An Empirical Evaluation of Undo Mechanisms

Aaron G. Cass
Union College
Schenectady, NY 12308
USA
cassa@union.edu

Chris S. T. Fernandes
Union College
Schenectady, NY 12308
USA
fernandc@union.edu

Andrew Polidore
Union College
Schenectady, NY 12308
USA
polidora@union.edu

ABSTRACT
While various models of undo have been proposed over the years, no empirical study has yet been done to discover which model of undo most closely aligns with what users *expect* an undo command should do. In this paper, we discuss the results of such a study that compares the ubiquitous linear undo model with two variations of selective undo: script selective and cascading selective. Unlike the script model, cascading selective undo takes into account dependencies between user actions. Our study shows that, for the application studied, when a user is asked to perform undo in the absence of any guidance, the user will tend to gravitate toward an undo mechanism that uses existing dependencies between user actions. Specifically, we show that subjects prefer the dependency-aware aspects of cascading undo over either linear or script selective undo.

Author Keywords
keywords: undo, selective undo, linear undo, empirical evaluation.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces – Evaluation/methodology, Theory and methods; H.1.2 [Models And Principles]: User/Machine Systems – human factors, human information Processing; General Terms: Design, Experimentation, Human factors

INTRODUCTION AND STATEMENT OF THE PROBLEM

In most commercial applications which support an undo feature, the implementation follows a linear model. In this model, the most recently performed action is the one that is undone. If a history list is supported, when a user requests some action A_i be undone, all actions recorded in the history list after A_i will be automatically undone by the system. This is clearly an improvement over requiring the user to repeatedly undo actions starting with the most recent and proceeding backwards through the history list until the desired action is reached. The history list provides the user with a more flexible way of interacting with the system. Is this enough flexibility? Is this the natural understanding of undo that users come to our applications with? If it is not natural, what mechanism for undo should replace it, in what contexts, for which users?

As part of a larger research program, we are attempting to address these questions. In this paper, we present our recently completely empirical study that was designed to help us answer whether linear undo is natural to users. In the study, we explicitly compare two alternative approaches to linear undo that were designed to address perceived shortcomings of the linear model.

Despite the predominance of the linear model in current software, there is little empirical evidence that this model of undo is the most helpful to users or even if this model is consistent with what users have in mind when they think about the meaning of undo in a specified application. In fact, the linear model seems to us to be far too restrictive to be useful to users. It seems natural that a primary benefit of an undo mechanism is that it can encourage a user to explore – if the user can readily return to previous states, the user can execute commands without fear of losing valuable work. And yet, linear undo does not allow freedom of exploration. Consider a word processor user who wants to change a document from a current state $\mathcal{A}$ to another state $\mathcal{B}$, but is unsure of the exact steps she should take to accomplish this. It would be preferable for the user to be able to perform a series of tentative steps from $\mathcal{A}$ towards $\mathcal{B}$ all the while knowing that she could return to state $\mathcal{A}$ at any time via undo – even if other unrelated tasks, such as the changing of a word at the behest of the spell checker, were performed during the exploration process. Selectively undoing the tentative steps without changing the spell check results allows for greater flexibility, and therefore promises to encourage more exploration than the linear model allows.

So, while linear undo has become the *de facto* standard, others have introduced *selective undo* models that have advantages over the linear model in certain contexts [2, 4]. In a selective undo model, the user can undo an arbitrary action from the history list without necessitating retraction of all subsequent actions. It is believed that this will be easier for users to use as compared to the linear model. Is that true? Is a selective undo model preferred by users over the linear model? Is selective undo what users have in mind when they

think of undo? In this work, we undertake a study to find out.

Note that there is no agreed-upon semantics for selective undo. In fact, there are competing models in the literature, each of which purports to be a natural model that users will find easy to understand and work with. These competing models differ in many ways, but one primary way is in how they account for dependencies between actions in the history list. Some selective undo mechanisms ignore dependencies between actions, essentially assuming that user actions are independent of each other. While it is clear that user actions are not completely independent of each other, it is still possible that users don't think of dependencies, nor want to, when considering undo. However, others have argued that dependencies are important and must be accounted for. Which of these approaches best fits users' mental models? Which selective undo mechanism is preferred by real users? Our study attempts to address these questions by comparing two selective undo mechanisms with the prevalent linear model.

In the following section, we describe the comparison approaches and other relevant related work. We then describe the current study, including instrument development, sampling techniques, threats to validity, and hypotheses. We then present and discuss results before concluding and discussing future work.

COMPARISON APPROACHES AND RELATED WORK

Selective undo, while not widely implemented, is not a new concept. It was first proposed by Berlage [4] over ten years ago. Myers and Kosbie [7] assumed Berlage's semantics and studied the model more formally. By implementing *command objects* in their Amulet user interface, Myers and Kosbie organized user actions into a hierarchy which allowed higher-level commands to be invoked by lower-level commands. Their system not only supported selective undo, but selective reusability of arbitrary commands on new objects.

This type of versatility is not available in the linear model and provides a means for undoing actions in collaborative environments. Note that in a collaborative environment, some sort of selective undo is required, because a linear undo would be confusing to users. Consider the case where a user has just completed an action that they decide to undo. If another user performs an action in the mean time, undoing the most recent action that is seen by the system will cause the wrong action to be undone. Therefore, at least from the point of view of a global history list, the system must undo a task that was not the most recently executed. In other words, a selective undo is required. Several selective mechanisms for collaborative systems have been suggested [6, 8–10]. These are, of course, more difficult to implement. As a result, in these systems any semantic constraints on how one action's retraction may affect other actions are embedded into the algorithms internally.

It is these dependencies between user actions that we will focus on here. Some proposed selective undo models (implicitly) assume that there are no dependencies or that they will be handled separate from the undo mechanism itself. Consider, for example, the *script* model, discussed by Archer et al [2], which works as follows. Given n user actions, $A_1, ..., A_n$, the state of the document after the selective undoing of step A_i is equivalent to the state of the document had it undergone the steps $A_1, ..., A_{i-1}, A_{i+1}, ..., A_n$ in that order. In other words, the end result is as if the n user actions of the "script" had been executed with the ith step removed. This does not take into account possible dependencies between actions in $A_{i+1}, ..., A_n$ and the action A_i to be undone.

Cass and Fernandes [5], on the other hand, describe what they call *cascading selective undo*. Unlike the script model, cascading selective undo uses semantics about dependencies to determine if other actions besides the one chosen by the user should also be necessarily undone to return the document to a viable state. In general, if action A_i is chosen to be undone under this model, and subsequent action A_j is dependent on A_i having been successfully executed, then A_j is also undone. A_j depends on A_i if A_j requires an output produced by A_i or if the application developer defined an *a priori* dependency between the two actions. They refer to the first case as a data dependency, while the second case is called a control dependency. As an example of a control dependency, consider two actions that a user performs on an Automated Teller Machine (ATM): entering the Personal Identification Number (PIN) and entering the amount to withdraw. The PIN will only be validated immediately before the committing of the transaction. However, for security and usability reasons, the application designer has decided to request the PIN first. That is, an ordering has been externally imposed on the two subtasks, even though there is no semantic relationship between them. Thus, there is a control dependency between entering the PIN and entering the amount to withdraw even though there is no data dependency. It is these control and data dependencies that cause cascading in the undo algorithm presented by Cass and Fernandes.

Dependencies are recursively computed, so that if A_k depends on A_j, and A_j depends on A_i, then A_k depends on A_i. A_k is said to be in the cascade of actions to be undone when A_i is undone. This approach seems promising to us because it provides the flexibility of a selective undo mechanism while at the same time recognizing that user actions are dependent on one another.

THE STUDY

The goal of this study is to determine which undo mechanism is a more natural choice for users. That is, we wish to know what model users choose when not restricted by the application they are trying to use. While the cascading model seems to have advantages, and we think it is worth attempting to implement it for some application domains, we think this is only warranted once we know something about how natural it will be for users to understand and use it. Because, to the authors' knowledge, no empirical work suggests that users find any particular undo model most natural, we choose to undertake a descriptive study as a first step.

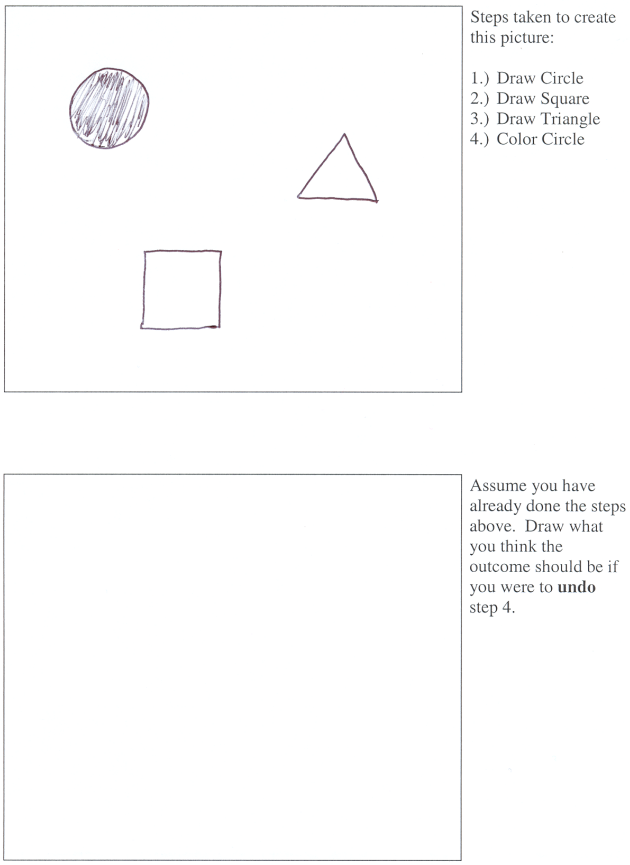

Figure 1. The instrument given to subjects for the undo task.

In the study, subjects are asked to perform a task in which they must perform undo. By observing how they choose to undo, we will learn what forms of undo are most natural. Our goal is to define a flexible task in which there is more than one possible undo mechanism that subjects *can* choose, and then determine what subjects *do* choose. We will then learn which of the possible mechanisms are most coherent with users' mental models. The task will be designed to compare linear, script, and cascading undo models.

We hypothesize generally that one of the undo mechanisms will be preferred over the others. Specifically, we predict that linear undo, because of the restrictions that it places on the user, will not be the preferred choice of many subjects. We do not know which of cascading and script will be most preferred, but we predict that one of them will be preferred.

In the rest of this section, we describe the instruments we have designed for this task and give details of how we carried out the study before we restate the hypotheses in terms of specific measures from the study and then discuss threats to validity.

Instruments

The study consisted of two instruments, both paper-based, involving the drawing of shapes in a prespecified area. Figure 1 shows one of the instruments in its initial state. Each instrument was roughly divided into an upper half and a lower half. In the upper half, three shapes were predrawn for the subject before the subject was allowed to see it: a square, a triangle, and a filled circle. A researcher hand-drew these shapes for a day's worth of subjects at a time. The figures were always drawn in the same positions. To the right of the frame in which these shapes where drawn, the following explanation appears:

> Steps taken to create this picture:
>
> 1. Draw Circle
> 2. Draw Square
> 3. Draw Triangle
> 4. Color Circle

In the lower half is a blank frame identical in size to the frame in the upper half. To the right of this frame are the following instructions to the subject:

> Assume you have already done the steps above. Draw what you think the outcome should be if you were to **undo** step 4.

The only difference between this and the second instrument is that the other task asks the user to undo step 1 instead of step 4. In each case, subjects were asked to draw what they believed would be the results of the undo.

We felt that a paper-based set of instruments would be preferable to a computer-based set of instruments for several reasons. First, we wished to de-emphasize the idea that the undo concept was tied to a specific implemented application. For those familiar with undo in particular applications, especially painting programs, such an instrument may bias subjects into thinking that we are searching for their knowledge about how undo *actually* works in such an application instead of thinking about how they would *like* it to work. Second, an instrument where the subject was required to draw the results allowed each subject more freedom to implement the undoing in the way she wished. We would then have a record not only of what was drawn, but where and how it was drawn as well. As will be discussed in the next subsection, the recording of *how* the shapes were drawn became an issue in the pilot study. This facet would have been difficult to record in a computer-based instrument. Third, since the two forms of selective undo mentioned are not widely implemented, a custom implementation of a drawing program or else a Wizard of Oz experiment would have had to be used. The paper-based instruments allowed us to set up the study quickly while providing the other benefits already mentioned.

Note that, with this instrument, we do not directly compare cascading selective undo with the script model to the fullest extent because the instrument does not give subjects any information about *a priori* dependencies between user actions. There is clearly a dependency between step 1 and step 4 in the task, but it is a data dependency – the circle drawn by

step 1 is used as input to the coloring operation in step 4. However, the current study does address an important question – namely, do users want an undo mechanism that makes use of dependencies between user actions? We felt that addressing this question was a clear first step toward evaluation of cascading and script models for undo. Furthermore, in order to test subjects' use of *a priori* dependencies, we would have had to give them training on those dependencies. We feel this would risk biasing subjects in favor of a cascading model.

Pilot Study

Before arriving at the final instruments described above, we ran a pilot study to determine initial flaws in our design. Specifically, we wished to determine the wording to use in the directions. We wanted to avoid, as much as possible, biasing the subjects in favor of one particular mechanism for undo. We worried that if we asked subjects to "undo" a step, they might be biased from previous experience with implemented applications into thinking that their task was to "undo" as they had come to expect it, not as it *should* be.

We therefore created two versions of our pilot instrument, one that asked subjects to "undo" steps and another that asked subjects to "reverse the effects of" those same steps. We created four drawing instruments identical to the ones described, except for the phrasing of the instructions. The first instrument reads:

> Assume that you were given the list of steps above. Draw what you think the outcome should be if you were to **reverse the effects of** step 4.

A second instrument had identical instructions, but step 1 was to be reversed instead of step 4. The third and fourth tasks were identical to the first two except "**undo**" replaced "**reverse the effects of**". The alternative wordings were boldface. We gave these four sheets in random order to four test subjects. Unlike the larger study which followed, we had subjects think aloud as they answered the questions.

We made two decisions due to the results of the pilot study. First, it became apparent early on that the test subjects were not confused by "undo", but were very confused by "reverse the effects of". To reverse the effects of step 1, one pilot subject colored in the background of the frame, leaving the shapes unfilled, thus producing a negative image. Another altered the positions of the shapes by reflecting about an arbitrary axis. Still another reversed the way the shapes were drawn (drawing the circle clockwise instead of counterclockwise, for example.) To eliminate confusion, we therefore chose to use solely "undo" in the directions when the tasks were given to the larger test pool.

The second result was to change the instructions to more clearly indicate that the upper half of the instrument (the original user actions) had already been executed. We wished to make it clear that only completed user actions are chosen to be undone.

first-year	4
sophomore (second-year)	10
junior (third-year)	8
senior (fourth-year)	6
not answered	1
total	29

Table 1. Frequencies for college year, based on expected graduation date, for subjects in the study

Setup of the Study

With the instruments designed, we proceeded to carry out the study. The empirical study was run on subjects who were kept anonymous in the analysis of the results. This was done by assigning each subject a number and then keeping a private list of the subjects' names linked with their corresponding number. One of the authors observed all of the sessions with subjects and was the only person to know the identities of the subjects. This observer was not involved in the actual analysis of the data and at no time gave the list of subject identities to the other authors.

The observer met each subject individually in a quiet study lounge. Subjects were not given any training or introduction to the instruments beforehand, though each was given a short written paragraph explaining the expected duration of the session (30 minutes), the ability of the observer to answer questions in such a way that only clarifies the instructions, and an assertion that the subject was not being tested in any way. With this sheet, they were also given a standard consent form which each subject read, signed, and dated. The observer then gave the subject the first of two task sheets, sat off to the side so as not to be distracting, and took notes on how the task was performed. When the subject was done with the first task, it was collected, the second was given, and the observer recorded the time taken for the first using a watch. The same was done with the second task. The observer administered the two tasks in random order.

Following completion of the tasks, the subjects were asked to fill out a questionnaire. The questionnaire asked the subjects to provide information regarding their general computer use, their experience with undo, and their reactions to the specific questions in the study[1].

Sampling

Subjects were encouraged to participate by paying them $20. We recruited subjects via flyer, email, and announcements in various classes. Specifically, announcements were made in an introductory course for computer science majors, a computer literacy course for non-majors, and a critical thinking and writing seminar taken only by first-year students. Despite the fact that we solicited subjects in courses where computer use is emphasized, a broad range of subjects actually participated. In total, twenty-nine subjects participated:

[1] In general, subject reactions were not noteworthy. The most common reaction indicated that many subjects had to pause to contemplate the meaning of undo, which is what we hoped they would do.

courses for computer science majors	3
non-traditional introduction to programming	3
general education computing courses	4
no computing courses	18
not answered	1
total	29

Table 2. Frequencies of subject responses to a question about previous computer-related coursework

four freshmen, ten sophomores, eight juniors, and six seniors (see Table 1). One student did not give a class level.

Table 2 summarizes subject responses when asked what computer coursework they had. Notice that the vast majority did not list any computer-related coursework at all, and few of those that had coursework reported having taken courses designed for computer science majors. Note that Union College offers general education courses such as the computer literacy course previously described and a course on history of computing. Union also offers non-traditional introductions to the computing field such as a course in MATLAB for engineers and a course dealing with introductory programming with multimedia files.

When asked about previous background with computer applications, the vast majority, twenty-six, said they worked with Microsoft Office products, suggesting that very few subjects lack experience with applications that implement some form of undo. In fact, six subjects specifically listed experience with the drawing tool Photoshop, which allows users to interact more directly with the command history than other applications. Note that all of these applications implement linear undo, so if the sample is biased, it is biased in favor of linear undo.

We also recorded the subjects' previous experience with undo, specifically asking in what ways they executed undo in computer applications. Sixteen subjects said that they picked undo from a menu. Three said they used a toolbar button. Thirteen used a keyboard shortcut (typically, Control-Z). Note that with a keyboard shortcut, only linear undo is possible. For this question, several subjects listed multiple undo methods.

Variables and Measures

Our study had one dependent variable: the type of undo the participants performed. The same tasks were given to all participants, and we coded their choice of undo as one of four values–linear, script selective, cascading selective, or other.

The instrument where Step 1 (Draw Circle) is to be undone produces different results depending on the undo model used. A linear undo should result in an empty drawing, since step 1 and the remaining 3 should all be undone. Under the script model, the final three steps should still be executed. The step of coloring the circle without first having drawn it is manifested as a filled-in region in the shape of a circle, but lacking the outline of the circle. The cascading model causes the circle to disappear completely, leaving just a square and a triangle, because the coloring of a circle depends on there being a circle to color. Two subjects produced responses that deviated from the above scenarios. They were coded as "other".

Hypotheses

The overall design of the study, therefore, was to use the above-mentioned instruments as instructions for tasks that subjects performed. The subjects drew what the drawing should look like after the undo operation is completed – in essence the subjects simulated execution of their own model for undo. We are only interested in the task in which subjects were asked to undo step 1, because for that task, there were several possible undo operations:

Linear Undoing step x causes all steps after step x to be undone.

Script Undoing step x causes the drawing to be in a state consistent with the state reached by executing all steps other than x.

Cascade Undoing step x causes the undoing of all steps that depend on step x having already completed.

We expected that one of these methods would be preferred over the others. In other words, the frequencies of the different methods were expected not to be equal. We therefore expected to be able to reject the null hypothesis $H_0(1)$ with alpha level at 0.05:

> $H_0(1)$ **(frequencies are equal):** Subjects will choose linear, script, and cascading selective undo with equal frequency.

We planned that if we found that this hypothesis could be rejected, we would then proceed to test further hypotheses. The next two concern linear undo. Researchers that have developed selective undo mechanisms (e.g. Sun [10], Prakash and Knister [8], and Berlage [4]) clearly do not think that linear undo is a natural concept. It restricts the user unnecessarily. We agree that linear undo is only easy for the application developer – the user probably does not think of their actions in the chronologically restricted way that linear undo enforces. Therefore, we expected that linear undo would be preferred less than the other two. We therefore expected to be able to reject the following null hypotheses with one-sided tests:

> $H_0(2)$ **(linear vs. script):** Subjects will choose linear undo and script selective undo with equal frequency.
>
> $H_0(3)$ **(linear vs. cascade):** Subjects will choose linear undo and cascading selective undo with equal frequency.

However, a more interesting hypothesis regards the naturalness of different kinds of selective undo. We did not know which selective model would be most natural to users, but we suspected that subjects would generally have a preference either for using dependencies or not using them when

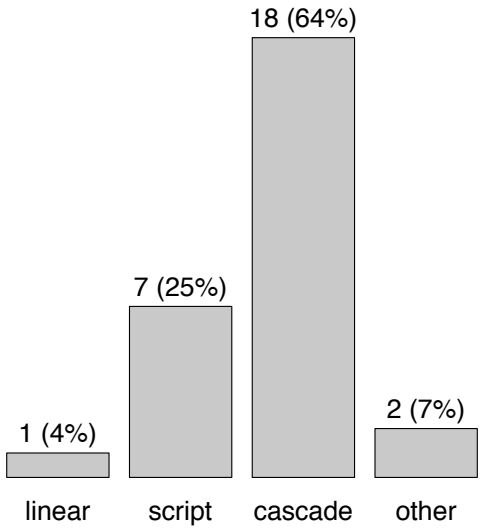

Figure 2. Frequencies (and percentages) with which each of the undo models was chosen by the subjects in our sample (N = 28)

simulating undo. Therefore, we expected that they would find either the script model, as defined by Archer et al [2], or the cascading model, as defined by Cass and Fernandes [5], most natural. We therefore expected to be able to reject the null hypothesis $H_0(4)$:

> $H_0(4)$ **(script vs. cascade):** Subjects will choose script and cascading selective undo with equal frequency.

Threats to Validity

Threats to Internal Validity

This study is designed to minimize threats to internal validity. By keeping the tasks the same for the different subjects, and randomizing the order that the tasks are done, we minimize the chance that the results we find are due to an artifact of the study design. Therefore, any differences we find are very likely to be real differences between the subjects on the variables we measure. Note also that we did not assign the subjects to groups, so there are no internal validity threats related to group selection.

Threats to External Validity

To ensure high likelihood that the results from our sample hold for the general population of computer users, we have attempted to get a representative sample of students across our campus. As discussed above, the subjects in our sample have similar experience with undo, and relatively few of the subjects have taken any computer science classes. In the end, the sample is perhaps not as representative as we would like, as it has a higher than proportional number of computer science majors. However, given that no subject is likely to have any experience with a non-linear undo mechanism, this does not seem such a large threat.

There are also risks that the results do not generalize to real-life situations. In order to focus the responses of subjects, we have contrived tasks for them to perform. The tasks themselves might not be representative of real tasks, for several reasons. First, the tasks are designed to be relatively simple so that they can be completed within the laboratory study, and they might therefore be simpler than tasks real users perform – and the increased complexity of the real tasks might encourage different mental models or different user behavior. In addition, real users are not likely to perform simple tasks *in isolation* – they perform tasks as part of a larger project. Our empirical study does not simulate this. Also, like most laboratory studies, we do not know to what extent the subjects' behaviors are affected by being measured (the Hawthorne effect). With the present study, we choose a laboratory setting because of the control it gives us. Future work will take a more ecological approach to ameliorate these risks.

In the Instruments subsection above, we presented our reasons for using paper-based prototypes in this study. We should point out here that this introduces a risk that the results will not generalize to computer-based applications. It is possible that users behave differently or adopt different mental models when using a computer application. We make these decisions knowing that there is this risk, which will be addressed in future work.

Threats to Construct Validity

In this study, we are attempting to discern the naturalness of linear and selective undo mechanisms. However, it is possible that we only measure the subjects' understanding of drawing programs, not their desires for undo in general. We do not think this is likely because almost all of the subjects are familiar with typical office applications – the subjects are not likely to have been confused by a task related to a drawing program.

Note also that we purposefully did not tell the subjects that the task represented a computer application – and hand-drew the figures so as not to accidentally encourage them to think in that way. We think this makes the results more appropriate – instead of measuring what subjects think existing computer applications do, we more accurately measure how they want undo to behave.

RESULTS AND DISCUSSION

For the task where Step 4 (Color Circle) is to be undone, all three undo models (linear, script selective, and cascading selective) produce the same result. The point of this task is to ensure that subjects understood the definition of undo correctly. One subject misinterpreted the task in a similar way to one of the pilot study subjects (though the two subject groups were disjoint.) Namely, the subject drew the shapes in reverse order. That subject's data is therefore not used in the remaining analysis.

After coding each of the responses of the remaining 28 subjects based on our interpretation of whether they performed linear, script, or cascading undo, we find that 1 used linear, 7 used script, and 18 used cascading, while 2 gave a response we could not code (see Figure 2). The graph seems to sug-

	Cascade	Script
Min	0.3261	0.800
First Quartile	0.9028	1.091
Median	1.1670	1.750
Mean	1.5400	1.815
Third Quartile	1.6250	2.236
Max	5.2500	3.500

Table 3. Summary of time ratio (T_1/T_4) for cascade and script subjects

gest a large difference in the frequencies of these different responses, with cascade being preferred in our sample almost two-thirds of the time, and linear preferred almost not at all. A χ^2 test confirms that it is highly likely that the real distribution of the frequencies is not uniform between the four possible values ($\chi^2 = 26$, df $= 3$, p $= 0.00001$). We therefore reject null hypothesis $H_0(1)$ with alpha level at 0.05.

However, this statistical test does not tell us if linear is really not preferred as the graph seems to suggest. To test this, we compare the frequency of choosing linear to the frequency of choosing script and cascade, respectively. In the first case, a χ^2 test finds that the apparent difference in frequencies is real ($\chi^2 = 4.5$, df $= 1$, p $= 0.01695$, one-sided). However, since there are only 8 data points for linear and script combined (each has an expected value of only 4), and χ^2 is generally considered valid only when none of the expected values is less than 5, this test is of questionable use. The comparison between linear and cascade, however, does show a significant difference ($\chi^2 = 15.21$, df $= 1$, p $= 0.00005$, one-sided). We therefore reject null hypothesis $H_0(3)$.

The graph also suggests that cascading is preferred over the script model. To determine if this effect is real (i.e. not due to chance), we again use a χ^2 test, which also finds a statistically significant difference ($\chi^2 = 4.84$, df $= 1$, p $= 0.02781$). We therefore reject null hypothesis $H_0(4)$.

Note that since we have performed multiple χ^2 tests, we should correct for inflation of α to ensure that we reject null hypotheses because they are unlikely to be true, instead of simply as the result of a statistical "fishing expedition". Table 4 summarizes the statistical results, with p-values adjusted using the R language [11], and according to the procedure defined by Benjamini and Hochberg [3]. Note that we still reject $H_0(1)$, $H_0(3)$, and $H_0(4)$.

These results indicate that for the task used in this study, cascading undo is most natural. However, we are also interested in how much effort it takes subjects to predict system behavior using the two selective models. If cascading is more natural, but script is easier for users to predict, it might be preferable to use a script model over cascading. It is for this reason that we timed the subjects in this study. For each subject, we have recorded the time they took to undo step 4 (T_4) along with the time to undo step 1 (T_1). Because undoing step 4 should be easy – it is the last step done and

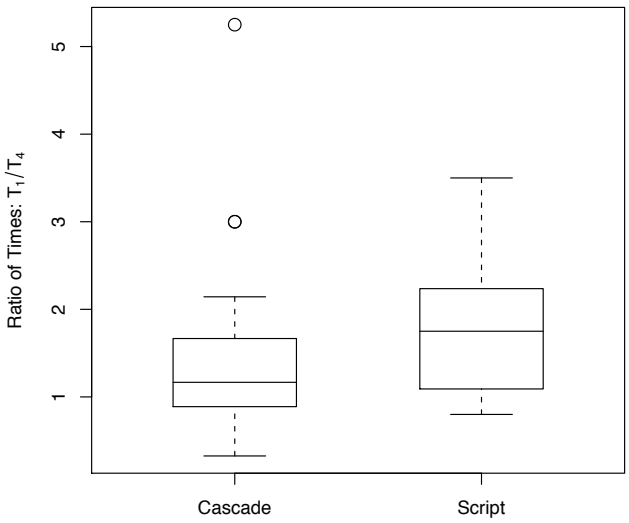

Figure 3. Summary of time taken for subjects to "execute" cascade and script undo models. The y-axis is the ratio of the time the subject took to undo step 1 (T_1) versus the time that same subject took to undo step 4 (T_4). The time to undo step 4 is therefore used as a baseline.

therefore no other steps depend on it – we expect that this should take subjects less time to complete than undoing step 1. Therefore, we compute the ratio T_1/T_4 for each subject. Table 3 summarizes this ratio for two groups: those subjects that executed cascade and those that executed script. Figure 3 shows a box plot of the same data.

The figure seems to suggest, aside from some outliers, that cascade takes less time to determine than script. However, because we did not randomly assign subjects to cascade and script "treatment groups", we cannot say whether this effect is real[2]. Furthermore, note that the recorded time includes both decision and execution time. That is, we did not differentiate the time it takes to think out the result of an undo from the time it takes to draw the result. Future studies can address this.

CONCLUSION

The community has understood for quite some time that there are better ways to support undo in computer applications than to force users to undo tasks in strictly linear fashion. Many applications now support a linear undo mechanism in which any user action can be selected for undo, and upon selection the system automatically undoes all actions that occurred after the selected one. This is supported in implementation by keeping a full history of actions performed instead of only keeping a copy of the most recent action (or the state that resulted from it). However, this is still a linear undo model and it is too restrictive.

[2]Even though this is not an experimental design with a manipulated variable, we have run a t-test to compare the real means of the two distributions (T_1/T_4 for cascade and T_1/T_4 for script) and we cannot reject the hypothesis that the means are the same (p=0.5526).

Hypothesis	χ^2	Degrees of Freedom	One-sided?	p	Adjusted p	Rejected?
$H_0(1)$ (omnibus)	26	3	no	0.00001	0.00004	yes
$H_0(2)$ (l vs. s)	4.5	1	yes	0.01695	0.02260	no
$H_0(3)$ (l vs. c)	15.21	1	yes	0.00005	0.00010	yes
$H_0(4)$ (s vs. c)	4.84	1	no	0.02781	0.02781	yes

Table 4. Summary of statistical results. In the last three hypotheses, l, s, and c stand for linear, script, and cascade, respectively. Hypotheses are rejected with $\alpha = 0.05$ for p-values adjusted according to Benjamini and Hochberg [3]. $H_0(2)$ is not rejected because χ^2 is not applicable when expected values are less than 5.

To support a less restrictive model of undo, others have introduced selective undo mechanisms that allow users to select arbitrary actions for undo. One primary way that these mechanisms differ is in their handling of dependencies between those actions. In a pure script model, the dependencies are ignored. In other models, if dependencies are noticed, a warning to the user is suggested as a system response [1]. The cascading selective undo of Cass and Fernandes, however, uses those dependencies to cause dependent actions to be undone along with the selected action.

Which of these models is most natural? Does it depend on the kind of application? Does it depend on the kinds of dependencies? In this study, we have started to address these questions. In the context of familiar applications such as drawing programs, and for data dependencies between actions, the cascading model is more natural than the script model to users. Furthermore, it is much more natural than the linear model.

The data suggest that linear is also not as natural as script, but further study is needed to assess this. The data from this study also suggest that the cascade model is easier for users to think about, but a controlled experiment is needed to test this hypothesis.

FUTURE WORK

There are several avenues for future work suggested by the work presented here. The present study suggests that linear undo is not preferred over selective models and that users prefer a model that makes use of dependencies between user actions, in at least one context. The community could profit from further study to determine if the results extend to other contexts. For example, in applications in which non-data dependencies exist between user actions, will users still find cascading to be natural? There is reason to believe that users know that some actions must necessarily precede others, but we think empirical evidence is needed. We therefore plan an experiment within a more dependency-rich domain.

The results of the present work can safely be generalized only to other relatively simple applications. We have conducted a pilot study with a more complete application – presentation software – and plan further experiments to ascertain the breadth of appeal of the different selective undo models.

Note also that this work does not fully address the cognitive load involved in using a non-linear undo model. One can argue that an advantage of linear undo is that the user need not think about dependencies in order to predict what will happen when a particular command is undone. However, it is not known how much thinking is required by the other models. We therefore plan an experiment that will time subjects in predicting the results of undo using different models. The timing results from the present work are suggestive, but a carefully controlled experiment aimed at precisely that point will give us more confidence in the results.

All of these planned studies can be designed as experiments in which we manipulate a variable to define treatments and randomly assign subjects to treatment groups. This is different from the current study, which was observational in nature. Now that we better understand the space of possible undo models that users will find natural, we can focus on these. One possible experiment design, which could again be carried out with pen and paper, would involve the following steps with each subject:

1. Assign the subject randomly to a treatment group.
2. Give the subject a list of actions that have been carried out, along with a depiction of the resultant state (similar to the instruments used in this study).
3. Show the subject a depiction of a state that results from undoing step x according to the treatment's undo model.
4. Ask the subject to rank (on a Likert scale) the correctness (or naturalness) of the resultant state, given that it results from undoing step x.
5. Time the subject's response time.

With this design, we could more directly compare the undo models of interest, and ensure that we have an even amount of data for each of the models, thus enabling us to draw conclusions more reliably. Note that this experiment design would measure something different than the current study – the current study determines what model users want or find natural, while the proposed new study compares two or more models on specific variables of interest. Both studies are needed.

And lastly, if and when we have amassed a series of experiments that suggest that cascading selective undo is valuable and needed, we will implement a cascading model in a representative application to directly address the implementation feasibility of the approach – we have a good idea of how this approach can be implemented, but the present work was

needed to determine whether the work is usable by users before we worried too much about how hard it will be to implement for programmers.

ACKNOWLEDGMENTS

We thank Brian Postow and Linda Almstead for allowing us to enter their classrooms to recruit subjects for this study. We also thank the subjects for their participation.

We thank Philip Gray, who inspired us to undertake this study. We also thank Daniel Burns and George Bizer for their help in analyzing the current study and designing future experiments.

The empirical study presented here was submitted, as proposal number 993, for approval to the Human Subjects Institutional Review Board of Union College, which approved the study.

REFERENCES

1. G. D. Abowd and A. J. Dix. Giving undo attention. *Interacting with Computers*, 4(3):317–342, 1992.
2. J. E. Archer, Jr., R. Conway, and F. B. Schneider. User recovery and reversal in interactive systems. *ACM Transactions on Programming Languages and Systems*, 6(1):1–19, 1984.
3. Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.
4. T. Berlage. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Transactions on Computer-Human Interaction*, 1(3):269–294, 1994.
5. A. G. Cass and C. S. T. Fernandes. Modeling dependencies for cascading selective undo. In *IFIP INTERACT 2005 Workshop on Integrating Software Engineering and Usability Engineering*, Sept. 2005.
6. D. Chen and C. Sun. Undoing any operation in collaborative graphics editing systems. In *Proc. of Intl. Conf. on Supporting Groupwork (GROUP)*, pages 197–206, 2001.
7. B. A. Myers and D. S. Kosbie. Reusable hierarchical command objects. In *Proc. of the ACM Conf. on Human Factors in Computing (CHI 96)*, pages 260–267. ACM Press, 1996.
8. A. Prakash and M. J. Knister. A framework for undoing actions in collaborative systems. *ACM Transactions on Computer-Human Interaction*, 1(4):295–330, Dec. 1994.
9. M. Ressel and R. Gunzenhäuser. Reducing the problems of group undo. In *Proc. of Intl. Conf. on Supporting Groupwork (GROUP)*, pages 131–139, 1999.
10. C. Sun. Undo any operation at any time in group editors. In *Computer-Supported Cooperative Work (CSCW)*, pages 191–200, 2000.
11. W. N. Venables, D. M. Smith, and the R Development Core Team. *An Introduction to R*. www.r-project.org.