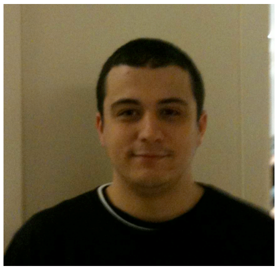
CPSC 444

Team χ^2 - Milestone 1

Collocated, Collaborative Diagramming



Alex Totolici
69229045

Andre Malan
58353061

Gabe Silk
93911014

Jre Sarenac
17191081

Piam Kiarostami
94358991

Problem Description

We intend to design a system and interaction methods that support collocated, collaborative diagramming. There are many design challenges that need to be addressed, of which we will focus on the following: flexible user arrangements, simultaneous user interactions and transitions between personal/group work. We will analyze current systems used for collaborative diagramming and create a novel interface that supports the needs of collocated users working together in real-time on diagramming tasks of varying complexity.

Our primary stakeholders include anyone needing to work together to create diagrams of varying complexity. This includes students, professors, software designers, project managers, CEOs, or the average "Joe" wanting to brainstorm a layout of his living room with his roommates. Our ideal vision is to implement such a system on a multi-touch interface.

Task Examples

Task Example 1 : Co-operative Mind-mapping

Two classmates work side-by-side to create a medium-sized *mind map* [fig. 5] to brainstorm for their upcoming CPSC 344 project. They need to quickly lay down their ideas and they are interested in coming up with as many concepts as possible, so that they have plenty to choose from.

Task Example 2 : Individually Designing a Flow Diagram

A student creates a large flow diagram for her class. The diagram has many nodes and a complicated structure. She saves it so she can later show it to her professor. The professor will need to inspect different subsections of the diagram to make sure they are accurate.

Task Example 3 : Co-operative Flow Diagram

Three biology students need to prepare a flow diagram for their class presentation. Their process includes a lot of adding and removing of map nodes, as they try to decide what information they want to present during the presentation.

Task Example 4 : Co-operative Diagram Creation

The two lead software architects of StartUp Co. are using diagramming software to design the infrastructure needed by their web-app. They have a number of different components that need to be laid out (servers, databases, firewalls etc) and they would like to switch (save/load) between multiple variants of their diagram.

Analysis

Collaborative diagramming involves the creation of a shared knowledge base through synchronous collaboration. As such, it is a subset of computer-supported collaborative learning (CSCL), which is "based on the idea that computer applications can be used as scaffolding and can support socio-cognitive processes for knowledge sharing and knowledge building" (Scardamalia and Bereiter, 1994; Paavola et al., 2002).

The current tools for collaborative diagramming fall into two categories: *physical methods* (including pen-and-paper, whiteboards, chalk, sticky notes etc.), and *digital methods* (including software implemented on various platforms).

Physical systems suffer from physical size constraints, where an overview of a diagram is hard to achieve if the diagram requires fine detail or is quite large. Quickly shuffling large numbers of links and nodes is difficult, and repeatedly having to do these operations either leads to numerous iterations until the user(s) get it right, or a product with poor readability that is not easily understood. The advantages of physical systems include easy of collaboration, free expression, and a minimal learning curve.

These issues are addressed, to some degree, by computerized systems that exist today. Desktop applications (such as OmniGraffle [1], Visio [2] and Creately [3]) simplify diagramming significantly. The advantages of digital systems are numerous: physical constraints are eliminated, rapid iteration is made easy, and the user gains the ability to easily distribute her work, save it permanently, and integrate it with other documents. However, collaboration is difficult when the device for which these applications is meant for is the PC. It is often the case that initial, rough sketches are made using pen-and-paper, where all the users can collaborate, and then these sketches are moved by one user to the computer, where details can be adjusted with the aid of software. But this process is slow and error-prone, and relies on duplication when the diagrams are moved from paper to software. We are envisioning a system that combines the relative ease-of-use and fluidity of paper with the editing capabilities of software, in a collaborative, collocated environment.

Task Example Analysis

Task Example 1 : Co-operative Mind-mapping

The two students use pen-and-paper. Their ideas come too quickly for either of them to have time and use their respective diagramming software. The computer approach is also limited in usefulness for this task, as it doesn't make synchronous input easy. The end-result is a pile of notes and scribbles that have no structure or chronologic organizations.

Task Example 2 : Individually Designing a Flow Diagram

The student uses desktop software to create her very large flow diagram. The professor has some difficulty navigating the diagram, as zooming and scrolling make it slow to quickly look at the nodes he is most interested in.

Task Example 3 : Co-operative Flow Diagram

The students use a whiteboard to list their ideas. As they decide on how concepts are linked together, their map becomes messy: concepts are drawn too close together, lines intersect and disappear under node bubbles, and parts of the map get erased accidentally. To finish off, the students take pictures of the resulting map and task someone to copy it into software.

Task Example 4 : Co-operative Diagram Creation

The two begin drawing on whiteboards in a meeting room, but they quickly run out of space to store all the variations of their design. Redrawing the same components over and over again is also tedious. They continue the design process with the aid of their computers, emailing back and forth the files representing the diagrams.

Design Direction

We are proposing a collocated multi-touch interface for collaborative diagram creation. There are already excellent and usable *distributed* collaborative diagramming programs available. One example is Creately [3], which is "designed for ease of use and collaboration" [fig. 1]. Although adequate for many tasks, Creately and other distributed diagramming apps provide limited communication mechanisms. In a study by Komis et al. [4], a distributed collaborative diagramming system was developed using a text messaging system for the communication channel. The researchers' findings suggested that the software, while it supported group thinking, was limited by the expressive power of the text-based communication channel:

> *While the shared Activity Space played an important role in the study, the textual communication tool has been also used effectively despite its limitations. One evident limitation in such setting was the lack of deictic power through which gestures in the shared Activity Space will be possible by the partners. So a suggestion emerging from this study is the design of adequate tools that combine multi-modal communicative acts in a single medium, improving the expressive power of the tool.* [4]

We believe that a collaborative, multi-touch diagramming system would solve these problems by allowing users to collaborate in the same physical space, allowing them to use natural methods of communication, including speech, gestures, facial expressions, and body language. We will experiment with some of the design decisions taken by Creately -- such as simple drag-and-drop items, fluid, in-context menus [fig. 2], snap-to connections between objects [fig. 3] -- and refine them through user studies to accommodate the collocated, synchronous aspects of the multi-touch paradigm.

We will be primarily developing using an SDK, and multiple mice in order to simulate the multi-touch interface. We are confident in our ability to work under these conditions as we have met with and discussed our project with someone who developed under similar constraints and was able to produce prize-winning software.

Appendix A : References

[1]: http://www.omnigroup.com/applications/OmniGraffle/ "OmniGraffle"

[2]: http://office.microsoft.com/en-us/visio/default.aspx "Visio"

[3]: http://creately.com/ "Creately"

[4]: Komis et al., in press. Komis, V., Avouris, N., & Fidas, C. Computer-supported collaborative concept mapping: study of interaction. *Education and Information Technologies*, *7*(2) 169–188.

Appendix B : Diagrams

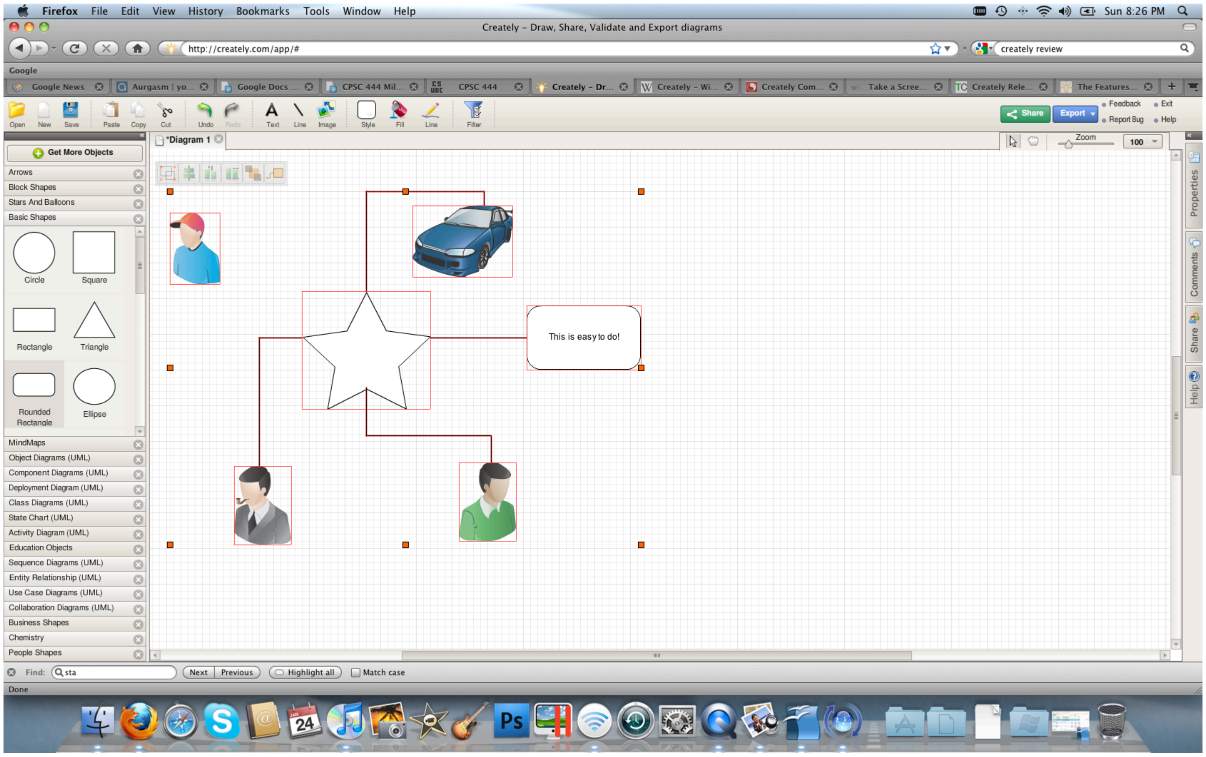


Figure 1 : Creately's Diagramming Application

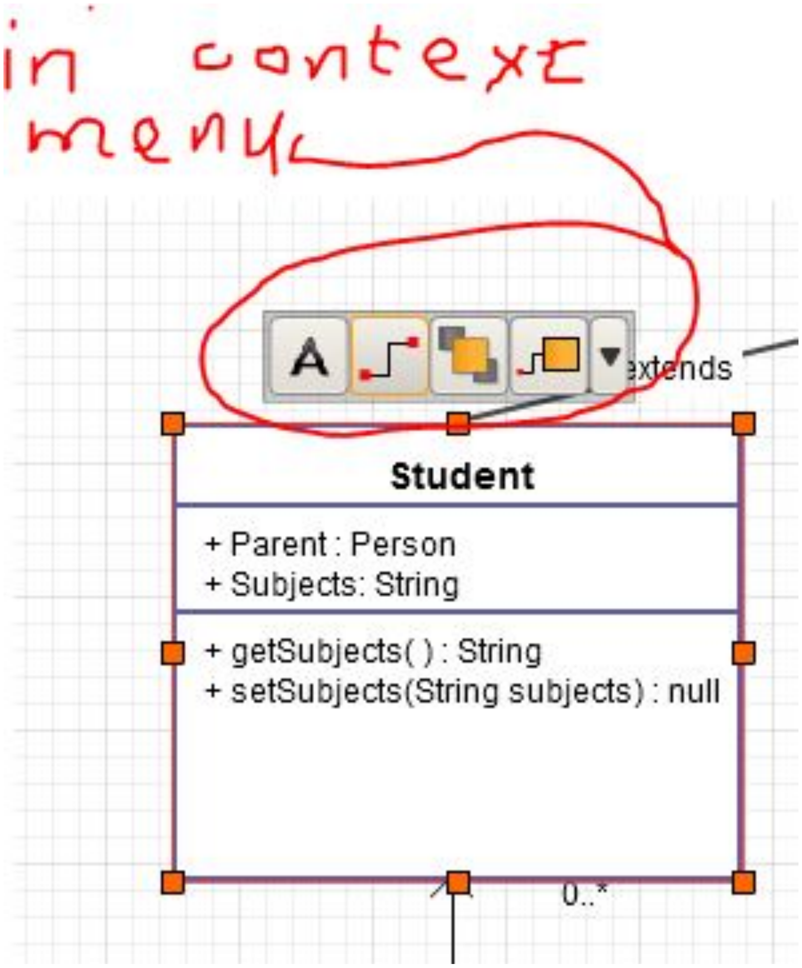


Figure 2 : In Context Menus

In context menus appear when you hover over an object. These allow users to know what they can do with an object and the experience can be extended very nicely for multi-touch.

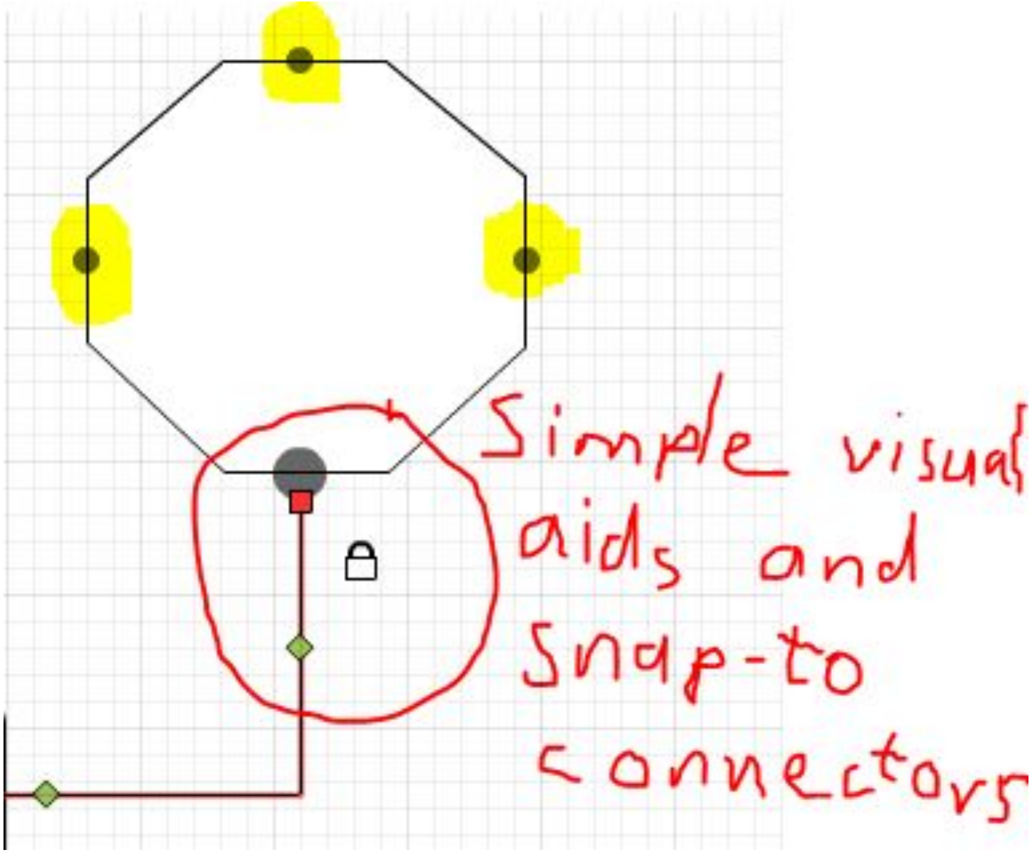
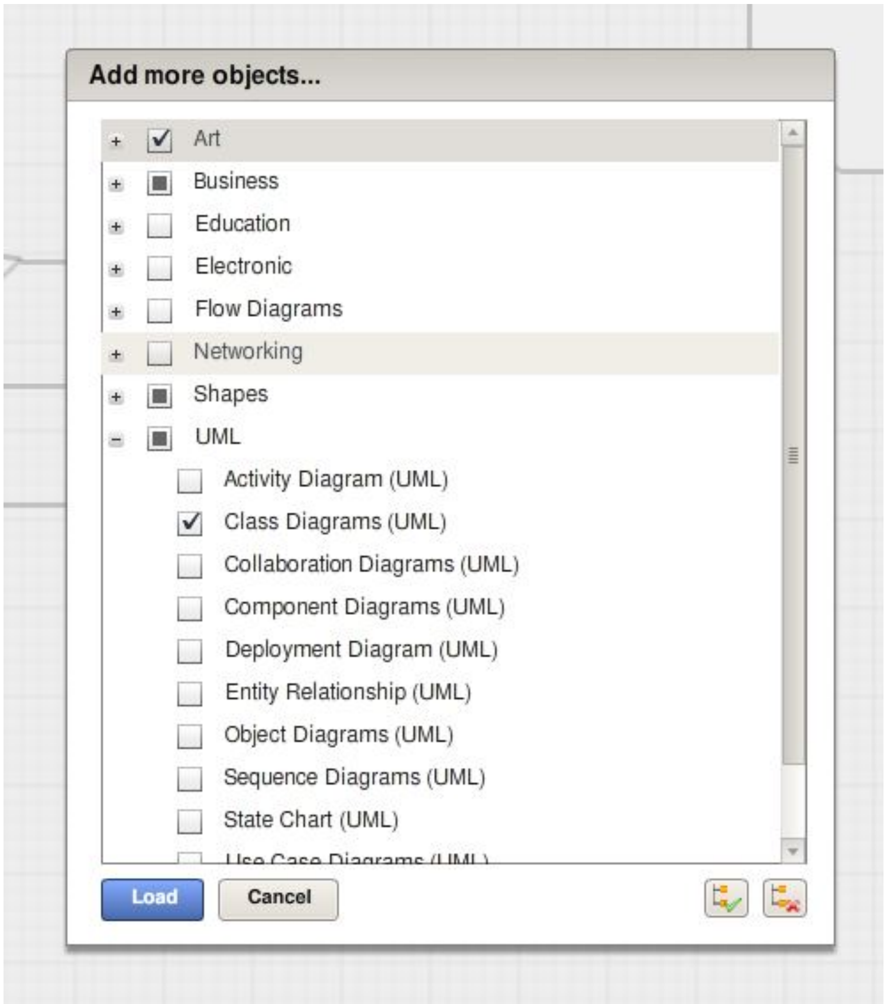


Figure 3 : Snap-to Connectors and Visual Aids

When a connector is brought close to an object visual signals automatically pop up to show how the object can be connected to. A large landing area and indication that the objects will connect ensure that users do not think that they have connected items when in fact they have not (a very common user error in diagramming applications).



Add more objects...

- + ☑ Art
- + ▣ Business
- + □ Education
- + □ Electronic
- + □ Flow Diagrams
- + □ Networking
- + ▣ Shapes
- − ▣ UML
  - □ Activity Diagram (UML)
  - ☑ Class Diagrams (UML)
  - □ Collaboration Diagrams (UML)
  - □ Component Diagrams (UML)
  - □ Deployment Diagram (UML)
  - □ Entity Relationship (UML)
  - □ Object Diagrams (UML)
  - □ Sequence Diagrams (UML)
  - □ State Chart (UML)
  - □ Use Case Diagrams (UML)

Figure 4 : Widget and Object Addition

Creately has extensive libraries of objects that users can turn off and on in order to make sure that only the tools that they need are available to them. There are also object presents based on the type of diagram that is going to be created.

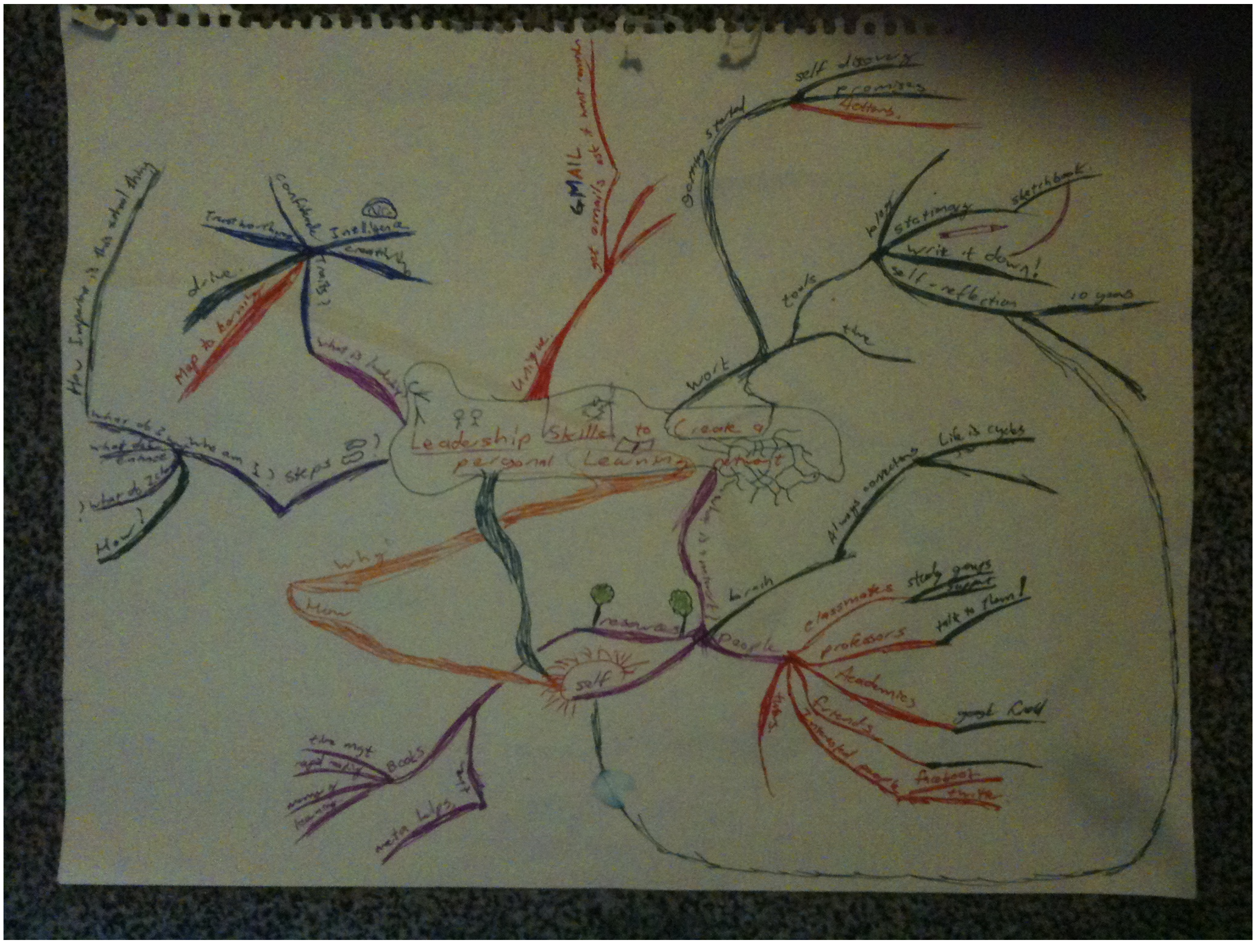


Figure 5 : Mind Map

An example of a hand-drawn mind map. Hand-drawn mind maps tend to become messy and cramped as soon as there is a lot of information to be captured. Mistakes are also really costly with hand-drawn mind maps.

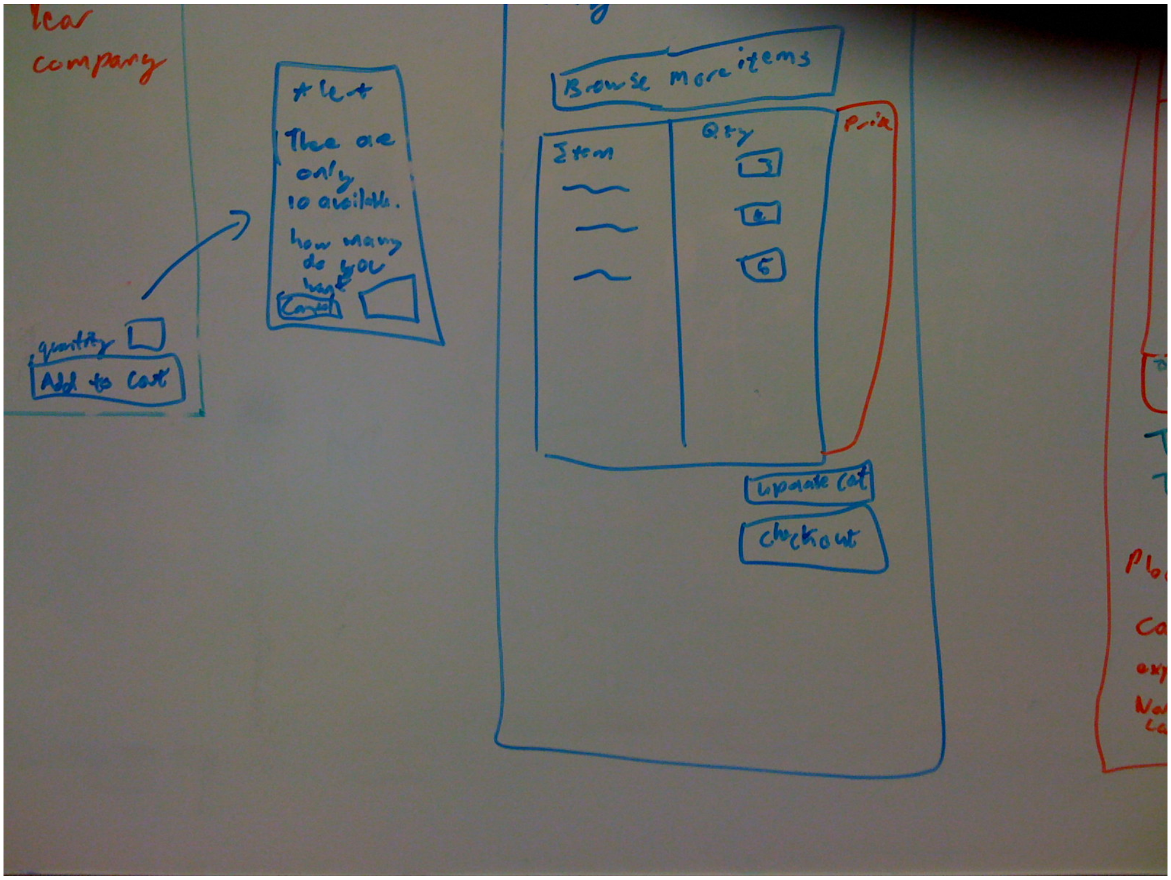


Figure 6 : Whiteboard Diagrams

Whiteboards are useful for collaborative design, but redrawing elements can be very tedious. Capturing the information in a usable format after the fact is also very difficult and information may be lost due to accidental erasure.