesis. This information would allow us to focus future studies down to the specific user groups that diagram collaboratively most often.

73 questionnaires were sent out, of which 40 were received fully completed. Out of the 40 questionnaires received, 4 were 1st/2nd years, 27 3rd/4th years, 1 graduate student and 10 individuals working in the technical industry.

Interviews

Unlike the questionnaires, which spanned a broad range of user expertise, our interviews focused on professionals and advanced students. We interviewed four people, including two MSc/PhD students and two industry professionals. We chose this approach because our questionnaire results showed a higher prevalence of collaborative diagramming among graduate students and industry professionals, and as such we felt they were more representative of the type of user our system would support.

Brief Overview of Study Protocol

Our study consisted of a questionnaire and a semi-structured interview. We wanted to determine our target audience, understand their needs, and see whether our high-level design idea would be applicable given their needs. We used both similar and complementary questions in the interview and questionnaire to allow for triangulation.

Questionnaire (see Appendix A3 for transcripts**)**

The main goal of the questionnaire was to solicit the information above from a wide audience, including undergraduate students, graduate students, and professionals. We wanted to correlate experience level with frequency of collaborative diagramming. This would answer the question of whether more experienced users collaborate more frequently than less experienced users.

The questionnaire was administered through Google Docs. We felt this was appropriate and understandable by our subjects, given that we knew ahead of time they had above-average proficiency in computers software.

Interview (see Appendix A3 for transcripts**)**

The interviews were conducted one-on-one in a semi-structured fashion, in any convenient location, following a set list of questions (see Appendix A2). The length of the interview was roughly 20 minutes. Many of the questions in the interview were the same as the questionnaire, in order that the results from both assessment types could be triangulated. However, we allowed some deviation from these questions in the interview in order to get more depth on current practices that we were unaware of.

At the end of each interview, we communicated our design verbally to our study participants. We avoided mentioning our intended design direction until the end of the interview, to avoid biasing our interviewees during the interview. We used general terms to describe the idea, with no reference to specific interface implementations. Communicating the high-level concept was relatively easy because: a) our participants were familiar with the domain (*ie*, they regularly used UML-like diagrams) and b) they were all familiar with the concept of multi-touch tables.

Summary Reporting of Results

Questionnaires

All 40 questionnaire respondents had previous experience with diagrams. When asked to rank their usage of four standard diagram types, users chose UML as the kind they use most frequently, followed by flow charts, entity-relationship and data flow diagrams (see Appendix A1.b figure 2).

Despite the fact that UML is the most frequently used diagram type, users felt least proficient in it (see Appendix A1.b figure 3). This could be due to several reasons, such as the complexity of diagramming rules for this type, the tools available to support it, etc. This is an important area to investigate in future evaluations, and is discussed in more detail in the conclusion section of this document.

We did not find that users had a preference for a specific type of diagramming tool (see Appendix A1.b figure 4). However, we allowed users to select more than one tool, which could be the leading reason to the balanced data. This question should have been followed up with: 'what is your preferred tool' which would allow us determine what tool is more desired than others and why.

Our study indicates that in current practice users work on diagrams collaboratively approximately 40% of the time (see Appendix A1.b figure 5). By comparison, 68% of users *prefer* working on diagrams collaboratively (see Appendix A1.b figure 6), and we feel this indicates the possibility of obstacles to collaborative diagramming.

By categorizing users into three main categories: novice (1st/2nd years), intermediate (3rd/4th/graduates), and professionals (working in industry), we found that professionals prefer diagram collaboration 73% of the time, followed by intermediate users at 68% (see Appendix A1.b figure 7). This supports our alternate hypothesis, and we can identify an emerging trend showing that more advanced users tend to collaborate more often.

The main reasons users prefer working alone or collaboratively were found to be:

Alone	With Others
Have more control - 46%	Idea generation (more feedback) - 59%
Create own iteration before collaboration -31%	Better end product - 48%
Get to move at my own pace -15%	Communication (everyone on the same page) -22%

multiple responses allowed so may add up to more than 100%

When asked what (if anything) the most difficult part about working with others on diagrams is, users found the following to be the most dominant issues (see Appendix A3):

- Interpersonal problems (67.5% of questionnaires)
- Merging diagrams/ideas (25% of questionnaires)

When asked to design their perfect collaborative diagramming tool there was very little consensus amongst users. We did gain many useful ideas for our design alternatives and the specific responses are listed in Appendix A1.a.

Interviews

Our interviews revealed that users create diagrams both individually and collaboratively. They almost always work collaboratively when the diagram is being used as a communication tool within a group project. Within projects, there are only two cases where users opt to work individually: when the project itself is a solo endeavor [A3.c-1], or when the diagram being made describes a pre-existing system [A3.a-1].

We found that collaborative diagramming tasks are for the most part unstructured, highly iterative, and ultimately for the purpose of communicating high-level ideas in the quickest and easiest manner possible. For this reason, the most frequently used tool for collaborative diagramming is the whiteboard. The advantages of the whiteboard are flexibility, low cost, rapid iteration, and a minimal learning curve [A3.a-6]. Users describe flexibility as the absence of any formalized structure or symbol set -- *ie* being able to draw anything, anywhere. However, flexibility also has a downside: users find that agreeing on symbols can sometimes be difficult, with several representations of the same concept appearing in different places [A3.d-7]. This obscures commonalities between different areas and makes diagrams hard to interpret [A3.d-7]. Another disadvantage of the whiteboard is lack of space [A3.a-5b], the inability to "capture" iterations [A3.a-5b] [A3.b-5], and the difficulty of rearranging and erasing items [A3.a-5b].

Users find this problem mitigated by existing software, which defines formalisms for symbols and the ways they interconnect [A3.c-6]. Software also has the advantage of readability and digital storage. However, users felt that software, particularly that which uses traditional WIMP-style input methods, is slow and inflexible [A3.b-6].

A problem in the current practice which we did not anticipate was the users' inability to interpret previously created diagrams. This was not due to legibility, but *lack of context*: diagrams drawn on the whiteboard make sense at the time they were written, but users experience difficulty interpreting them later upon review [A3.b-5]. This is partly due to the fact that capturing iterations is difficult on a whiteboard [A3.a-5b]. When the diagram is changing rapidly, making sense of the latest revision is not an isolated task. It depends on the earlier versions, which have been erased, changed, or moved, but still exist in the minds of the people working on the diagram. People who walk into a diagramming task typically will not understand the full meaning of the diagram, because they were not present from the beginning.

Conclusions

Our results indicate a strong preference towards collaborative diagramming. Intermediate and expert users have reported, in aggregate, that they collaborate on diagrams about half the time, but that they almost always prefer it to solo diagramming tasks. The desire for tools that are flexible, easy-to-use and with a minimum learning curve indicates that the existing tools are inadequate for supporting this task in the way users want to accomplish it.

The rapid iteration typical to the early stages of design makes flexibility a paramount feature to any tool looking to assist with this kind of activity. For this reason, users currently employ 'low-tech' methods of diagramming (pen and paper, whiteboards, sticky notes) as they find them to be the most flexible. However, users have expressed an appreciation for the clarity and readability of software-generated diagrams. An ideal collaborative diagramming system should retain the simplicity and lack of structure typical of low-tech methods, while allowing users better ways to manipulate and polish their diagrams. This would be best accomplished by a system that allows both pen-based and multi-touch input, and that can handle both free-form diagramming or the use of widgets form a library.

The key strengths of our proposed system are the same as those of the low-tech tools used as inspiration. Users can quickly draw, erase and redraw their diagrams without hassle. They are not forced to use a particular set of symbols, or for their diagrams to adhere to any predefined structure or standard. They can use an infinite canvas for their work, and can freely zoom in and out as needed, thereby addressing one of the biggest issues that have been identified by users with low-tech systems. Additionally, we intend to provide version control-like features, so that the diagramming process can be replayed and users can review their changes step by step. Several study users brought up the issue of recollection and the difficulty of understanding why certain parts of the diagram ended up in a certain way. By providing a history for the diagram and a means to replay it, we hope to improve users' understanding of the diagram even if they review it long after the initial design.

There are some weaknesses with the system as well. First, the cost of a multi-touch surface is potentially prohibitive in certain environments. Second, users may be intimidated to use our system, due to the advanced technology it is based on. Third, although we intend to provide an 'infinite' canvas, the physical dimensions of the surface are a limiting factor in how effective the collaborative process is. Cost and dimensions are closely connected, and a larger table may be available for users able to afford it.

We think these weaknesses are of only temporary concern. The successful entrance of multi-touch in the consumer marketplace (thanks to innovations such as iPhone) means demand for such devices is on the rise, and as such manufacturers are working on reducing costs. The pervasiveness of multi-touch is the reason for why we believe that intimidation is not going to be a limiting factor in terms of adoption of our system, either, as more appliances are touch-based and users become increasingly adjusted to them.

The first iteration of our proposed system should focus on translating the simplicity of the whiteboard approach to the multi-touch surface, while providing basic software features such as manipulation and editing of diagram elements. We think such a prototype would be a powerful statement about the direction we envision for the system as a whole, and would allow us to further test with users and identify the next set of required refinements.

Requirements Definition

Executive Summary

Design a tool for collaborative, collocated UML diagramming, that supports flexible manipulation, revision history, and focuses on addressing interpersonal issues.

Prioritized Requirements List

Absolute Must Include

In our user studies, we isolated four critical aspects that our users required. Those aspects were the ones that stood out above all others as the things users needed from a diagramming tool. They are:

- **Must support and enhance interpersonal interactions.** Over 67% of our survey respondents mentioned that one of the largest obstacles for diagramming collaboratively was interpersonal problems (see Appendix A1.a). Our system needs to provide ways to measurably enhance the interpersonal interactions between users, reducing conflict and allowing consensuses to be reached earlier.
- **Must support both structured and non-structured input.** In our interview data it became clear that users have two competing issues with current interfaces. Pen and paper as well as whiteboard interfaces provide great flexibility to our users, but the lack of structure was cited as hindering user understanding in the long run. Alternatively, software input methods were too rigid, not allowing users to quickly create rough, freehand drawings quickly. Our interface needs to bridge the gap between these two incumbent systems.
- **Must record the collaborative process.** Our data indicates that users are often unable to understand diagrams after having drawn them collaboratively. They mentioned that if they could quickly review how the diagram had come about, then they would be able to see what thoughts went into the creation of the diagram.
- **Support Novice, intermediate and expert users.** From our data we noticed that the more experienced a user was, the more likely they were to prefer diagramming collaboratively. However, from our data we noticed that novice users suffer from interpersonal problems that do not deter expert users. Our system will enhance interpersonal interactions and our hypothesis is that this will give novice users some of the benefits of collaborative diagramming, while reducing the negative aspects.

Should Include

- **Space for collaboration.** Users felt that current methods of collaborative diagramming did not have enough space to be as effective as they could be. Space was an issue on two levels. Firstly, there was not enough surface area to represent all the ideas proposed by groups of people. Secondly, users felt that it only really supported one person interacting with the diagramming tool at a time. This feature is in the 'should include' as it will would greatly enhance their collaborative diagramming experience, but it does not contribute to our problem statement.

Could Include

- **Domain-specific toolsets.** Users wanted tools that would aid themee a in the creation of their diagrams by having tools specific to the diagram that they were drawing. We could include it as it would make iterating on our interface more easy, however due to the lack of focus on collaboration we may exclude it.

Exclude

These features were suggested by participants in both our questionnaire and interviews. We decided not to implement these features as they would be time consuming, broaden our scope too much and take away from our focus of collaborative and collocated diagramming. (see Appendix A3 for the user comments)

- **Conversion from diagram to technical information.** When asked to create their perfect system, several of our users mentioned a system that would convert diagrams to some kind of usable code outline.
- **Templates for a variety of tasks.** Several users wanted detailed templates to help them start their diagrams.

Design Alternatives

Tabletop Application

The application would exist on a multi-touch table surface. It would have an interface that allows users to sit around the table and work from all four sides of the table surface.

Pros

Puts users around a table which provides a great amount of eye-contact and interpersonal communication. We feel that this is important as in the questionnaires, our user's largest problem with collaborative work was disagreements amongst group members (see analysis of results).

From our interview data it was also shown that most collaborative diagrams are created using a turn-based system, some groups often to having one person dedicated to drawing. Seated people generally also need to stand up in order to contribute to the diagram. The table will reduce these problems.

The table's multi-touch interface that allows for pen input would provide many opportunities for solutions to the problems of flexibility and structure.

Cons

These types of tables are not widely available and very expensive [11].

Table interfaces present a directional challenge, as users are sitting at all angles of the interface and thus special care needs to be taken to ensure that everyone has access to the tools they need.

SMARTBoard-like System

We could develop a system that augmented a standard whiteboard. Although a similar system has already been developed by Smart [10] , we would work on an application that is more focused towards technical diagrams and more flexible and usable.

Pros

Comes closest to the traditional white-board which our users (especially our interview subjects, mentioned as their preferred diagramming tool). This system would provide an intuitive way to create structured and unstructured input methods.

Cons

The main problem with this technique is that it does not introduce anything new to support interpersonal skills. This technique would also not address our "should include" requirement of allowing more space for users to interact with the system.

Multiple Screens - One Room (mouse input)

We could use standard available hardware and develop a system that allows users to be in the same room on their laptops and drawing on a shared work surface. The main innovation would be an architecture that allows the users to see each other's mouse pointers and diagram edits instantaneously.

Pros

Leverages hardware that is already available to users, allows people to have their own space. Having their own space also means that the user's view is never occluded by other users.

Cons

Puts screens between users and each other. This may seriously limit eye contact and communication. It is not clear how it would be possible to augment interpersonal interaction with this method.

Does not allow for direct manipulation with the hands. Users like the flexibility of traditional methods and it would be very difficult to replicate that flexibility with a mouse input. Touch also seemed to be very important to the users we studies, which is not supported well by traditional screens.

Scanning Tool

We could develop a system that allows users to easily capture white-board and paper diagrams, automatically "cleaning them up" to eliminate transfer effect and loss of information.

Pros

Would leverage users existing method of creating diagrams. In fact, when asked to design their perfect diagramming system 4 of our users mentioned something similar to this.

Would provide the flexibility of traditional methods as well as the recording abilities of software technologies for diagramming which is one of our absolute must include features.

Cons

This method would also not focus much on interpersonal skills

Iterating over diagrams would be difficult as once the diagram becomes electronic it can no longer be converted to its original form (unless it was a paper sketch). Many interview and questionnaire respondents mentioned that they iterate over diagrams.

The algorithms required to "clean up" hand-drawn input would be very difficult to create.

Appendix A

Appendix A1.a

Questionnaire Questions

Legalities

Do you agree to the ethics consent form?

[ ] yes

[ ] no

Can we contact you for an interview in the future?

[ ] yes

[ ] no

If yes to previous question, please provide us with your name and email address.

As outlined in the consent form we promise not to share your contact information with anyone outside our group.

____

Demographic Questions

What is your gender?

[ ] Male

[ ] Female

How old are you?

[ ] 18 - 25

[ ] 26 - 35

[ ] 36 - 45

[ ] 46 - 55

[ ] 55 - 64

[ ] 65+

What is your current employment or student status?

[ ] 1st year university

[ ] 2nd year university

[ ] 3rd year university

[ ] 4th year university

[ ] Graduate student

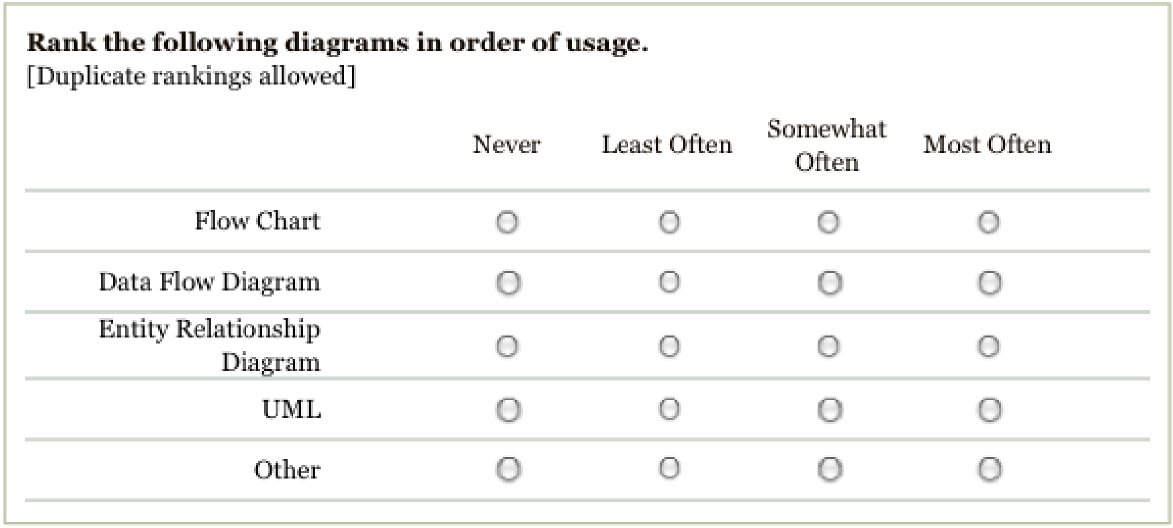
[ ] University Professor

[ ] PhD Student
[ ] working in technical industry
[ ] other (please specify) ______________________________

Diagram Questions

Below we are going to ask a few questions about diagrams. For our purposes, we are going to define diagrams as drawings that show **interactions** between different elements in a **system**.
Have you ever created a diagram?
[ ] yes
[ ] no

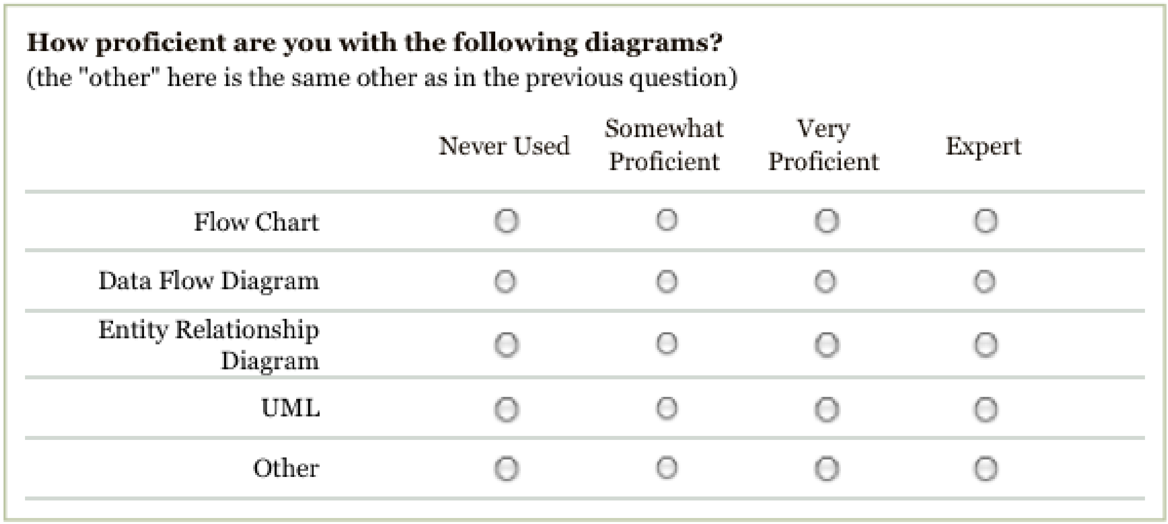
_________________________continue only if yes to previous question________________________



Rank the following diagrams in order of usage.
[Duplicate rankings allowed]

	Never	Least Often	Somewhat Often	Most Often
Flow Chart	○	○	○	○
Data Flow Diagram	○	○	○	○
Entity Relationship Diagram	○	○	○	○
UML	○	○	○	○
Other	○	○	○	○

If you ranked "other" above, please specify.
____



How proficient are you with the following diagrams?
(the "other" here is the same other as in the previous question)

	Never Used	Somewhat Proficient	Very Proficient	Expert
Flow Chart	○	○	○	○
Data Flow Diagram	○	○	○	○
Entity Relationship Diagram	○	○	○	○
UML	○	○	○	○
Other	○	○	○	○

How many software development projects have you worked on?

____

What tools have you used to create diagrams with?
[ ] Pen and paper
[ ] Whiteboard
[ ] Computer software
[ ] Other

If you answered "computer software" above, please specify what software you used.

____

If you answered "other" above, please specify.

____

When you work on diagrams, what percentage of the time do you work with others?

____

When you work on diagrams, do you prefer to work with others or alone?
[ ] Alone
[ ] With Others

Why do you prefer working alone or with others?
[In regards to the previous question]

____

What (if anything) is the most difficult part about working with others on diagrams?

____

If given infinite money, time and resources to create your perfect diagramming tool, what would it look like? Could you describe some specific features that it would include?

____

Thank you for taking the time to fill out our questionnaire.

Interview Questions

* diagram refers to software design diagram

1. When creating diagrams, do you usually work collaboratively or solo?
Why or why not?

2. When creating diagrams, do you prefer working collaboratively or solo?
Why or why not?

3. When creating diagrams with someone, do you usually work collocated (in the same place) or remotely?
Why or why not?

4. When creating diagrams with someone, do you prefer working collocated or remotely?
Why or why not?

5. What are the main problems you encounter when working collaboratively on a diagram?

6. What technologies have you used for creating diagrams (pen and paper count as a technology)? What did you like/dislike about each of them?

7. What do you dislike about the current technologies for creating diagrams?

8. What type of tool could help you work on diagrams collaboratively and in the same place?

Appendix A1.b

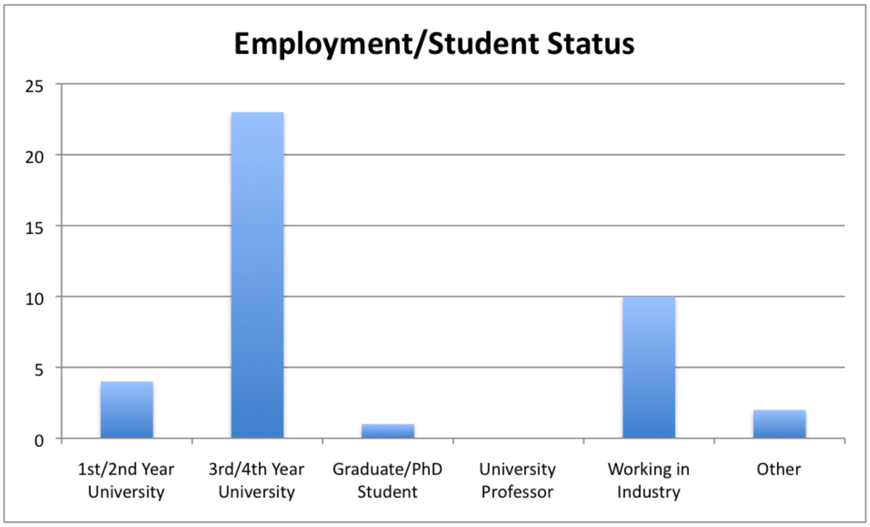


Figure 1: Employment/Student Status

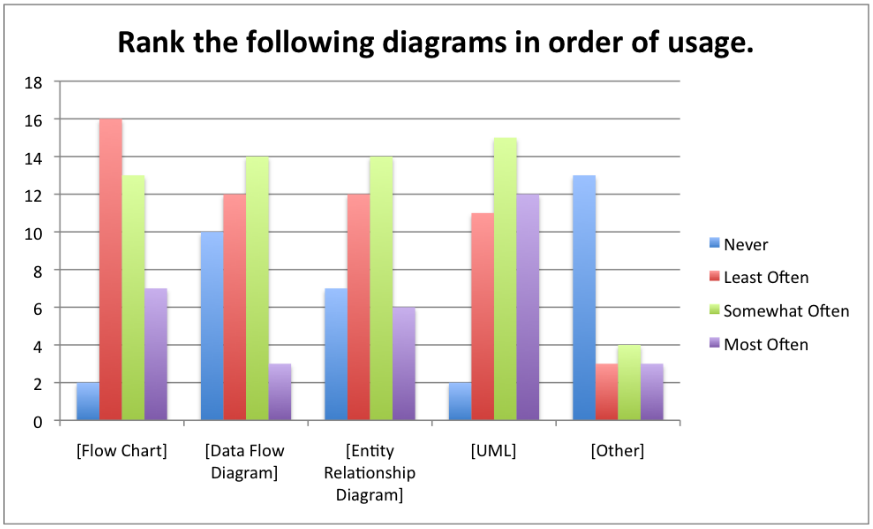


Figure 2: Diagram Usage

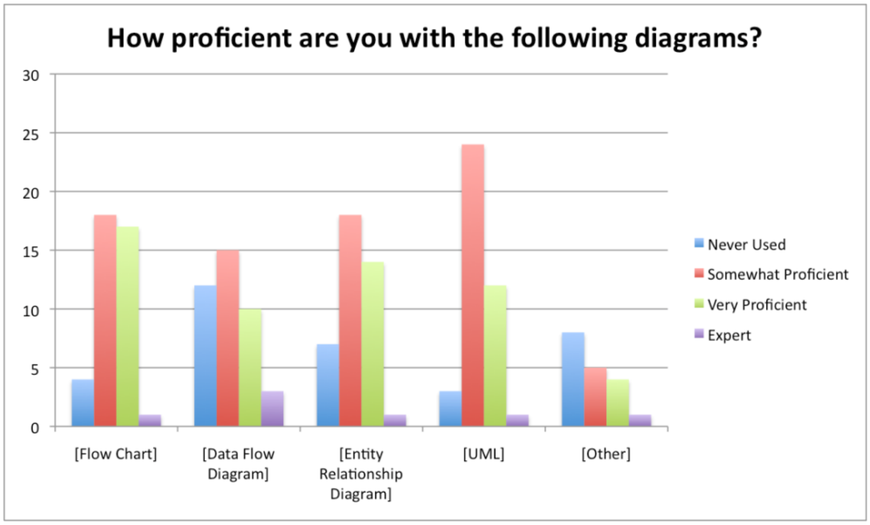


Figure 3: Diagram Proficiency

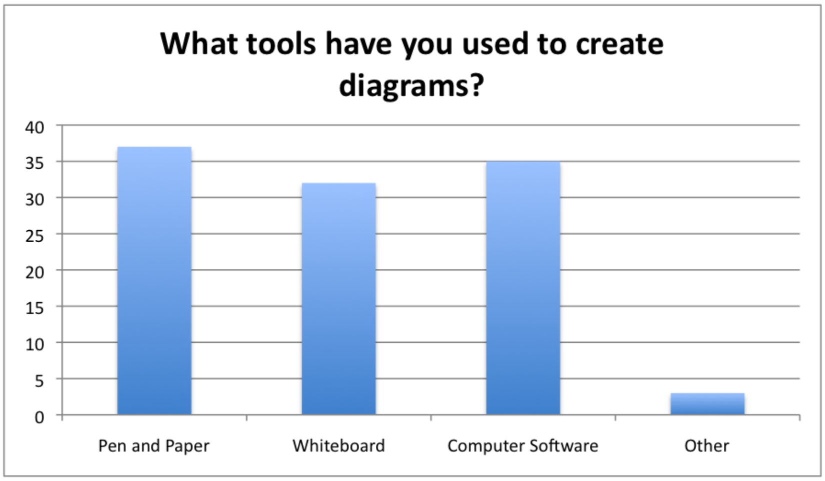


Figure 4: Diagram Tools

*People may select more than one checkbox, so percentages may add up to more than 100%.

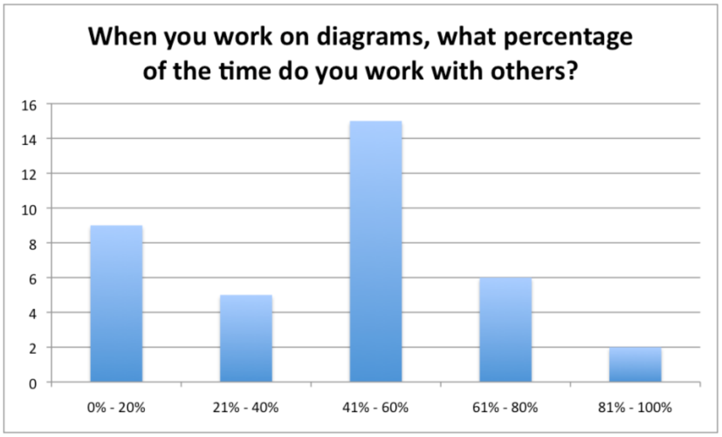


Figure 5: Collaboration Percentage (Actual Practice)

Mean: 43.9%

Median: 50%

Standard deviation: 27.3%

Outliers: none

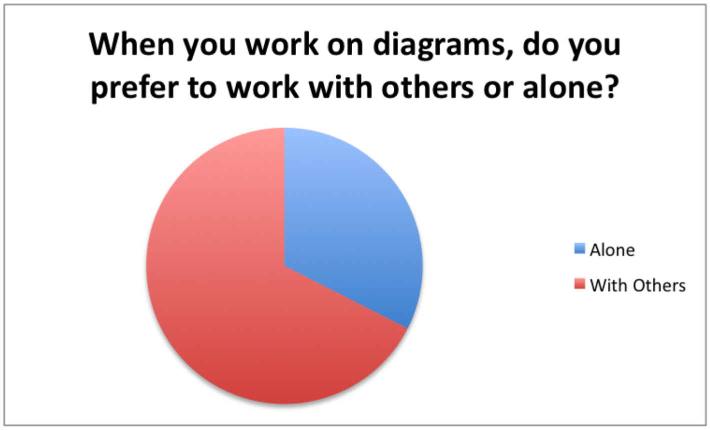


Figure 6: Collaboration Preference

68% of users *prefer* creating diagrams with others.

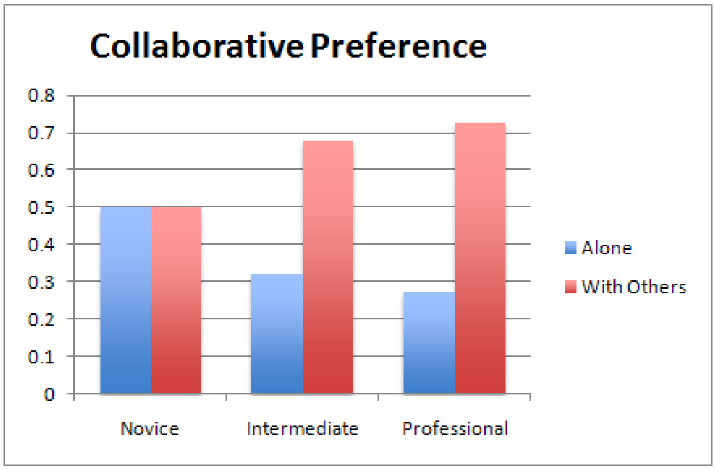


Figure 7: Collaboration Preference by Category

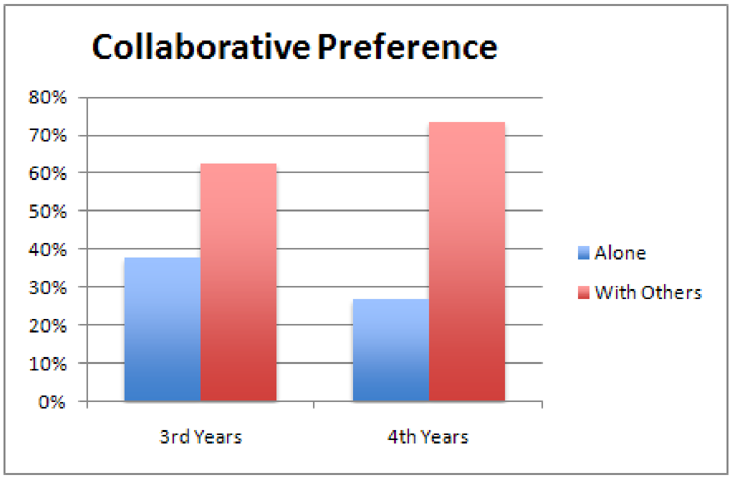


Figure 8: Collaboration Preference: 3rd Years vs. 4th Years

Appendix A2

1. Co-operative UML Design/Brainstorming

Steve J. and Bill G. work together on designing the architecture for their upcoming software product. They meet in Steve J.'s garage and begin sketching out ideas, but because they are complete opposites in their abilities and focus it is difficult to create one architecture that suits them both. So they keep coming up with more and more design alternatives. They are able to rapidly create architectures by using a UML-like diagramming approach, as they do not care about the low-level details yet—they simply want to get their ideas down and get a feel for each other's thoughts. Once they are done, they go through what they have made together and settle on one that shares their vision and can be implemented best given their varied abilities.

MODIFIED: Focused the problem to be around conflicting personalities who are trying to solve a difficult problem together and need to arrive at a unified vision. Added more details about how they are diagramming based on our user study.

2. Co-operative UML Design/Editing

Ada L., Allen T. and Grace H. are three students working together with an already-built UML-like diagram, one that only includes all the high-level details of their software project. They begin to add details to the existing architecture simply by editing the current diagram. Eventually, they discover that the current design will not suffice for their needs, and an entirely new class is required. They create the new class and add it to the diagram, changing and moving around the connections between classes as needed, all without scrapping or re-doing their previous work.

MODIFIED: changed completely from MS I; transformed into an extension of Task Example 1 in order to illustrate that diagramming is required at multiple stages in the software development lifecycle.

3. Co-operative UML Design/Different Roles

The system architect and one of the developers go over the system design together. The developer is mostly observing, and the architect is explaining the design verbally while highlighting certain aspects of the design by drawing attention to those areas of the diagram, showcasing details as she sees fit. At a certain point, the developer notices an error in the system and needs to chime in and underline the possible problem. He is able to quickly interrupt the architect by highlighting that aspect of the design and bringing it to her attention by moving the design to the appropriate location, all without having to ask for or be transferred control in order to interact with the system.

ADDED: added as an extension of the previous task examples; a few users in our questionnaire mentioned that in industry it is rare for the person implementing a system to be the one who designed it. As such an interesting power dynamic sometimes surfaces during collaborative tasks, where one user does most of the design while the other has limited input but does most of the work. We wanted a task example that showed how someone in the role of limited input could easily step in, interrupt, modify, and thus better understand and build a system if they are allowed a more open and less authoritative interface.

4. [Removed]

DELETED: While in MS I there was only marginal overlap between Task Example 4 and Task Example 1, the modifications made to Task Example 1 (and Task Example 2) make this task example redundant.

Appendix A3.a

Info

Graduate from UBC -- working in industry 4 years
Female

1. When creating diagrams, do you usually work collaboratively or solo?
Why or why not?

When I'm documenting after the fact, I generally work alone. If its part of the design, then I need to work with other people. I do this in a large shared space (ie whiteboard) -- one person starts, and people just kind of jump in. There's not really an order to it. There's a lot of iteration to it. When we're making something at work, someone will start, and someone else will jump in and tweak it -- there's a lot of erasing going on. It sometimes becomes hard to read.

2. When creating diagrams, do you prefer working collaboratively or solo?
Why or why not?

I prefer doing the first pass solo -- but when you're trying to refine the design, you work with other people. I don't enjoy designing in a vacuum.

3. When creating diagrams with someone, do you usually work collocated (in the same place) or remotely?
Why or why not?

We're generally in the same space.

4. When creating diagrams with someone, do you prefer working collocated or remotely?
Why or why not?

Absolutely prefer working collocated. I hate working remotely.

5a. What are some advantages to working collaboratively in the same space?

Everyone is on the same page. Its easier to get a point across. Its easier to communicate across cultural barriers when you're in the same space. A lot more data: facial expressions, body language, tone of voice, hand gestures, laughter.

Or even just a communication difference between people. When you have a clash of personalities, its a lot easier to overcome when you're in the same space.

5b. What are the main problems you encounter when working collaboratively (and collocated) on a diagram?

Generally we run out of space. Its also hard to capture the iterations. Especially if you're whiteboarding something. You start erasing things and moving things around. The only thing you have is the input that was originally there -- ie a separate piece of paper. You make a bunch of decision as you iterated, which get forgotten later down the road. At the end of all the iterating on the whiteboard, ie the table, the whiteboard, you only have the thing you reached an endpoint at. You don't have anything in between. Unless you're very vigilant about it yourself.

To capture all that in-between data you would need to take photos, and a lot of notes. I take photos of intermediate stages. I tend to take more photos at the end. I generally don't find this useful though -- you need photos and notes. A lot of the time a photo will suffice, but someone in the room will say "Oh well we tried that but blah blah blah ..." I think its useful later when you're trying to cut something later, and then put it back in, and people are trying to figure out why the initial decision was made.

How do you normally deal with managing the photographs -- we put them in a giant shared folder. It's not a living document. That's sort of par for the course for any kind of documentation though. Any kind of large data set -- its lost really easily, unless people are constantly adding to it. Past data that isn't current gets lost -- unless people make a point of revisiting it.

6. What technologies have you used for creating diagrams (pen and paper count as a technology)? What did you like/dislike about each of them?
pen and paper, whiteboard, post-its

What do you like about those tools?
They're quick, cheap, off-line, portable, don't need a power source, you can do it on-site, with clients, many people can work on it at once -- the biggest part is that its cheap and fast. And easy for people to collaborate. There's no barrier.

My co-workers like using this Mac software called Omnigraphle.

7. What do you dislike about the current technologies for creating diagrams?
Lack of space.

Comments on design direction:
- I think its a neat idea

Do you think it would be useful?
- I'm not completely convinced that its necessarily better than using pen and paper, because people iterate on documents anyway -- there's no point in having a super shiny version. And I think there's something to be said about having a low-tech way of collaborating. People tend to feel a little more empowered. This is really true when I'm working with clients. Very, very important when I'm working with clients.
- Cheap and fast, that's the way to get things done. I'm not convinced that a multi-touch table for doing diagrams is the best way to go. Also the intimidation factor is big, when dealing with clients.

Appendix A3.b

Info
professor / grad student at UBC
Male

This is the transcript from the interview I conducted with Emerson Murphy-Hill on February 2, 2010 -- regarding our work in CPSC 444 (Gabe)

1. When creating diagrams, do you usually work collaboratively or solo?
Why or why not?
Collaboratively. Useful for communicating, designing with someone else. For myself, its unusual that the software that I want to build is so complicated that I feel the need to diagram.

When working solo, I typically start with something small, and expand it with the features that I need (refactor etc.). In that case, I don't really need UML diagrams.

Collaboratively I use it to talk about the architecture with someone else. They're trying to explain things to me or I to them.

What's convenient about UML is that there is a structure that you agree on. You want to have something visual to latch on to.

Strict UML? *No*

2. When creating diagrams, do you prefer working collaboratively or solo?
Why or why not?

3. When creating diagrams with someone, do you usually work collocated (in the same place) or remotely?
Why or why not?
Always in the same place. I can't think of any examples where I have done it remotely.

Can't think of any examples where I have done it asynchronously.

Example of a solo usage of UML: Backtracking a bit -- as I was browsing the code, I created UML diagrams with Visio, based on an existing codebase. I wanted to have a better handle on the important classes and their relationships.

The deal was that I basically made two diagrams -- the old class hierarchy and the new one. I was trying to find correspondences in the two versions.

4. When creating diagrams with someone, do you prefer working collocated or remotely?
Why or why not?

(see above answer)

5. What are the main problems you encounter when working collaboratively on a diagram?
I typically work on a whiteboard. Its easier to erase and change things. One difficulty is documenting what you talked about. So you've got it on the whiteboard -- a temporary thing. But when the meeting is over, how do you record that. do you need to? typically my experience is the one person is giving advice, the other one is in more of an implementation role. The other person is managing, or has done this sort of thing before, or knows what it should be like. They do that, then they break apart.

You typically leave the whiteboard with the person who is doing the implementing.

I feel that I need to keep a copy of it, but you're not sure its been very useful. I don't know that I use it later.

Another problem, is that there is stuff on the whiteboard and it makes sense at the time, but when you come back later there is not enough info to see the context of it. What was preliminary, what was absolutely agreed on, etc...

For instance, there is a class sitting there -- maybe one of us thought that we agreed on something else. The diagram is too abstract. It doesn't specify things that you talked about but didn't write down.

6. What technologies have you used for creating diagrams (pen and paper count as a technology)? What did you like/dislike about each of them?
Visio and Omnigraphle. Essentially I think the problem is that they are slower than you are at inputting things versus the human at the whiteboard. Typically you have to choose the tool then choose the tool then draw it somewhere -- this takes many steps. More difficult to do things. Not quite as flexible. You can't add extra things. You are limited. Life tends to be more fluid than what the tools have to offer. For instance you can't include a UI mockup within a diagram.

7. What do you dislike about the current technologies for creating diagrams?
(see above answer)

8. What type of tool could help you work on diagrams collaboratively and in the same place?
Typically on a whiteboard. Something that would automatically capture what you talked about. Something that captured more richly than what you just wrote. Gestures, speech are all things that are missing.

Class diagrams are definitely the most common. Structure is good.

Electronic whiteboards. I used one years ago, and it was terrible. It was so low fidelity that it made it painful. You draw something on it, and the projector projects it.
I might think about what multi-touch would really buy you. What real advantages does it offer? What you could do in the software that you couldn't do in the real world.

You can try and do some image recognition, feature recognition etc. I think I have some paper references for you. Those might be interesting.

Appendix A3.c

1. When creating diagrams, do you usually work collaboratively or solo?
Why or why not?

solo project, sure, solo; depending on scope and importance of subsystem.

2. When creating diagrams, do you prefer working collaboratively or solo?
Why or why not?

3. When creating diagrams with someone, do you usually work collocated (in the same place) or remotely?
Why or why not?

collocated generally (assumption: 100% of time)

4. When creating diagrams with someone, do you prefer working collocated or remotely?
Why or why not?

assumed collocated always for collaboration

5. What are

- *people have to take turns with the pen; ordering on who has the right to talk*
- *making sure there's a common language*
- *idioms*

6. What technologies have you used for creating diagrams (pen and paper count as a technology)? What did you like/dislike about each of them?

- *omnigraffle, whiteboard, pen+paper*
- *hierarchies of formalism in being to spec; if it's for just for self or small meeting, whiteboard or paper*

7. What do you dislike about the current technologies for creating diagrams?

- *huge iPad would be awesome; hands better than mouse*

8. What type of tool could help you work on diagrams collaboratively and in the same place?

- *whiteboard apps (adobe connect for meetings)*

Appendix A3.d

1. When creating diagrams, do you usually work collaboratively or solo?

a. Why or why not?

2. When creating diagrams, do you prefer working collaboratively or solo?

a. Why or why not?

3. When creating diagrams with someone, do you usually work collocated (in the same place) or remotely?

a. Why or why not?

Yes. The workplace doesn't allow telecommuting.

4. When creating diagrams with someone, do you prefer working collocated or remotely?
a. Why or why not?
Collocated. The ability to read facial expressions and see people's reactions is quite important, and ideas flow much better when everyone's in the same room.

5. What are the main problems you encounter when working collaboratively on a diagram?

6. What technologies have you used for creating diagrams (pen and paper count as a technology)? What did you like/dislike about each of them?
Pen, paper, whiteboards.

7. What do you dislike about the current technologies for creating diagrams?
Analog tech makes it hard to erase/move items.

Pens and markers run out of ink. Paper rips, gets stepped on, and gets lost. Handwriting can be hard to read. Whiteboards don't always erase well. Agreeing on symbols can sometimes be difficult, with several representations of the same concept appearing on a single board, which can obscure commonalities between different proposals. It's not always clear where lines and arrows are actually going from and to. Rearranging what's on the paper or board already to make room for another datum is impossible.

Indeed, the biggest strength of these media are that you need to think before you write. That you never think enough before you do is their weakness. :)

8. What type of tool could help you work on diagrams collaboratively and in the same place?
Something like a big whiteboard, but with quick erase. The ability to recognize handwriting and help diagramming is also important, though formalized systems (UML, ERD) need not be implemented as, in general practice, they are rarely followed to the letter.

Full Log:

[5:14 PM] < alexandru> so i was just going to ask a few follow-up questions about the questionnaire you filled in. this is part of our user research for the advanced HCI course I mentioned, we're just trying to get a sense of how people do diagrams (when it comes to sofeng. related tasks)

[5:15 PM] <@bmp> Right.

[5:16 PM] <@bmp> So what follow-ups you got for me?

[5:16 PM] < alexandru> you mentioned you mostly do collaborative diagramming. is this generally collocated as well?

[5:16 PM] <@bmp> Yes. We don't even allow employees to work off-site.

[5:17 PM] < alexandru> would you prefer it remotely, do you think?

[5:18 PM] <@bmp> No. When doing collaborative work, facial expressions, the ability to point at things, sketch quickly using your hands...these all end up surprisingly important.

[5:18 PM] <@bmp> I *can* work in online collaboration--like what we're doing for this project, in fact--but I prefer in-person if possible.

[5:19 PM] < alexandru> what kinds of problems, if any, do you generally encounter when doing collab diagrams? (i.e. are people fighting for the pen, is the medium less-than-adequate for some other reasons)

[5:21 PM] <@bmp> Noting that we keep these meetings to five people or fewer, with three probably being the norm:

[5:21 PM] <@bmp> I don't think I've had a problem with people fighting for the pen. Ever.

[5:21 PM] <@bmp> The one thing that does get frustrating is having to erase the board.

[5:21 PM] <@bmp> I sometimes wish I could just put the board off to the side and throw up a new one.

[5:21 PM] <@bmp> I know some people make digital boards that effectively let you do that, but I've never used one.

[5:22 PM] <@bmp> On the other hand, having to copy the board down onto paper has never really bothered me.

[5:22 PM] <@bmp> Having to copy it makes sure I actually understand what's ON the board.

[5:22 PM] <@bmp> Too often, when I simply consult the board again later, everyone's forgotten what a little piece of it means.

[5:22 PM] <@bmp> The act of copying usually helps me realize that something's not actually clear.

[5:22 PM] < alexandru> do you generally have someone do the drawing, while others push ideas back and forth, or does everyone contribute directly on the whiteboard

[5:22 PM] < alexandru> ?

[5:22 PM] <@bmp> I don't know that my colleagues feel the same way, though--and the inability to email a board is in fact annoying.

[5:23 PM] <@bmp> People take turns.

[5:23 PM] <@bmp> Only time we ever have a dedicated author is when we're brainstorming.

[5:24 PM] < alexandru> besides erasing the whiteboard, what other shortcomings have bothered you, in any amount, about the tools you've used? (we can include pen and paper or software as well here)

[5:26 PM] <@bmp> Pens and markers run out of ink. Paper rips, gets stepped on, and gets lost. Handwriting can be hard to read. Whiteboards don't always erase well. Agreeing on symbols can sometimes be difficult, with several representations of the same concept appearing on a single board, which can obscure commonalities between different proposals. It's not always clear where lines and arrows are actually going from and to.

[5:26 PM] <@bmp> Rearranging what's on the paper or board already to make room for another datum is impossible.

[5:27 PM] <@bmp> Indeed, the biggest strength of these media are that you need to think before you write. That you never think enough before you do is their weakness. :)

[5:28 PM] < alexandru> speaking of symbols, do you usually strive for accurate-to-spec diagramming (i.e. UML, ERD), given the task, or is there generally some freedom?

[5:28 PM] < alexandru> (well put)

[5:28 PM] <@bmp> Hell no. There's freedom.

[5:28 PM] <@bmp> Certain symbols are almost ridiculously flexible; I suspect most people at Fog Creek, given no context, would draw the same symbol for hard disks, databases, and servers, for example.

[5:28 PM] <@bmp> (A cylinder.)

[5:29 PM] <@bmp> (Or a box. Depends on who's doing it.)

[5:29 PM] <@bmp> I get that the advantage of having a standard system is that you always do know what the symbols mean. I think the disadvantage is probably that you find yourself locked into thinking of only those symbols.

[5:29 PM] <@bmp> While I hate UML and ERD (and am happy to explain why), I'm not against formalized symbols in general.

[5:30 PM] <@bmp> I think part of why I've never felt any need to argue it at work is that what we're diagramming in any given meeting rarely has much to do with what we did previously.

[5:30 PM] <@bmp> Is it cash flow? Activation sequence? Communication lines? Failsafe sequences?

Network topology? Data flow?

[5:31 PM] <@bmp> These all clearly require different symbols, but I *think* I've rarely needed to diagram the same general concept more than once or twice, so formalizing just hasn't struck me as wortwhile.

[5:31 PM] <@bmp> </answer>

[5:31 PM] < alexandru> and finally, do you have any thoughts on how your ideal tool would be like? i sense it would share some similarities with a whiteboard, minus the already-mentioned short-comings. any 'dream features'? (there's no platform constraint)

[5:32 PM] <@bmp> I'd love a board that improved what I drew and wrote to look better.

[5:32 PM] <@bmp> Think the UPS ads.

[5:32 PM] <@bmp> If you have those in Canadia.

[5:32 PM] <@bmp> Apple's Newton also tried and failed to do that concept.

[5:32 PM] <@bmp> That, combined with the ability to rearrange what's already on the board, would be wonderful.

[5:33 PM] <@bmp> The things I really like are the tactility and immediacy of the board. Having increased ability to manipulate after having drawn without erasing and redrawing multiple times would be a logical and direct improvement, properly executed.

[5:33 PM] <@bmp> Having the ability to have a separate, more visible copy of the board might also be useful, although that's not usually a problem when there's 3-5 people in the room.

[5:33 PM] < alexandru> thank you again for your time with both the questionnaire and this follow-up, the information is immensely valuable for our research

[5:34 PM] <@bmp> No problem!

[5:34 PM] < alexandru> i'll make sure to keep you updated in case we come up with something worth while ;)

[5:34 PM] <@bmp> Cool. :)

[5:34 PM] <@bmp> Psyched to hear of it.

[5:35 PM] <@bmp> All right, I'm gonna start doing some of the other stuff I've got to do tonight, unless you still need me.

[5:35 PM] <@bmp> Take care.

[5:36 PM] < alexandru> thanks again, cheers!

Appendix A4

Confirmation of Adherence to the CS444 Ethics Protocol

This form must be completed by each team member and submitted with your portfolio.
My signature below indicates that I have read the CS444/CS544 Ethics Protocol and have abided by it throughout my CS444 class project.

Name	Signature	Student Number	Date
Alex Totolici		69229045	Feb. 8, 2010
André Malan		58353061	Feb. 8, 2010

Gabe Silk		93911014	Feb. 8, 2010
Jré Sarenac		17191081	Feb. 8, 2010
Piam Kiarostami		94358991	Feb. 8, 2010

Appendix C

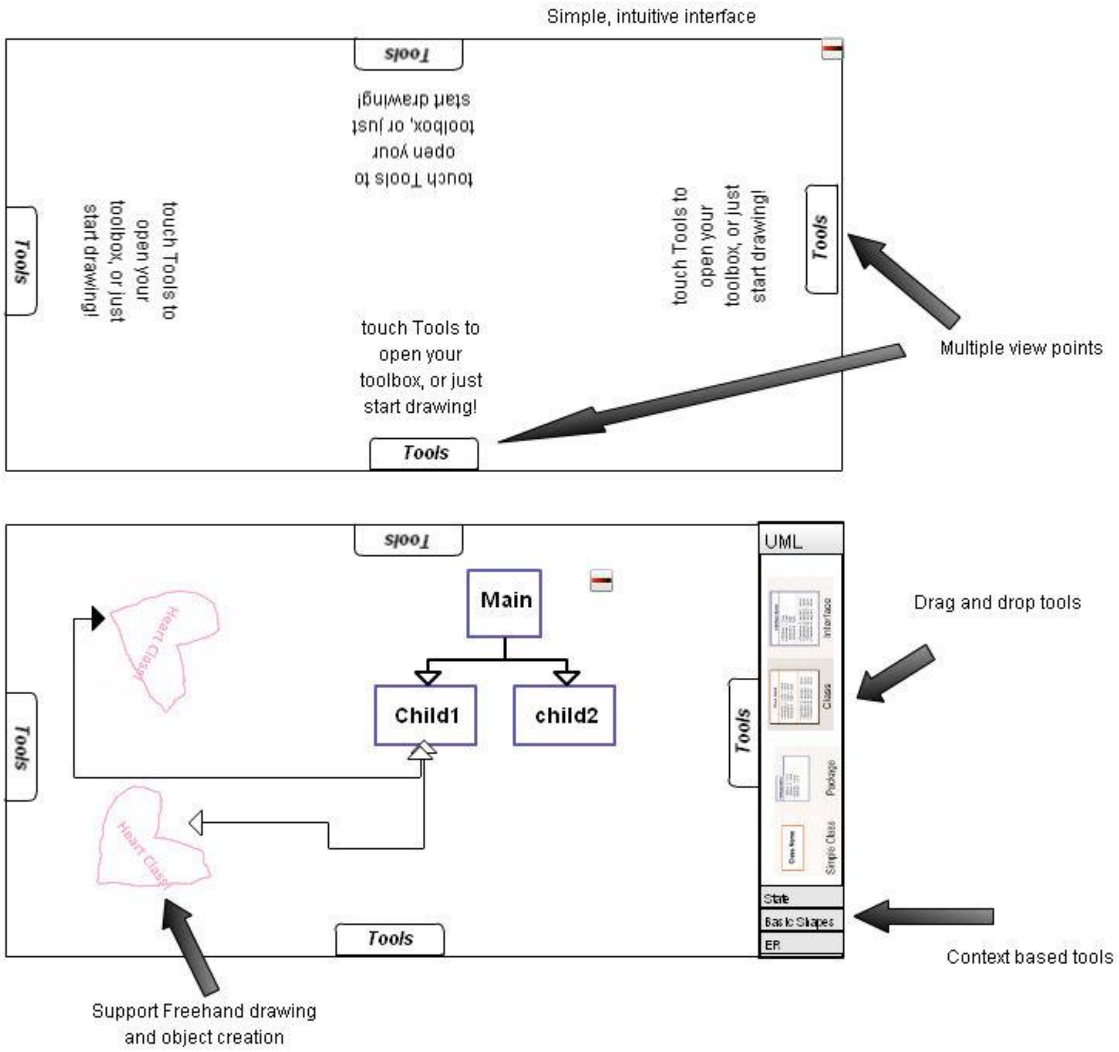

Simple, intuitive interface
Tools
touch Tools to open your toolbox, or just start drawing!
Tools
touch Tools to open your toolbox, or just start drawing!
Tools
touch Tools to open your toolbox, or just start drawing!
Tools
touch Tools to open your toolbox, or just start drawing!
Tools
Multiple view points
Tools
UML
Main
Child1
child2
Heart Class
Heart Class
Interface
Class
Package
Simple Class
State
Basic Shapes
ER
Tools
Tools
Tools
Drag and drop tools
Context based tools
Support Freehand drawing and object creation

Figure 1: Table Based Environment Sketch

Our sketch of a table-based interface. The first frame is the initial view and the second frame is the one after users have been working on the diagram.

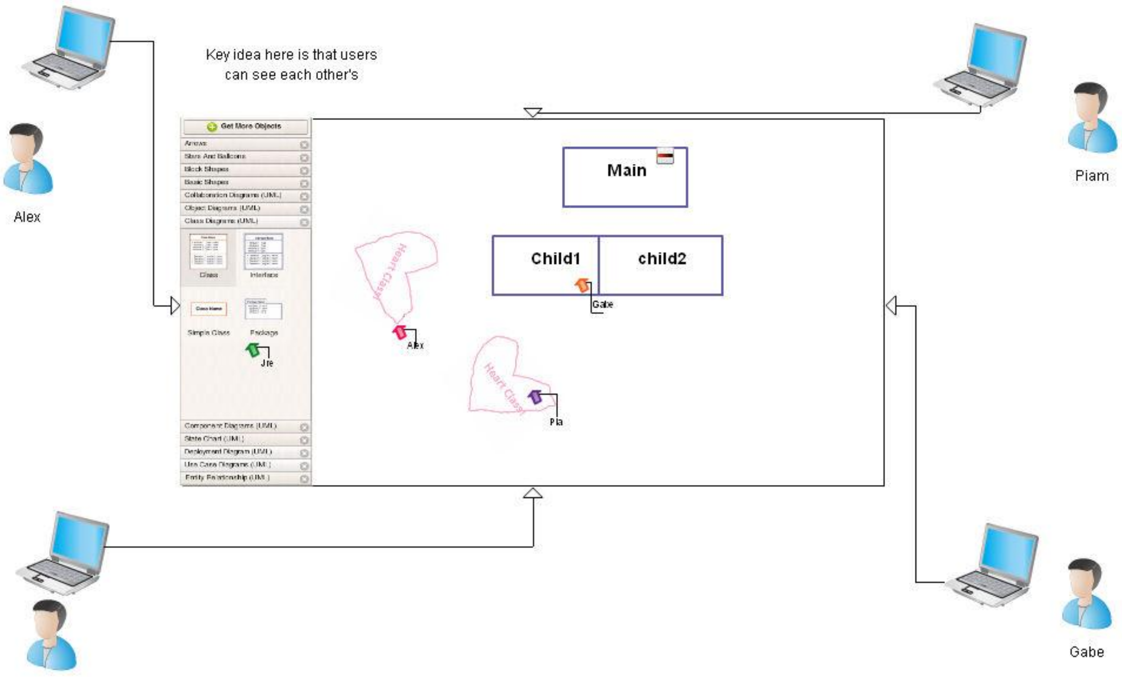


Figure 2: Multiple Computers Sketch

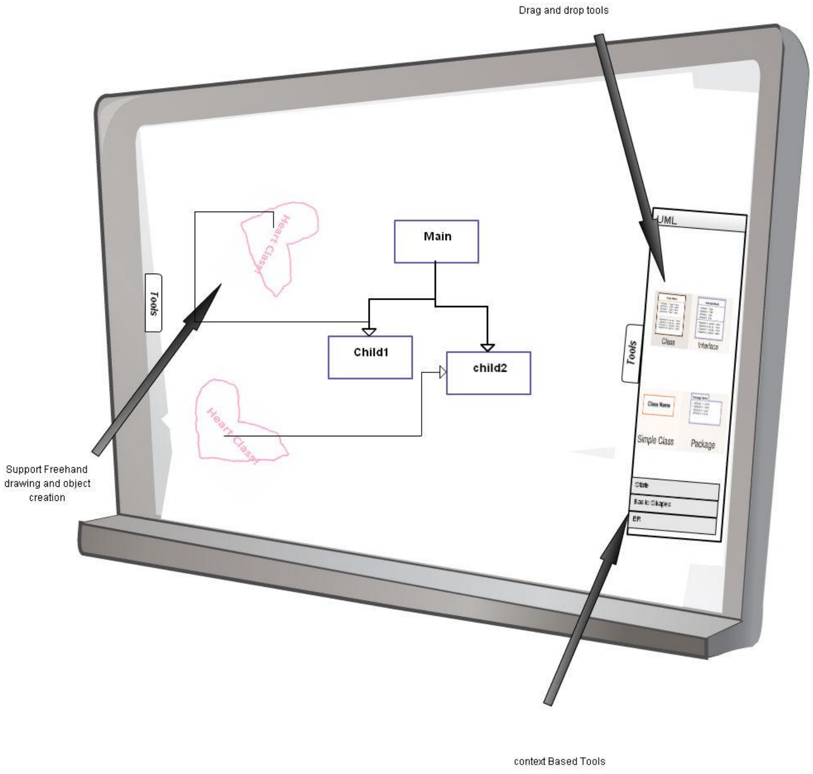


Figure 3: Smart Board interface

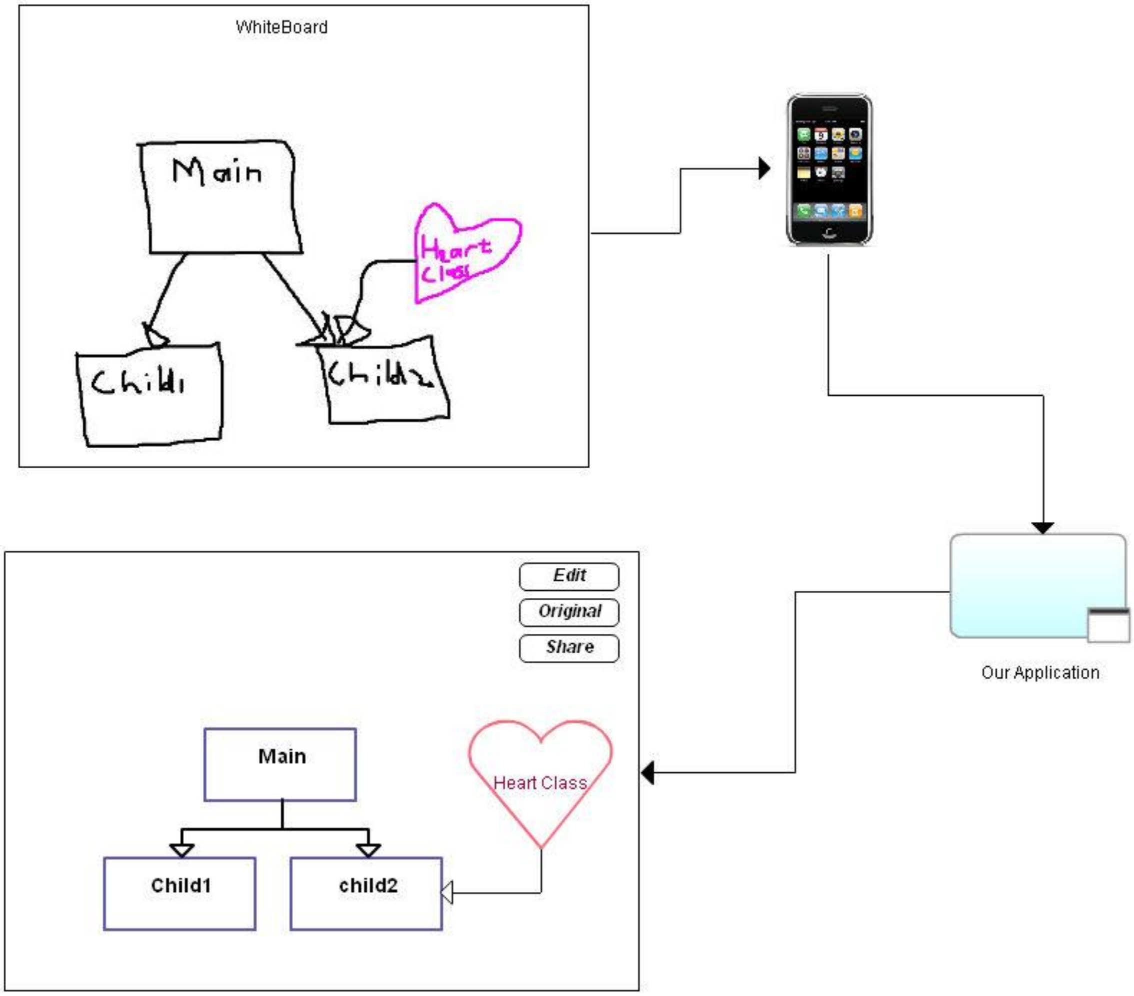


Figure 4: Scanner System