CPSC 444
Team χ^2 - Milestone III

Collocated, Collaborative Diagramming

Alex Totolici
69229045

André Malan
58353061

Gabe Silk
93911014

Jré Sarenac
17191081

Piam Kiarostami
94358991

March 8, 2010

[A] Low-Fidelity Prototype

Overview

The low fidelity prototype included all the basic elements of our design, including the diagram builder screen, where most of our users' collaborative tasks would take place, as well as the taskbars and options menus. It is intuitive and can be used with minimal instruction. The prototype is supports task examples 1-3, as described in the Milestone II report (also available in appendix A1). For a task breakdown of each task example, refer to appendix A2. For photographs of the prototype, refer to appendix A3.

Instructions

Because we are designing for a multi-touch surface, many of the actions available to our users are gestural, consisting of combinations of touch-points and movement on the screen. The user of our prototype must understand what actions are available and in which states. Therefore, **users should refer to table P1 while they are using the low-fidelity prototype.** This is a full listing of the gestural actions available in any given state, which can be seen below.

For the purposes of this prototype, we have divided the diagram elements into two categories: "UML box" and "Line". These two types of object behave differently in terms of creating, editing, moving, and deleting. Users must know that **lines snap to existing objects and can be dragged onto secondary objects to form a connection**, whereas **UML boxes are independent of each other, and can be dragged into any position**.

Table P1: Available actions

	Action	**Gesture**
A1	*Drag and drop (UML box)*	Press and drag a widget from the tool menu onto the canvas area
A2	*Drag and drop (Line)*	Press and drag onto the "from" class, then click on the "to" class
A3	*Create new UML box*	Single tap the UML box icon from the tool menu
A4	*Rotate diagram*	Rotate the entire diagram by using two fingers and moving them in the same direction about a circle (exactly the same as rotation on a Mac)
A5	*Move object*	Touch with one or more fingers and move the selected object(s)
A6	*Select objects within region*	Touch with one finger and drag out an arbitrary closed perimeter around the objects to be included in the selection. After the perimeter is connected, all objects within are selected.
A7	*Select/focus single object*	Single-tap on an object to both select and give it focus
A8	*Deselect*	Click outside on any empty space
A9	*Replace Object*	Drag and drop the new item onto the old item

Walkthrough Report

The walkthrough report was conducted internally, using the low-fidelity prototype. We brainstormed our task analyses as a group and conducted the cognitive walkthrough individually. We merged our key results and compiled them into the walkthrough report, which can be seen below.

	Results
Task Example 1 (see Appendix A1.a)	**When saving, opening, or creating a new diagram, users may be confused by the use of ellipses as the label for the file menu**. In keeping with traditional computer-based interfaces, the adoption of a 'File' label may allow for better discovery. **When opening a diagram, there may be confusion between the revisions that a particular file may have and the file browser** (which lists all the diagrams the tool knows about). Further testing is needed to assess user tendencies. For example, do users always want the latest revision during the initial open? Should the open file dialog always present a way to select specific revisions, instead of defaulting to the latest one?
Task Example 2 (see Appendix A1.b)	**Users attempting to select and move a group of objects may be unable to discover exactly how to move them together, keeping the group structure intact** (see Table P1, action A6). We intend to provide a tool button that users can explicitly click on to enable selection mode, at which time the application will also display a hint that the touch-drawing technique is also available.
Task Example 3 (see Appendix A1.c)	**Users attempting to select a certain object may be confused by the distinction between focusing and selecting an object** (see Table P1, action A7). We expect to leverage the transfer effect from computer-based tools, however, where clicking objects also gives them focus (makes them editable and modifies context-sensitive menus). We will also highlight/focus all newly created items automatically in order to make the distinction more intuitive.

Appendix A

[A1] Task Examples

[A1.a] Task Example 1

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.

[A1.b] Task Example 2

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.

[A1.c] Task Example 3

A system architect and one of his 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.

[A1] Task Breakdown of Task Examples

Walkthrough Questions

- Q1: Will the user try to achieve the effect that the subtask has?
- Q2: Will the user notice that the correct action is available?
- Q3: Will the user understand that the wanted subtask can be achieved by the action?

- Q4: Does the user get feedback?

Task Example 1
subtask: create new (blank) diagram
  q3: no. ellipses for the label are confusing. better to use a File label
subtask: add a box (to represent a class)
  all: yes
subtask: enter the name of the class
  all: yes
subtask: repeat 5x
  all: yes
subtask: add an arrow
  all: yes
subtask: save/checkpoint
  q3: no. ellipses for label are confusing, better to use File as a label
st: repeat x 2
subtask: open previous revisions and choose one
  q3: no, ellipses confusing, use File; file browser will be familiar. orientation may be a problem (do diagonally)

Task Example 2
subtask: open existing diagram
  q3: no. ellipses for the label are confusing. better to use a File label
subtask: change arrow types for existing arrows
  q2: no.
  q3: no, see q2
subtask: add a box (to represent a class)
  all: yes
subtask: enter the name of the class
  all: yes
subtask: select a group of classes to move
  q2: no
  q3: no
  resolution: selection / grouping not obvious
subtask: move existing classes
  all: yes
subtask: change connections between classes
  all: yes

Task Example 3
subtask: open existing diagram
  q1: yes
  q2: yes
  q3: no. ellipses for the label are confusing. better to use a File label
  q4: yes
subtask: highlight an element within the diagram
  q1: yes
  q2: possibly not -- they might not know that touching on an object highlights it

q3: yes
q4: yes

subtask: highlight a new element within the diagram

q1: yes
q2: yes
q3: yes
q4: yes

subtask: move the entire diagram

q1: yes
q2: possibly not -- the action is gestural. They might not know it is available.
q3: yes
q4: yes

[A3] Photos of Low-Fidelity Prototype

[A3.a]

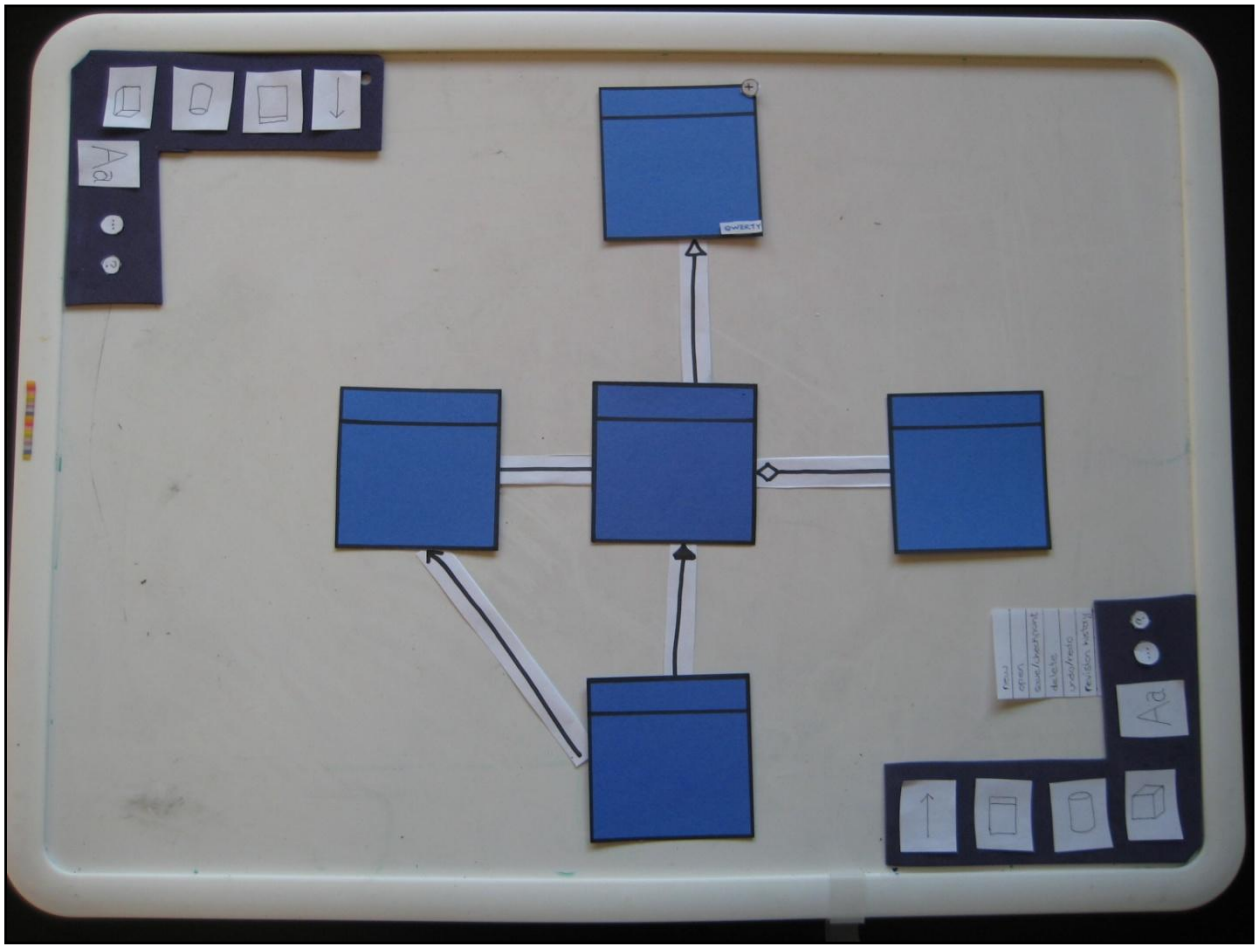

Figure A3.a: Five UML boxes connected by lines of various types. The menu panels are replicated in the corners to allow all users easy access to tools/options.

[A3.b]

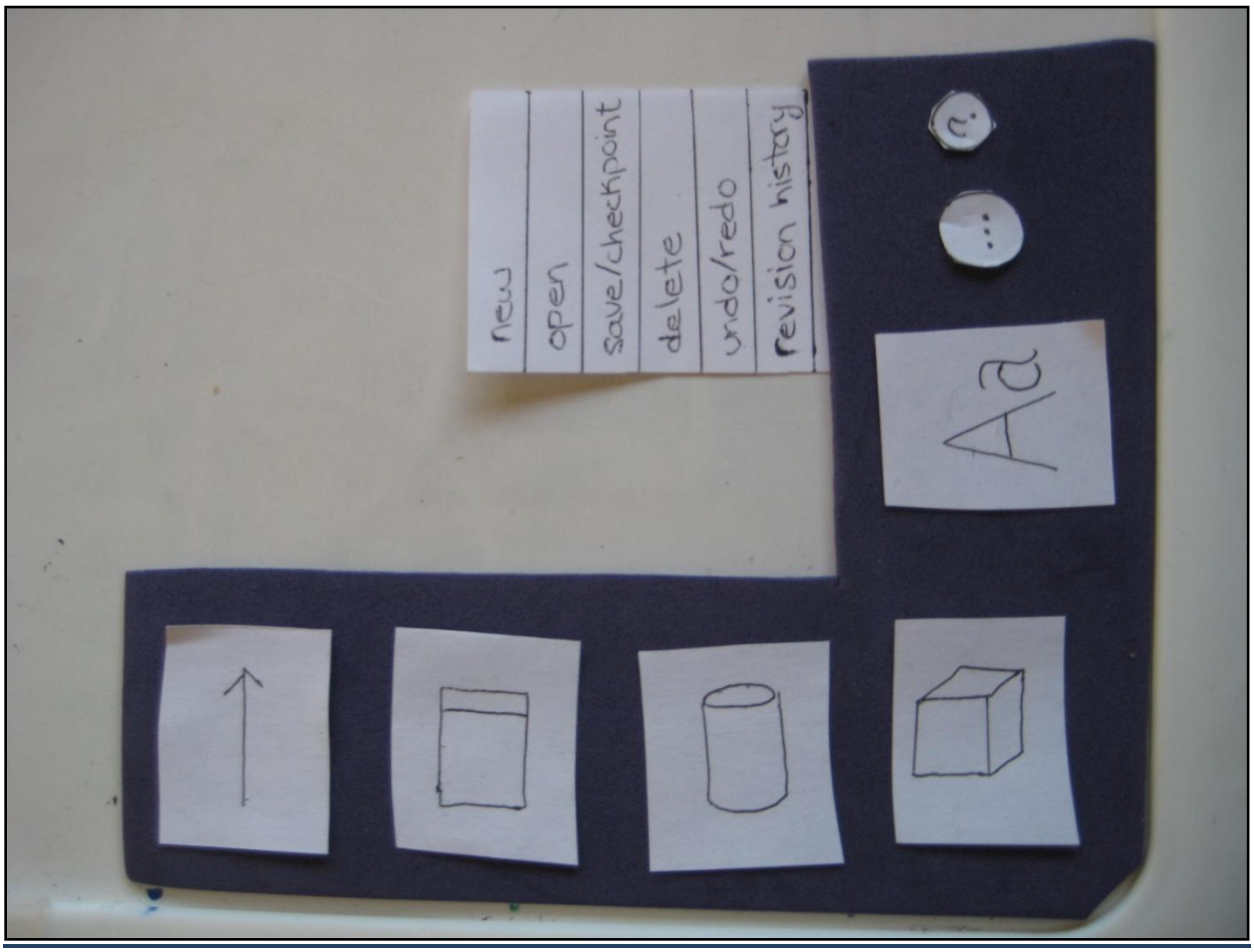

Figure A3.b: Close-up of menu panel. The menu is fixed to the corner of the screen. Icons are oriented so as to allow two collaborating users to both see one or more icons oriented in their direction, which could facilitate sharing of tasks.

[A3.c]

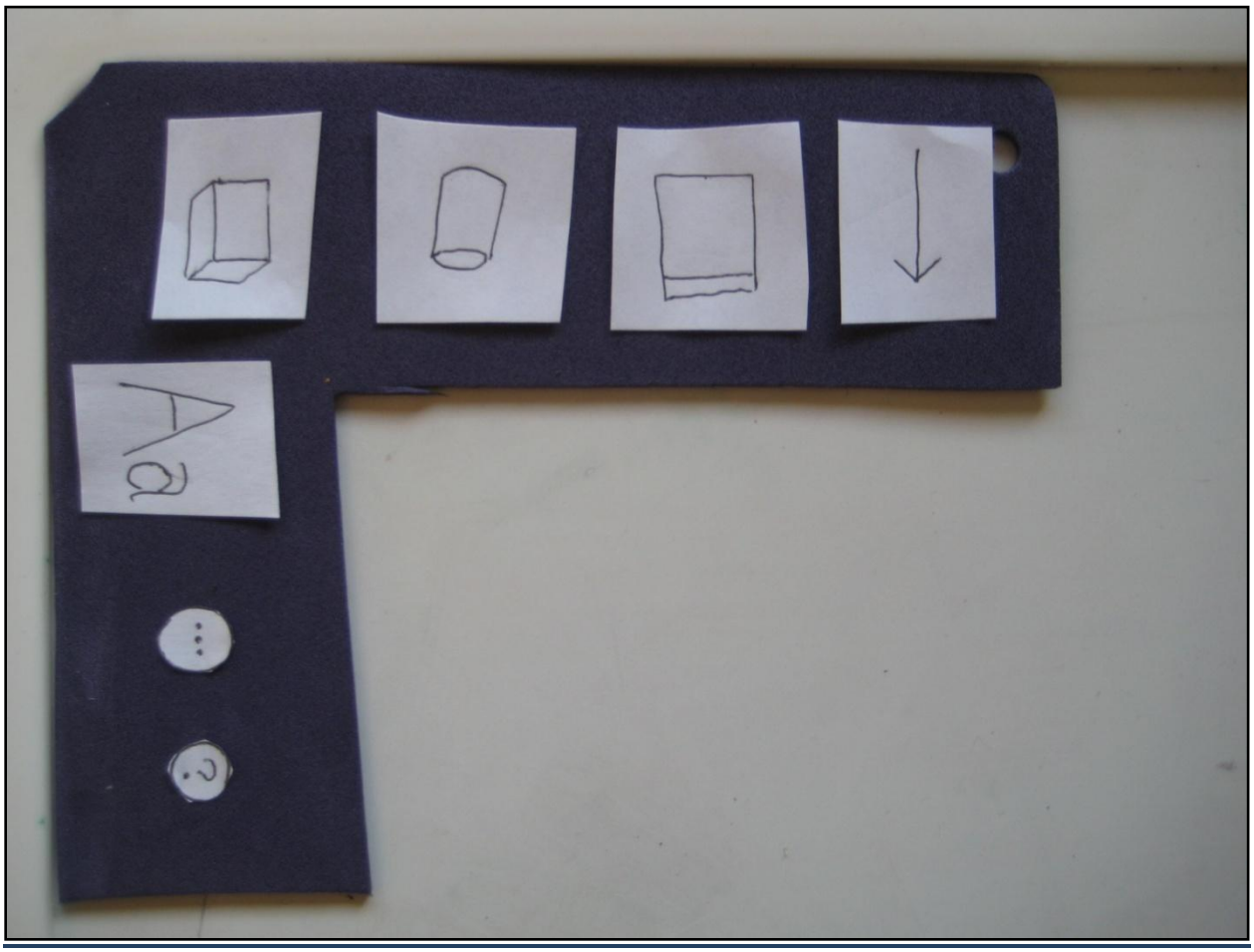

Figure A3.c: Close-up of top menu panel. The menu is fixed to the corner of the screen, in the opposite corner to the menu depicted in A3.b. Note that the icons are oriented towards users who would be sitting on the opposite side of the table.

[A3.d]

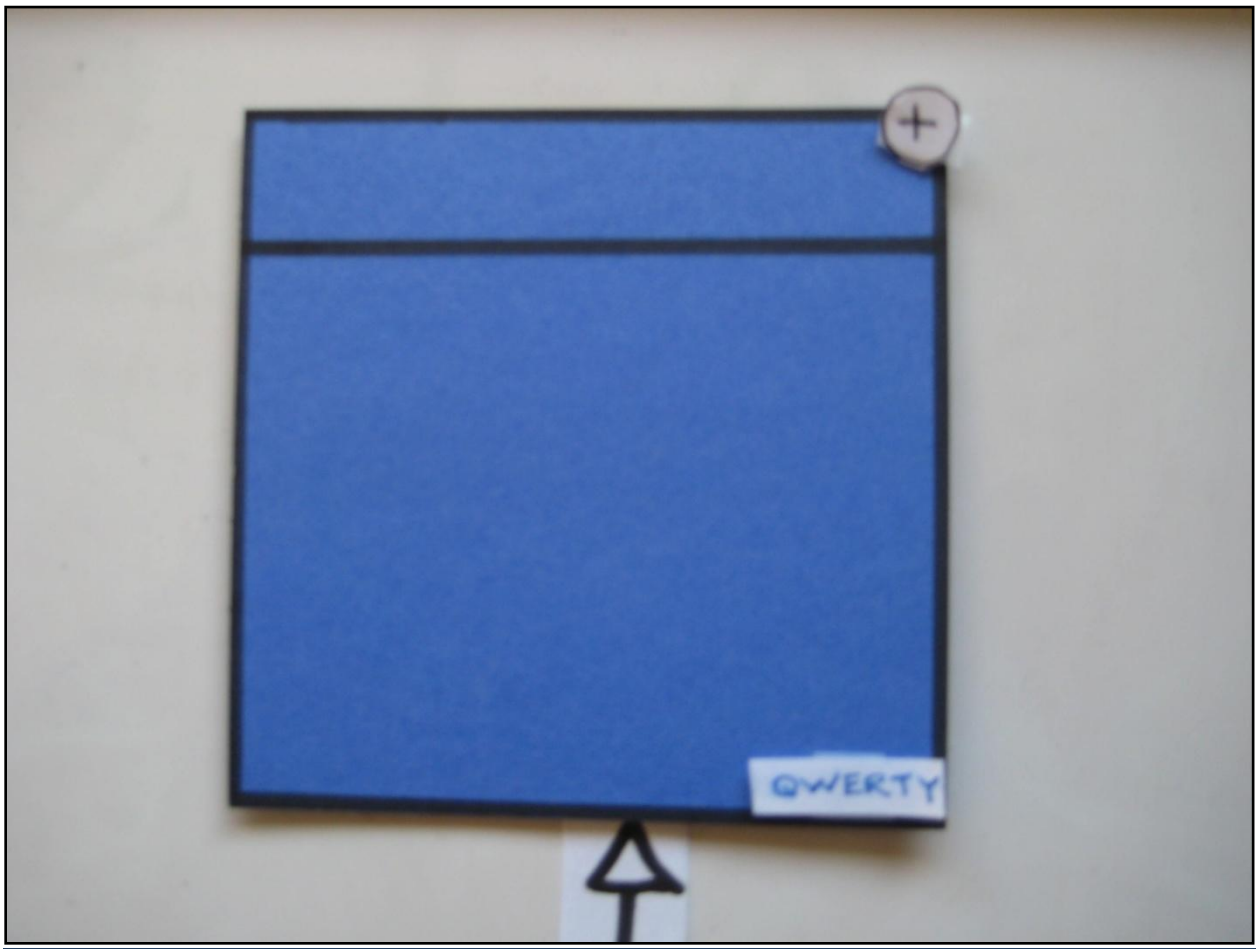

Figure A3.d: Close-up of a UML box. The (+) button in the top-right corner gives context-sensitive options for the object. The "QWERTY" button in the bottom-right corner brings up a keyboard input screen (we imagine something similar to the iPhone).

[B] Experimental Design

Goals of Experiment

Our goals for the experiment were to assess whether the system:

a) Allows users to collaborate effectively
b) Gives users a performance benefit over traditional tools
c) Provides a degree of user satisfaction at least as high as with traditional tools

Remarks

The key challenge in evaluating our system was to establish solid metrics for each of these evaluation goals. In particular, measuring collaboration proved to be difficult, given the somewhat complex and subjective nature of the concept. We opted to use a hybrid of subjective and objective methods to triangulate on, with certain methods and metrics drawn from the literature.

In order to establish that our design approach is viable, we hope to see an increase in user collaboration over conventional tools, while also showing that in the worst case, our system is not significantly worse with regards to functionality and enjoyment.

Experiment method

Participants

Our participants will include upper level university students. We are planning to recruit three teams of three users each for our experiment, for a **total of nine participants**. We will only recruit users who have completed our questionnaire from milestone II. The questionnaire will be used as a screening process to determine which users should be selected and also to determine how groups will be formed.

Conditions

We are comparing three different levels or conditions of the independent variable, namely interface type. The three conditions are **Whiteboard**, **Software**, and **Prototype**.

Tasks

The users are going to be asked to create a diagram to represent their design architecture for a given program. The task will require simply connection high-level architecture. The task will be designed to be too large for the participants to complete in the amount of time given, thus (hopefully) forcing participants to work collaboratively. There will be three tasks and each participant will complete one task per condition. See Appendix B1.a for the tasks given to participants.

Design

Our experimental design consists of **three trials on groups of three people** each. Specifically, for each diagramming method, we will be looking at **three metrics of evaluation** (collaboration, performance and satisfaction).

Collaboration will be measured using three well-established metrics: an objective measure of the time a group spends talking, coupled with a numeric count of the number of suggestions made[1], followed by self reporting by questionnaire and interview[2].

Performance will be measured by using a predetermined marking scheme that was designed for the problems we are assigning. The problems themselves are selected to be of equivalent difficulty by using old exam questions of the same worth.

Satisfaction is measured by self reporting[3] [2], using a questionnaire. The satisfaction ratings based on the questionnaire cannot be considered independent, and should be considered in aggregate and compared between groups during the analysis phase.

Our diagramming methods consist of three levels (white-board, computer, and our medium fidelity prototype); therefore, we will be assessing our three dependent variables using **three 1-way ANOVA tests**, **one for each dependent variable**. Our subjects will be randomly assigned different questions on each diagramming method, so as to minimize learning effects. A short practice question will be given before each round to minimize learning curve effects.

Procedure

1. Each group of users will be given an instruction sheet with a single UML diagramming problem on it (see *appendix B1.a* for instruction sheet; see *appendix B1.b* for UML problems)
2. Each group will be directed to one of the three stations – whiteboard, external software, prototype – and given a short warm-up practice question to be done at that station
3. Each group will be given one of the UML diagramming tasks, and told to use the tool(s) at their station to attempt the assigned task
4. Users will be given 10 minutes to complete each task
5. A stopwatch will be used discreetly at each station by the experimenters to record how much time users spend talking

---

[1] Morris, M.R. and Winograd, T. Quantifying Collaboration on Computationally-Enhanced Tables. *CSCW 2004 Workshop on Methodologies for Evaluating Collaboration Behaviour in Co-located Environments.*

[2] Carl Gutwin , Saul Greenberg, The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces, *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, p.98-103, June 04-16, 2000

[3] Kevin Baker , Saul Greenberg , Carl Gutwin, Empirical development of a heuristic evaluation methodology for shared workspace groupware, *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, November 16-20, 2002, New Orleans, Louisiana, USA

6. After the allotted time is up, we will take pictures of users' end products for later scoring, and erase whatever they have created
7. Groups will then be rotated onto the next station in round-robin fashion
8. After all three trials are completed, we will conduct a brief unstructured interview to determine the users' overall impression of each of the diagramming tools, followed by a questionnaire involving a self-assessment of their perceived level of collaboration and overall satisfaction.

Apparatus

The experiment will have three stations. Each station will have its own set of equipment along with the following common equipment:

- 3 sheets of paper, each one with 1 of the 3 tasks to be performed
- 1 sheet of paper with a practice task to be performed
- 1 sheet of paper with instructions for how to complete the tasks
- 1 sheet of paper with suggestions for how to use the medium
- Timing device
- Digital camera

The individual stations and their equipment are as follows:

Whiteboard Station

- Whiteboard
- 3 whiteboard markers, consisting of 3 different colors

Computer Station

- Computer with browser open and *Creately*[4] loaded (see appendix B2 for screenshots)
- *Creately* set up with only the tools needed to complete the task

Table Prototype Station

- *SMART Table*[5] with our prototype loaded (see part C for screenshots of our prototype)

Variables

Confounding Variables

- Facility in software architecture design: users may take more/less time to come up with design ideas
- Facility in diagram creation
- Computer ability
- Familiarity with target OS
- Disposition
    - if we are measuring enjoyment post-experiment, then the general "disposition" or "attitude" of our participants may be considered an independent variable

[4] A web-based diagramming and design application service operated by Cinergix, Pty Ltd. (http://www.creately.com/)

[5] A multi-touch, multi-user touchscreen developed specifically for primary education (http://www2.smarttech.com/st/en-US/Products/SMART+Table/

Independent Variables

- Diagramming Method
    - three levels: whiteboard, software, prototype

Dependent Variables

1. Collaboration level
    a. Measured by amount of time a group spends talking, coupled with a numeric count of the number of suggestions made, followed by self reporting via questionnaire and interview
2. Performance
    a. measured using a predetermined marking scheme taken from the exams in which the problem(s) appeared
3. Satisfaction
    a. measured using a 5-point Likert scale (see Appendix B1.b)

Hypotheses

1. **Null:** User collaboration is equal regardless of the diagramming tool used
   Alternative: Collaboration level is not equal among all three methods of diagramming
2. **Null:** User performance is equal regardless of diagramming tool used
   Alternative: User performance level is not equal among all three methods of diagramming
3. **Null:** User satisfaction level is equal among all three methods of diagramming
   Alternative: User satisfaction level is not equal among all three methods of diagramming

Planned Statistical Analyses

We will analyze each of the dependent variables in turn, using 1-way ANOVA on each of the three levels of the independent variable. For example, to test null hypothesis #1 (see above), we would conduct a 1-way ANOVA on the collaboration score from whiteboard vs. software vs. our prototype. If we obtain a critical F score, we will further investigate to determine which diagramming method yielded the highest collaboration level. This will be accomplished by doing a cross-combination of t-tests. This procedure will be repeated for each level of the independent variable. Summary reports on the analysis results will be compiled for each of these levels and presented in a subsequent report.

Expected Limitations

Due to the difficulty of measuring collaboration, we fully accept that our study is limited in its scope of assessment. Our metric of collaboration – measuring the amount of time spent talking coupled with a simple count of the amount of suggestions made – does not fully take into account parallelization and delegation of tasks, as well as other levels of collaboration, such as speech acts, gestures, and body language.

Our metric of performance could be skewed towards academics rather than people in industry as it's taken from university exams. Also, the ability of our participants to answer exam-style questions could

significantly affect the outcome of our performance measure, without giving useful information as to how our users would perform under normal conditions.

Measures of satisfaction using self reporting can be positively skewed due to prior acquaintance with our subjects. We hope that an aggregate comparison between subject groups can counteract this by showing a preference for one method over the other rather than being a pure measure of satisfaction.

Appendix B

[B1] Study Instruments

[B1.a] User Instruction Sheet

Your group will be performing a UML-like design task. You will have 5 minutes for each task, and you should focus on completing as much as possible, but you are not expected to finish.

Do not worry about all the specifics of UML diagramming, especially when it comes to differences between arrow, box and line types. Please do try and fill in class names, fields and methods as appropriate. We will be assessing how much you complete and the quality of your team's design.

Whiteboard
At this station you will be using a whiteboard to complete the design. Each of you will receive a different color pen and everyone should use their own pen throughout the process.

Creately
At this station you will be using a diagramming tool called *Creately*. You will only need to use the box tools and the connectors.

Prototype
At this station you will be using our prototype.

Please take a minute to familiarize yourself with the tools at your disposal. Inform the facilitator when you are ready to receive your tasks and begin the experiment.

[B1.b] User Tasks

The following are the tasks we will have users complete on the various interfaces (whiteboard, computer-based, prototype). They have been obtained from previous CPSC 310 exams in order to be as equivalent in difficulty as possible. The marking scheme following each task description is going to be used to gauge participants' completion rate, in an effort to obtain a quantitative reading of their performance with the various tools.

1. Draw a UML class diagram for the following software system for modeling a bank. Make sure to include multiplicities in your diagram.

Each of the bank's customers can access their account(s) through withdrawals, deposits, or balance inquiries at a bank machine. Each transaction (ie, withdrawal, deposit or balance inquiry) must store the date and time that the transaction occurred. Once a month, a statement that contains a list of all of the transactions that were completed over the last month is generated for each account and mailed to the customer. The bank must be able to produce a list

of all of its customers as well as a list of transactions that were completed by a particular bank machine.

Classes: Customer, Account, Transaction (10 pts each, total 30)
Links: (10 pts each, total 20)

- Customer ↔ Account(s)
- Account ↔ Transaction(s)

Fields: (2pts each, total 20)

- *Customer:* name, accountIds[]
- *Account:* uid, balance, transactionIds[]
- *Transaction:* tid, type, amount, date, bankMachineId

Methods: (2pts each, total 20)

- *Customer:* newAcct(uid), delAcct(uid), generateTxList(), getName(), setName()
- *Account:* getId(), newTx(type, amount)
- *Transaction:* getBMId(), getAmount(), getType()

2. Draw a UML class diagram for the following software system for modeling a restaurant. Make sure to include relationships, key methods, and multiplicities in your diagram.

At a restaurant, groups of customers are seated at tables and each table is served by one server. Each customer orders from a menu. The menu contains sub-menus as well as individual items. At the end of the meal, each table is given a bill, which lists all of the items that were ordered by the customers at that table, along with the prices of the items and the total amount owing by the table.

Classes: Table, Menu, Item (10 pts each, total 30)
Links: (10 pts each, total 30)

- Table ↔ Menu
- Menu ↔ Menu(s), Item(s)

Fields: (2 pts each, total 12)

- *Table:* tid, orderedItems[]
- *Menu:* submenus[], items[]
- *Item:* name, price

Methods: (2 pts each, total 18)

- *Table:* getBill(), setMenu(), getMenu()
- *Menu:* addItem(), removeItem(), addSubMenu(), removeSubMenu()

- *Item:* getName(), getPrice()

3. Draw a UML class diagram for the following system. Be sure to include the multiplicities, key methods, and attributes.

You are writing a retail store management system. The store sells clothing including jeans, tops, jackets, and shoes. The system needs to track the inventory and the sales. For each sale, the system must know the items that were sold, the method of payment and the cashier who completed the transaction. Each cashier has a name, contact information, and a unique id. The inventory includes all of the items that the store sells. Each item has a specific cost associated with it.

Classes: Cashier, Sale, Item, Inventory (10 pts each, total 40)
Links: (10 pts each, total 30)

- Sale ↔ Cashier, Item(s)
- Inventory ↔ Item(s)

Fields: (1 pt each, total 13)

- *Cashier:* uid, name, contact
- *Sale:* saleId, paymentMethod, cashier, items[], totalCost
- *Item:* name, type, cost, countAvailable
- *Inventory:* items[]

Methods: (1 pt each, total 7)

- *Cashier:* getUid(), getName(), makeSale()
- *Sale:* getTotalCost(), getCashier()
- *Item:* getPrice()
- *Inventory:* getItems()

[B1.c] 5-point Likert scale for measuring user satisfaction

Overall, how well did tool X perform, in terms of letting you do what you needed to accomplish the assigned task(s)?

1. Very well
2. Well
3. Neutral
4. Poor
5. Very poor

Overall, how enjoyable did you find using tool X to be?

6. Very enjoyable
7. Somewhat enjoyable
8. Neither enjoyable nor unpleasant
9. Somewhat not enjoyable
10. Not enjoyable at all

Overall, how satisfied were you with tool X?

11. Very satisfied
12. Satisfied
13. Neither satisfied nor dissatisfied
14. Dissatisfied
15. Very dissatisfied

Would you use tool X again?

16. Definitely
17. Possibly
18. Not sure
19. Likely not
20. Definitely not

[B2] Screenshots of *Creately*

[B2.a] Selecting/focusing on an object

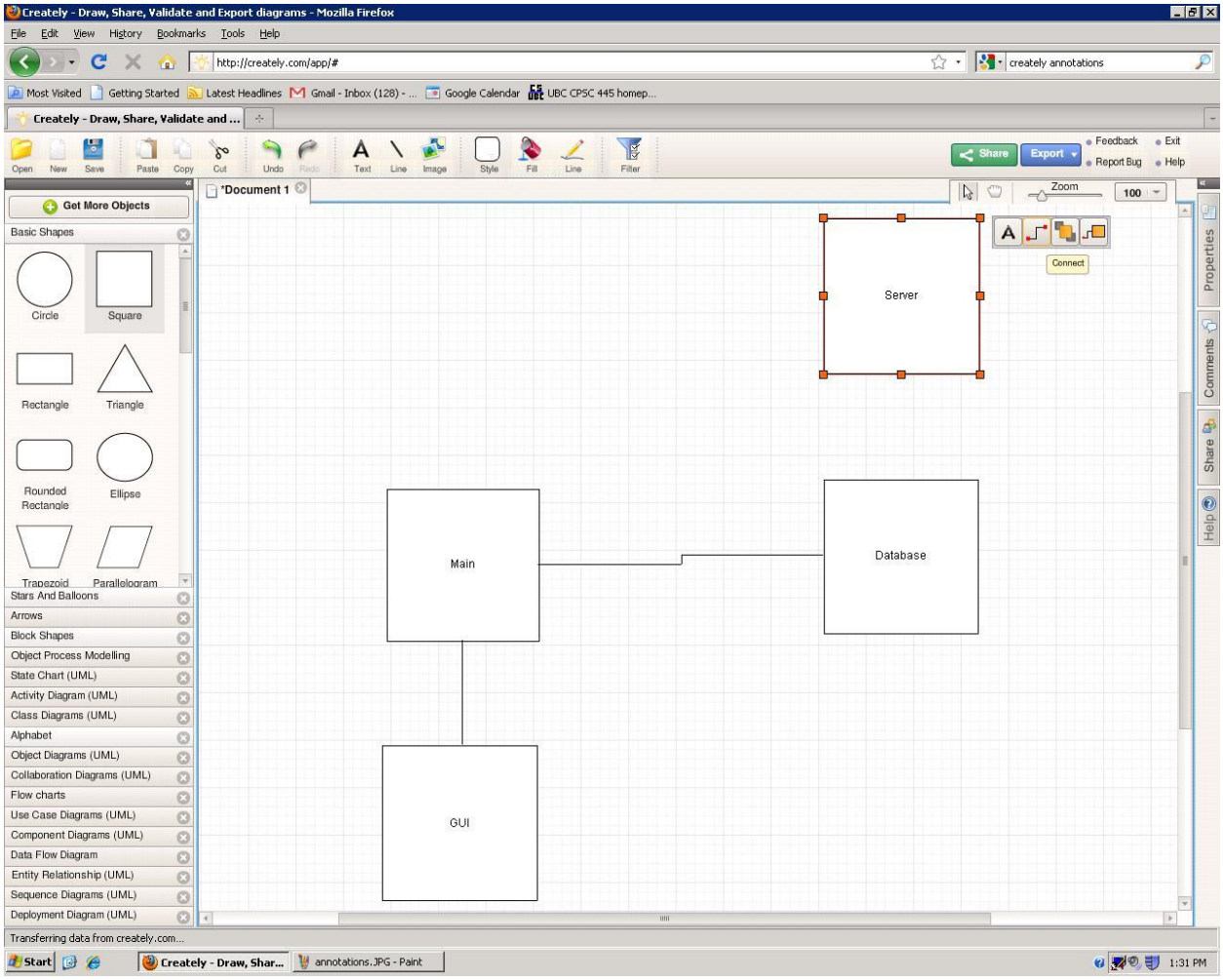

Figure B2.a: Clicking on an object both selects and focuses on the object, producing a context-sensitive menu

[B2.b] Multi-attribute boxes with interconnections

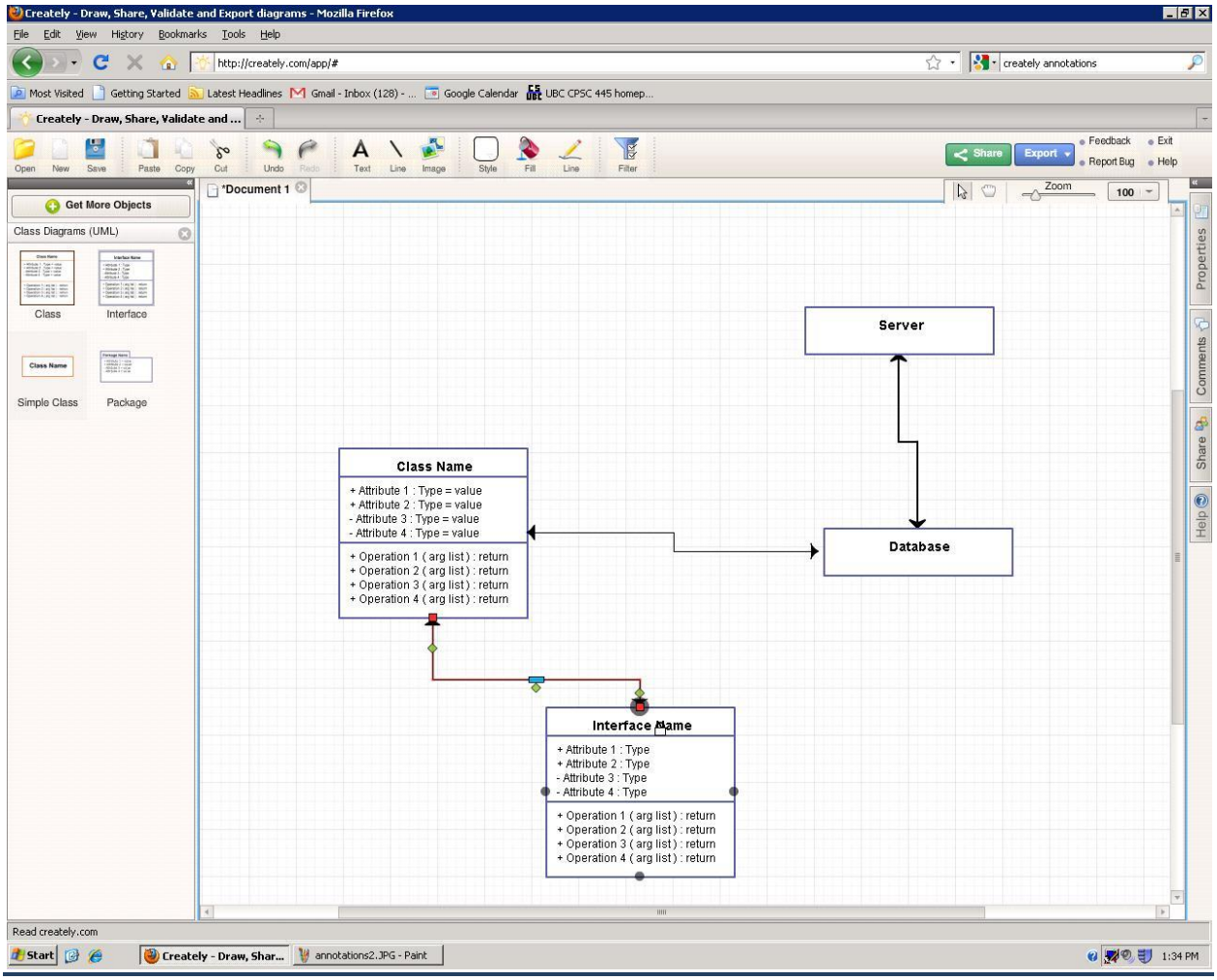

Figure B2.b: Creately allows for highly detailed objects to be created with flexible interconnections. This figure also shows the ability to select and modify a line object. Line endpoints can be dragged to new positions, or onto different objects.

[C] Medium-Fidelity Prototyping

Rationale of Prototyping Approach

We have chosen to adopt a hybrid horizontal/vertical approach to the medium-fidelity prototype, with the full horizontal functionality visible, but only select features having vertical depth.

We wanted to implement a substantial portion of the functionality required for our task examples (see *appendix A1*), while also producing a prototype that could be used to support subsequent user evaluations on tasks similar to these examples. At the same time, we didn't want to spend too much time worrying about features that would be peripheral to the user study's requirements. Therefore, we opted to avoid vertical implementation of several components, including the revision history, file navigator, menu interactions, and dialog box interactions. We focused instead on implementing all the features required for the diagram creation portion of our design, including: adding, editing, deleting, selecting, moving, and linking objects (UML boxes *and* lines). The goal was to get the most useful results from the user study, with the least amount of implementation effort.

We wanted the medium-fidelity prototype to stand on its own – to be functional without any outside help. This is because in order to conduct the user study most effectively, we decided that Wizard-of-Oz techniques should not be used. We felt that this type of intervention would interfere with group collaboration, and skew our measurements of it.

We did not choose a traditional tool with which to implement our prototype. Instead, we used the SMART Table SDK, so that our medium-fidelity prototype could be evaluated in its natural setting: a multi-touch surface. This ensured that all the gestural actions involving multiple touch-points, which constitute the main functionality of our system, were available to the users. It also meant that our prototype could be used simultaneously by multiple users seated around a table, which we believe is one of the key factors affecting user collaboration levels. If we attempted to conduct a user study on our prototype in any other environment, we feel that our results would not reflect our design concept.

Prototype Illustrations

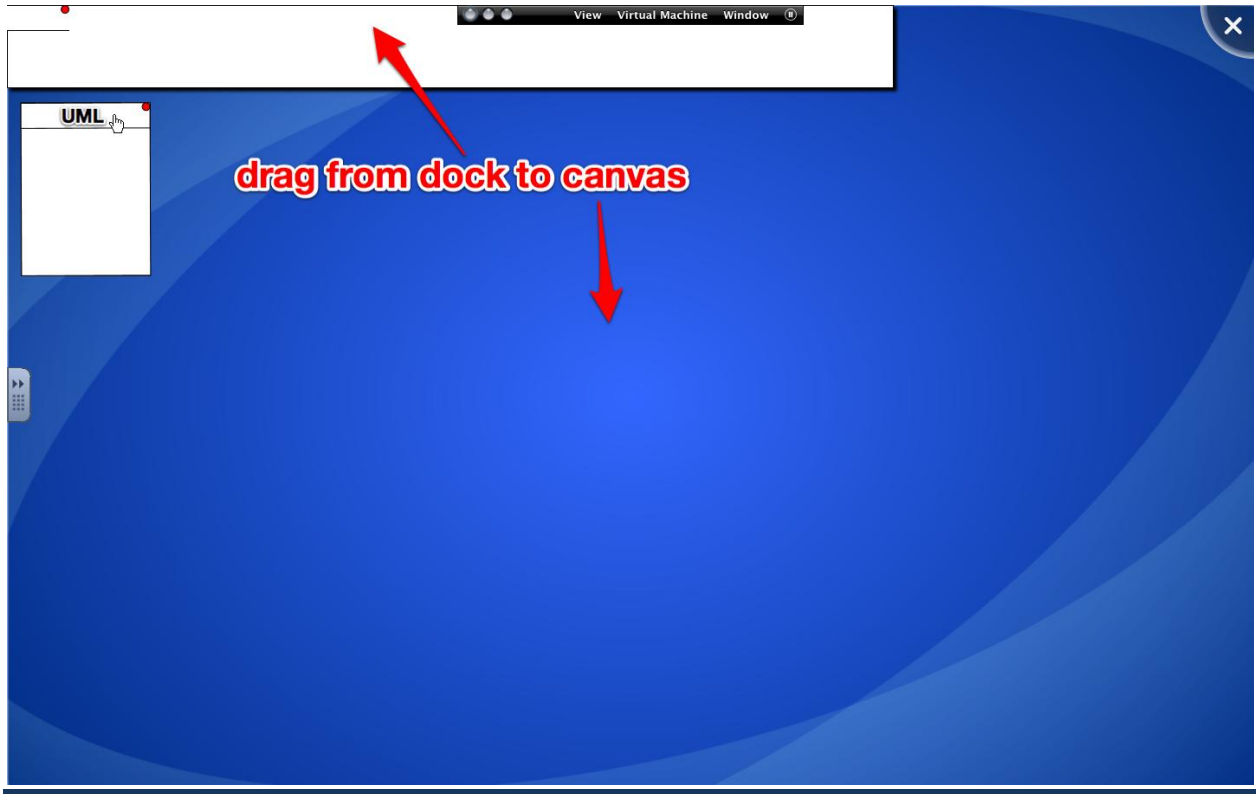

Figure C1: Objects can be dragged from the taskbar onto the screen area. The UML box on the top left can be dragged to new positions and connected to other objects with interconnecting lines.

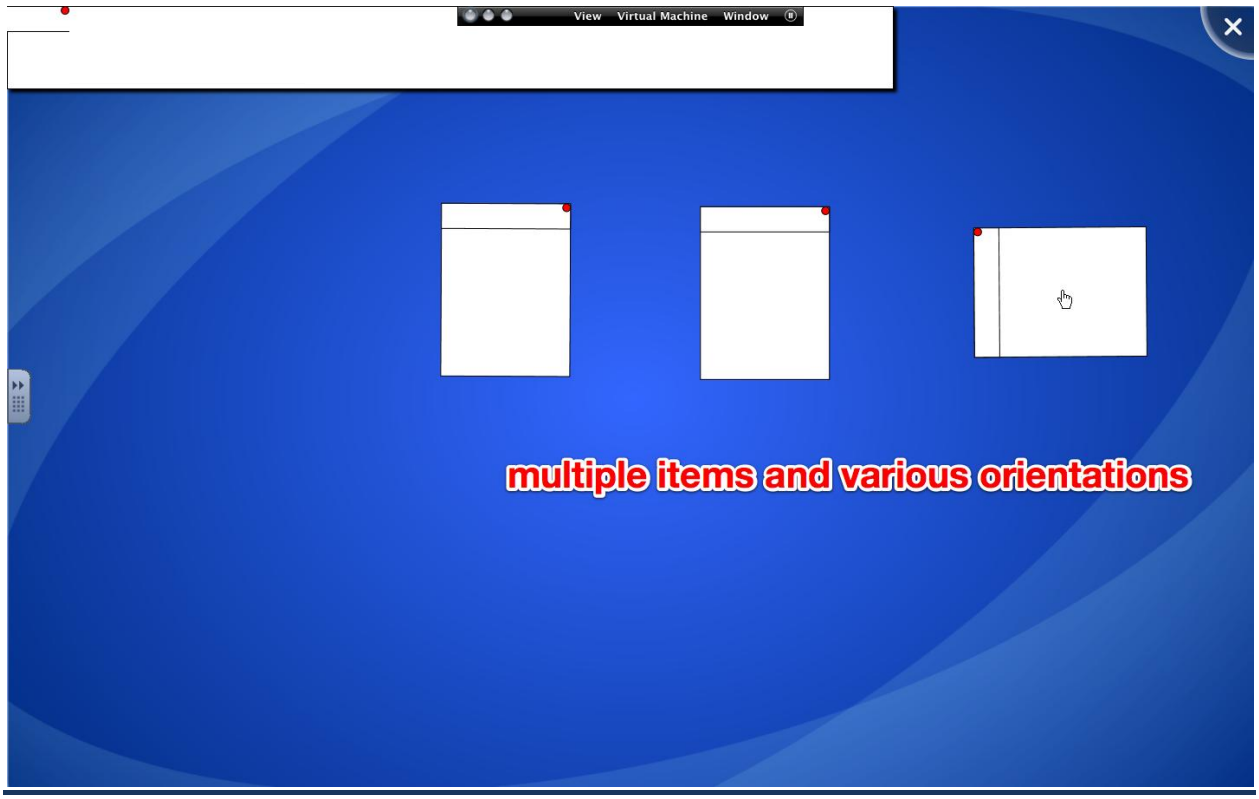

Figure C2: Items can be dragged, rotated, and repositioned