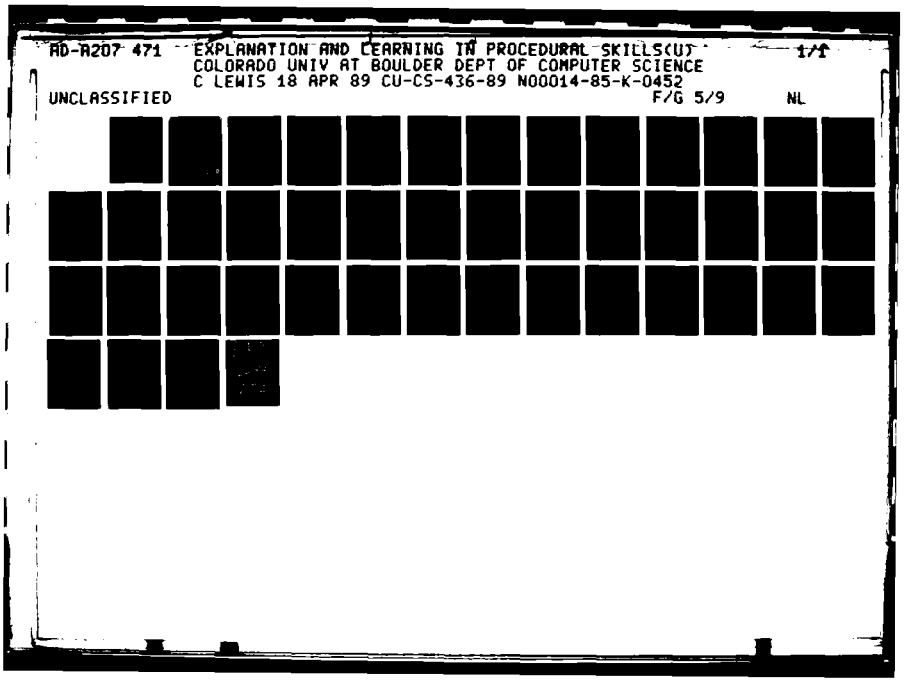
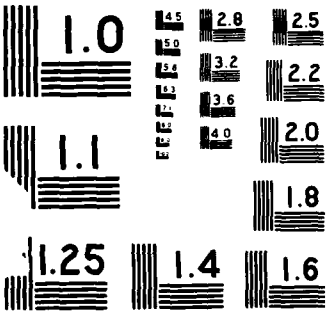

AD-A207 471 EXPLANATION AND LEARNING IN PROCEDURAL SKILLS(U) 1/1
COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE
C LEWIS 18 APR 89 CU-CS-436-89 N00014-85-K-0452
UNCLASSIFIED F/G 5/9 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

THIS FILE COPY

1a	**AD-A207 471**
1b RESTRICTIVE MARKINGS	None
2a	
3 DISTRIBUTION / AVAILABILITY OF REPORT	Distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	CU-CS-436-89
5 MONITORING ORGANIZATION REPORT NUMBER(S)	
6a NAME OF PERFORMING ORGANIZATION	University of Colorado
6b OFFICE SYMBOL (If applicable)	
7a NAME OF MONITORING ORGANIZATION	Office of Naval Research (code 1142CS)
6c ADDRESS (City, State, and ZIP Code)	Campus Box 430 Boulder, CO 80309
7b ADDRESS (City, State, and ZIP Code)	800 N. Quincy Street Arlington, VA 22217
8a NAME OF FUNDING / SPONSORING ORGANIZATION	Office of Naval Research
8b OFFICE SYMBOL (If applicable)	code 1142CS
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	N00014-85-K-0452 (R&T 442C009)
8c ADDRESS (City, State, and ZIP Code)	800 N. Quincy Street Arlington, VA 22217
10 SOURCE OF FUNDING NUMBERS	PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)

Explanation and Learning in Procedural Skills -- Final Report

12 PERSONAL AUTHOR(S)
Clayton Lewis

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM 1 Sep85 TO 31Dec88	89/4/18	41

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report summarizes the findings of an investigation into the role of explanations in learning procedures. Experimental and theoretical results from studies of the analysis of examples and generalization methods, and issues remaining open, are presented. Three previously undistributed technical reports are included.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
Dr. Susan Chipman	202/696-4318	code 1142CS

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—607-044

Explanation and learning in procedural skills--
Final Report

Clayton Lewis
Institute of Cognitive Science and
Department of Computer Science
Campus Box 430
University of Colorado
Boulder CO 80309

clayton@sigi.colorado.edu
(303) 492 6657

April 17, 1989

Abstract: This report summarizes the findings of an investigation into the role of explanations in learning procedures. Experimental and theoretical results from studies of the analysis of examples and generalization methods, and issues remaining open, are presented. Three previously undistributed technical reports are included.

Acknowledgements: This research was supported by the Office of Naval Research under Contract No. N00014-85-K-0452; related work was supported by US West Corporation, AT&T, Hewlett-Packard, and the Institute of Cognitve Science. I thank Steven Casner, Victor Schoenberg, Mitchell Blake, D. Charles Hair, and Cathleen Wharton for their assistance.

Introduction.

My goal in preparing this report is to supplement the published output of the project. Accordingly, I will not recap at length work that has been published, but will instead aim to provide an account of those aspects of work not elsewhere reported that may be of interest to fellow researchers in the area, and to outline work too recent to have been reported more fully. I also attempt to provide an overview of the overall scope of the project, too broad to be appropriate in reports of specific findings, but perhaps of value to readers undertaking their own attack on issues discussed here.

Background: Phenomena to be Explained, Basic Issues to be Dealt with, the Original Hypothesis.

The EXPL project was planned to investigate two interrelated issues, one arising from the study of learning to use computer systems, and one, more general, visible as one of the enduring threads in studies of thinking and learning. Subjects in studies of computer systems were observed to make up *explanations* of things they saw. Why were they doing this? Did it have some utility? In general, psychologists at least since Wertheimer have asked, what is the point of understanding something?

We observe that understanding, intuitively identified, facilitates learning, but how and why? Study of the particular outcropping of explanations in the human-computer interaction domain seemed a promising way to address the more general issue, and the EXPL project was chartered to do this.

The starting point for the work was the chapter "Understanding what's happening in system interactions" (Lewis, 1986b) in Norman and Draper's *User Centered System Design* volume. This chapter sketched an account of how particular episodes for which Lewis and colleagues at IBM had collected protocols might be explained. The notion of explanation that was invoked was not rigorously defined, but centered on establishing connections between user actions and system responses associated with them. It was suggested that such analyses of procedures could be produced by a combination of bottom-up heuristics and top-down application of prior knowledge. The first order of business for the EXPL project was to build a simulation model of this process, to determine whether these ideas could be made operational, and to elaborate specific hypotheses about how such analyses might be carried out by human learners.

The EXPL model.

The original conception of the EXPL model was as a set of graded*constraints* on explanations of sequences of events. For example, an explanation which accounted for all aspects of an event was to be preferred to one leaving some unaccounted for. The constraints would constitute a kind of axiomatic description of what made a good explanation that was divorced from any particular implementation scheme. I attempted to build a PROLOG program that would construct explanations directly from statements of the contraints. I failed to produce a workable program along these lines and changed approach to one in which I programmed in PROLOG specific methods for building explanations satisfying the constraints given a sequential presentation of a sequence of events. In hindsight I think the direct constraint approach offers advantages justifying another attempt along the original lines, but using an implementation medium more suited to this task than PROLOG. I return to this point in considering future work at the end of this report.

The basic EXPL model proved quite easy to implement, once the approach of building the analysis sequentially was adopted. Descriptions of resulting model can be found in Lewis (1986a, 1988a), Lewis, Casner, Schoenberg and Blake (1987); I include a summary here.

In outline, EXPL consists of three sequential phases. The first phase, *encoding*, is performed manually. Events in the sequence to be analyzed are represented as simple sequences of almost arbitrary tokens. The only restrictions on the choice and use of these tokens are that tokens intended to represent things which must be present to be referred to, such as entries in menus displayed on the screen, must be marked, and events which make such tokens present must be begin with the reserved token SHOW.

Encoded events are marked as representing user actions or system responses, and are delivered to the *analysis* phase in chronological sequence. This phase applies a small collection of heuristics which place causal links between user actions and particular tokens in the representation of subsequent system responses. Such links might indicate that a token representing a particular operation, say DELETE, was apparently controlled by a particular user action, such as TYPE DELETE, or that a token representing a particular object, such as SHOE, was specified by the user action CLICK SHOE. Heuristics also place prerequisite links which trace what prior system response made the referent of SHOE available to be acted on (so that user actions which caused this action can be tied into any plan involving CLICK SHOE.)

The output of the analysis phase is passed to the *generalization* phase, whose task is to produce plans for accomplishing novel goals on the basis of what was learned from any examples that have been seen and analyzed. Originally EXPL used only one generalization method, *synthetic* generalization, in which the links placed in analysis are interpreted as describing preconditions and results for specific operators seen as user actions in examples. The generalizer is just a simple planner which attempts to accomplish a stated goal using this repertoire of operators. Later, following the work of Anderson and Thompson (1986) a generalizer based on the PUPS analogical generalizer was built and incorporated. This made it apparent that the problem of analyzing examples can be separated from the problem of generalizing them, and that many different generalization schemes could be supported on a common base of analysis, a point developed in Lewis (1988a). Later still Cathleen Wharton built a third generalizer that converted EXPL analyses into productions in the form used in Polson and Kieras's Cognitive Complexity Theory (1985)

Almost at once it became apparent that the small number of heuristics for analysis building that the model incorporated were capable of analyzing surprisingly complex event sequences, with essentially no knowledge of the semantics of the sequences. This fact, coupled with difficulty in identifying pertinent background knowledge in human subjects (discussed below) led to postponement of efforts to incorporate the originally planned top-down processes in the EXPL model.

The original heuristics, identity, loose-ends, and previous action, were joined by others during the course of work on the model. Obligatory previous action, which requires that any system response have at least one link to the immediately previous user action, was added first. A number of variants of identity were incorporated, to handle cases in which components of events shared features without being strictly identical. For example, all entries in a menu might be erased by an action whose description refers to the menu but not to each entry on it. The "group identity" heuristic allowed the encoding of the menu items to use a common "stem", also used for the menu itself, which allowed EXPL to trace the relationship between the menu and the items. Another variant of the identity heuristic, which might be called the back-chaining heuristic, was proposed by Catherine Marshall. It allows causal links to be drawn between system responses that share elements, rather than just between system responses and user actions. Consider the following interaction,

an encoding of an interaction with a telephone system.

S: RINGSTYLE SMITH NORMAL
U: STAR 5
S: RINGSTYLE BROWN NORMAL
U: STAR 6
S: RINGCOUNT BROWN 3
U: STAR 1
S: RINGCOUNT BROWN 1

The problem here is to determine what user actions were responsible for specifying BROWN, RINGCOUNT, and 1 in the final system response. The 1 can be dealt with simply with the identity heuristic, but where do the other components come from? The back-chaining heuristic looks for situations in which consecutive system responses share components, and links them. Thus RINGCOUNT in the last response is linked back to RINGCOUNT in the previous response, where it is tied to the previous user action, STAR 6. BROWN is chained back through two previous system responses to the second response, where it is tied to STAR 5. Thus the analysis recovers the facts that STAR 5 determines the party, and STAR 6 the data item to be dealt with for that party. More work is needed to determine just how STAR 5 and STAR 6 would be used to select a particular party and data item, but this is progress.

A complete analysis of this example remains beyond the scope of EXPL. The natural representation to use in reasoning about it is a table whose rows are parties and whose columns are data items. In this framework STAR 5 moves down the rows and STAR 6 moves across the columns. EXPL has no way of devising such a representation, or using it in its representation of actions and their outcomes. There are two parts to this problem. First, the idea of using a particular data structure to organize the interpretation of an interaction has to be proposed by some heuristic. It is not clear on what basis such heuristics should act (and for that matter it is unclear whether human learners can construct such hypotheses without specific hints to do so.) The heuristics would operate at the encoding phase, rather than in the analysis phase, so that the system responses could be redescribed in terms of motion along rows or columns. Second, the generalizer has to be able to devise procedures for finding given rows or columns from examples. This should work out once the system responses are cast in terms of operations on the underlying data structure.

Casner (Casner and Lewis 1987) built a somewhat similar extension to EXPL to cope with interactions involving hidden events. These are interactions in which critical events occur behind the scenes, and must be inferred from the system responses that are explicitly signalled. A common example is the cut and paste interaction, in which cut causes not only a visible deletion but and invisible copying of the deleted material into a buffer, from which it is copied by the paste operation. Casner devised a collection of recognition heuristics which identified certain possible classes of hidden events based on their surface symptoms (such as changes in the system's response to identical commands.) If one of these heuristics was found applicable

then a revised encoding of the interaction, incorporating the proposed hidden event, was constructed.

These are by no means simple extensions to EXPL, and highlight the limitations imposed on the current EXPL system by its separation of encoding and analysis. The success of analysis is severely constrained by the encoding it starts from. Further, it seems very likely that the appropriate encoding of an interaction is influenced by the analyses various possible encodings support. I report below some data supporting this claim. This means that EXPL's simple serial staging of encoding and analysis is fundamentally incorrect, and should be replaced by a scheme in which the construction of encodings and corresponding analyses interact.

Another limitation of the original EXPL model that emerged in applying it to a wide range of examples was the reliance on a single encoding for any given event. Later versions incorporate a system of annotations, similar to those used in the representational scheme of PUPS (Anderson and Thompson 1986), that permits a given event to be encoded in many alternate ways. The need for this arises when the same event may be connected to neighboring events in multiple ways. Consider the system response of highlighting a particular spreadsheet cell, say B3. If this is preceded by the user action of clicking on B3, the description of cell B3 as such is obviously crucial in tying this even to its predecessor. But suppose the same event is instead preceded by pressing RETURN when cell A3 is highlighted. Now the encoding must bring out the fact that cell B3 is the one immediately below A3. Similar cases can be constructed in which one encoding is needed to tie an event to its predecessor and a different encoding is needed to tie the very same occurence of the event to its successor, so that a scheme which simply allows alternative, but not coexisting encodings, is inadequate. In the PUPS-like representation one encoding is chosen, say B3 in the example, but annotations are added containing the information that B3 is below A3 and above C3.

Another limitation of EXPL brought out by attempted applications, and never satisfactorily dealt with, is its inability to segment long event sequences in a principled way. The need to do this arises in connection with the loose ends heuristic, which attempts to tie together unexplained user actions with unexplained system responses. Some means of limiting the scope of the heuristic is needed, to make sure that loose ends are tied up only within what can be considered to be a coherent episode, and not reaching across indefinitely long intervening sequences of events. The issue is not just that these long-distance loose-ends ties are usually wrong, but that they prevent the usually correct previous action connections from being formed. One heuristic criterion we explored was preventing loose-ends links from crossing identity links, but this proved error-prone in breaking up some sequences from real demonstrations.

A related unsolved problem is that of identifying the goal of a subsequence of actions. If this can be done reliably then the segmentation problem above can also be solved by blocking the connection of loose ends across intervening goals. But goal indentification is problematic for EXPL's largely semantics-free methods. Consider

the appearance of a menu. In the normal case this is only an intermediate goal of the actions that lead to it, since the intent is to make some selection from the menu. So tying together loose ends across the display of the menu is perfectly sensible (and often necessary). But it can happen that the display of a menu is an end in itself, as when the intent is to learn what is on the menu rather than to select from it.

Another connection between the correct operation of loose ends and the correct identification of goals is that determining what are really loose ends depends on determining where the major goals of actions are. Consider the effect of selecting an item from a menu. Usually there are two: some action is selected, but also the menu disappears. Correct analysis usually requires that the disappearance of the menu be treated as an unimportant side effect of the selection, so that the selection remains eligible as a loose-ends cause of some later system response. But what principled basis is there for this determination? There are clearly cases in which the sole, and central, effect of an action is to cause something to be deleted from the screen. How can the disappearance of a menu be discriminated from the purposeful deletion of some other item? Concretely, in viewing a Macintosh demonstration, how could one discriminate selecting the go-away box on a window from selecting something from a menu? As with other limitations of EXPL it seems that the semantic depth of EXPL's knowledge must be increased to cope with such problems.

Empirical Studies.

Top-down analysis. The original conception of EXPL presumed that background knowledge would play a significant role in analyzing examples, with top-down fitting of expected patterns complementing the bottom-up action of the heuristics. We attempted to gather thinking-aloud protocols in which this background knowledge could be identified, as a first step toward filling in this aspect of the model. While we were successful in collecting what seemed like appropriate protocols we failed to find any evidence of the level of background knowledge in which we were interested.

We devised a fictitious problem setting which would motivate subjects to generate a description of how a system might work based only on a very general specification of its function. Subjects were told that they were to brief an emergency team responding to a disaster in a chemical plant. The team needed to control certain valves in the system, but no specific documentation on the computer system which operated the valves was available. Subjects were to do whatever they could to prepare the emergency team for their task. We expected that subjects would provide a decomposition of the required task into necessary specification steps, whose general nature could be described but whose order and particular form would be unknown, as in "You'll have to specify the valve in some way, and you'll have to indicate what you want to do to it, like open or close."

No subject gave us this level of discussion. Instead subjects enumerated specific schemes for the task based on systems they had used: "Well, if it is like UNIX you'd have commands followed by options and then the name of a valve." Even when we

sought out subjects with very little computer experience we did not escape this very concrete approach. One subject related the tasks to a video game he had played and another remembered a program he had worked with in a science lab.

The failure to find evidence for general expectations about how tasks would be performed, coupled with the unexpected success of EXPL's unaided bottom-up heuristics, led us to defer incorporating top-down processes in EXPL. It is of course possible that more abstract top-down schemata than those that appear in the protocols are actually used. But it also is plausible that the protocols are pointing in the right direction, and that top-down processing is guided mainly by resemblance to very specific precedents. This is an area in need of further exploration.

Tests of heuristics. Lewis (1988a) reports experimental tests of whether the heuristics in EXPL are used by people, but the testing did not include all the heuristics proposed for EXPL. Reasonably strong evidence was found for the identity and loose-ends heuristics, but no good test was devised for previous action or obligatory previous action. As noted in that report testing of heuristics is complicated by the dependence of the action of the heuristics on the details of the encoding of events. Further, at that stage of the project we lacked any adequately rigorous definition of precisely what these heuristics were, outside of the details of the implementation of the EXPL model. With the definitional framework provided by the *control* notion (discussed below) more informative empirical study of the heuristics would be possible.

Role of analysis in learning from real demonstrations. In parallel with the development of the EXPL model we undertook to investigate the extent to which ease of analysis of examples in EXPL corresponded with the ease of learning from those examples in realistic learning settings. The reports by Schoenberg and Lewis, and by Lewis, Hair, and Schoenberg prepared some time ago but issued with this report, describe these efforts. The approach used was to ask subjects to view a video recording of a demonstration of a real software system, and then undertake tasks related to those demonstrated. The recorded demo was encoded and anlyzed by EXPL, so that we could determine where the difficulties were as predicted by EXPL, and could then compare these with problems encountered by the subjects. The investigations were only partly successful. EXPL was able to detect a few problems in the interfaces studied which did show up in subjects' performance. But we were hampered by a number of problems. (1) Demonstrations are hard for subjects to observe, especially when, as in the systems we studied, critical events may occur on the screen or on the mouse (when a button is pressed.) We used split-screen presentation and enhanced sound effects to try to counter this problem. (2) It seemed to us subjects did not invest very much in really following the demonstration. We manipulated instructions to try to influence them, but this probably indicates a real limitation of the EXPL model: people are probably not as assiduous in extracting cues from examples as EXPL is. (3) It was difficult to relate problems in performing tasks with specific episodes in the demonstration. Many operations were demonstrated more than once, and some operations could be performed in ways other than those shown in the demonstration. As a result there was uncertainty about just where a difficulty in analyzing the demo should show up in performance. (4) We did not

create a control in which subjects attempted tasks without having seen the demo. This should be remedied in further attempts.

Interaction of encoding and analysis. As noted above it seems unlikely that EXPL's strict serial separation of encoding and analysis can be correct. We devised a situation in which we expected the availability or unavailability of a good analysis, determined by context, to influence how a given event is encoded. We exploited an ambiguity in describing operations on objects in which an operation that is shown acting on objects of a particular kind can be seen as applying only to objects of that kind or on any objects. In EXPL this difference shows up in the encoding of the effect of the operation, so we can look for influences of analysis on encoding by introducing contextual variations that should affect analysis and looking for differences in the interpretation of the operation.

Specifically, one group of items presented subjects with two consecutive screens, the first containing two X's and the second blank. The operation intervening could be thought of either as deleting X's or as clearing the screen. The intervening command was either PX or PY. A subsequent probe item asked subjects to indicate what the affect of applying this command to a screen containing an X and a Y would be. In EXPL the command PX, encoded as P X, together with an encoding of the system response as something like DELETE X, leads to a reasonable analysis, while the command PY with the same encoding of the response does not. Conversely, PY fits nicely into an analysis with encoding CLEAR, while P Y does not. Thus if the encodings participants choose are influenced by the associated analyses we predict that participants will expect the command PX to remove the X and not the Y, but participants who saw the command PY in the same context will expect it to delete both the X and the Y. That is, the encoding of the event of the two X's disappearing will be influenced by the form of the command that is seen to cause it, something not possible in EXPL's serial treatment.

This prediction was borne out for these items and for similar items in which a doubling operation rather than a deletion operation was used. Significantly more participants assigned a letter-specific interpretation to commands for which an identity cue was available in analysis than commands for which no such cue was offered.

This finding suggests that the encoding of events for analysis cannot be separated from the analysis process itself. The multiple encoding scheme introduced in later versions of EXPL would allow for this, so that an initial, analysis-independent encoding could be modified to reflect the results of analysis. This has not yet been undertaken.

*Role of learning in shaping language structure.*The debate between empiricists and rationalists about the acquisition of language has been limited by the poverty of our conceptions of learning. As long as Skinner could rely only on simple inductive learning methods it was easy for Chomsky to attack the idea that language could be effectively learned, and to argue that much linguistic structure could not be learned.

Anderson (1983, p.301) took up Skinner's argument in the context of a more elaborate learning theory, proposing that linguistic structure reflects the scope of effectiveness of a variety of learning mechanisms. But these mechanisms are still essentially inductive in Anderson's scheme, with some specific a priori constraints added. The advent of *analysis-based* , non-inductive learning methods such as those embodied in EXPL offers the prospect of reframing this old argument. Learning mechanisms that exploit causal analysis and analogy may have a better chance of accounting for the observed structure of language than their predecessors.

We attempted to investigate the ability of learning mechanisms to shape linguistic structure by adapting Bartlett's repeated transmission paradigm. We devised random command languages (with some structure built in as described below) and asked participants to study examples of command-outcome pairs and then generate commands to produce new outcomes supplied by us. Thus each participant produced a new corpus of examples, based on his or her efforts to extrapolate the examples seen to cover new outcomes. These generated corpuses were presented to new participants in the same manner as the original random languages, and these second-generation participants were again asked to produce commands for outcomes they had not seen, in this case the same outcomes as appeared in the original random corpuses. Thus each original random corpus spawned a succession of derived corpuses, each resulting from a participant's attempt to extrapolate the examples seen to new outcomes.

While many kinds of structure might be introduced into the derived corpuses by this extrapolation process we expected EXPL's robust identity heuristic to have an easily detectable effect. We expected any identity relations that appeared between commands and outcomes to be salient and well-recalled, and hence to be preserved where possible in the extrapolated corpuses. To prime the pump we ensured that each random corpus had a proportion of identity cases in it. Further, we expected participants to introduce new identities as they attempted to generalize from examples which contained identities. So we expected the number of identities in successive corpuses to increase. Along with this we expected the success of participants to extrapolate accurately, that is, to provide the same command that was presented with a given outcome in the corpus just before the one they saw, to increase.

Analysis of results focuses on the corpuses appearing at plies 0 (the original corpus), 2, and 4. Under the procedure used, all these corpuses have the same set of outcomes, so the commands supplied by subjects can be directly compared. Increase in accuracy can be gauged by the number of command tokens in plies 2 and 4 that agree with the corresponding commands in ply 0 or 2. The median number of correct tokens at ply 2 was 2.5, while at ply 4 it was 6.5. Of 22 sets of corpuses 15 showed an increase in accuracy and 5 a decrease, a preponderance significant at the .05 level.

This increase in accuracy cannot, however, be attributed to identity cues. There was no increase in the median number of identities across plies 0, 2, and 4; numbers of

identities actually descreased, but not significantly. The Spearman correlation between the increase in number of identities from ply 0 to ply 2 and increase in accuracy from ply 2 to ply 4 was .22, not significant (n=22).

Other aspects of the corpuses did change in a way that seems to have contributed to the increase in accuracy. Some output tokens appeared more than once; when participants had to assign a command to these in the following ply their accuracy was influenced by whether the multiple occurences were associated with a consistent command token or with different tokens. Proportion correct for tokens with multiple consistent commands was .50 at ply 2 and .86 at ply 4 (medians; difference not significant), while those with inconsistent associated commands had median proportion correct of .00 in each case. This difference of accuracy between output tokens with inconsistent and consistent commands (significant at ply 2 but not ply 4 by sign test) suggests that corpuses with greater consistency would be reproduced better, so that increases in consistency during the repeated transmission process would lead to improved accuracy. This is so: the median proportion of output tokens with multiple occurences which were associated with consistent commands increased from .00 to .45 to .66 across plies 0-4; the increase at each step is significant by the sign test. The Spearman correlation between increase in consistency from ply 0 to ply 2 and increase in accuracy from ply 2 to ply 4 was .40, significant at .05 (one-tailed).

Increase in consistency cannot account for the entire increase in accuracy, however. Some output tokens were not seen as outputs at all in the previous ply, and so could not be reproduced without some form of extrapolation. Identity is one means of doing this; even though there was no increase in identities some correct reproductions did exploit identities (a mean of .32 tokens at ply 2 and .55 at ply 4 were correctly reproduced this way). Another means of reproducing the commands for "orphan" tokens, those which did not appear as outputs in the previous ply, is to assume a reversible connection between command and output. If output O was not seen as an output at the previous ply, but was seen as a command, with output O', then use O' as the command to obtain output O at this ply. These "reversals" accounted for a mean of .09 correct reproductions at ply 2 but .59 at ply 4; this increase is significant at the .05 level by the sign test. This assumption of reversibility should result in an increase of cases within a corpus in which a command-result pair O-O' also occurs as a reversed pair O'-O. The mean number of such reversals did increase across plies 0-4 from .09 to .90, the increase being significant at .05 by the sign test. This increase in reversals did not correlate significantly with increased accuracy, however, even though (as mentioned above) the reversals were responsible for a small but growing number of correct reproductions.

In summary, the repeated transmission study did demonstrate increases in the learnability of corpuses, but the identity heuristic appears to play only a minor role in this, being used to produce some commands but not leading to an overall shift in the structure of the corpuses. A tendency to assign consistent commands to output tokens that occur more than once seems to have been more important: the degree to which this change occurred in a sequence of corpuses proved to be correlated with

increased accuracy. An increase in the number of reversible command-output pairs also occurred and contributed to the increase in accuracy.

These results bear out the plausibility of Anderson's proposal that linguistic structure could result from the action of learning and retransmission, though they do not implicate the sort of learning mechanisms involved in EXPL. Increased learnability did result, and was associated with structural change in the corpuses.

Analysis and recall. Just as we expected (and now have demonstrated) that analysis could affect how events are encoded, we expected that analysis could shape how sequences of events were recalled. Mack (1984) had observed that participants who viewed demonstrations of text editor operations sometimes interpolated imaginary events that made the sequence of events more sensible to them ("I guess I missed it but there must have been a command to make it move that text over, " when no such command was shown because the system was in insert mode.) In other protocol studies we had observed that participants would produce significantly distorted reviews of what they thought they had seen in attempting to explain what was happening. The EXPL model makes specific predictions about what analyses of human-computer interactions should be acceptable, and hence of what distortions would be needed when recalling events to make them seem sensible.

To test these predictions we constructed two deliberately odd commands. One command mentions two letters as arguments and deletes only one of them. The other command mentions one letter but deletes two. We included these commands, with their outcomes, in event sequences which we asked participants to study. After a delay we showed them the screen state they had seen just before the odd command, and the screen state shown just following, and asked them to recall what command had intervened in the sequence they had studied. While most participants recalled the command correctly, several participants "recalled" a cleaned-up version of the odd command. The command name was recalled correctly but the argument structure was adjusted to conform to the expected analysis. Of 54 participants 12 produced these specific predicted distortions; there were 2 other distortions not predicted.

The implication of this finding is that systems that are difficult to analyze on EXPL lines may be difficult to learn for two reasons. Not only may the difficulty of analysis make generalization difficult, but hard-to-analyze sequences may simply be recalled inaccurately.

Retention and generalization mode. Another possible linkage between recall and generalization concerns the distinction between "superstitious" and "rationalistic" generalization, as defined in Lewis (1988a). Superstitious generalization preserves any features of examples which are not understood, while rationalistic generalization preserves only those features which are understood. As Lewis (1988a) argues, differing generalization mechanisms may naturally behave in one of these manners or the other. Because of the dependence of superstitious generalization on

retaining uninterpreted features of examples, one might expect that retention demands should affect generalization mode: superstitious generalization should be more difficult as retention becomes more difficult. The availability of semantic interpretations for those features of examples needed for rationalistic generalization might favor rationalistic generalization as retention becomes more difficult.

To test this idea we devised an example interaction with an unnecessary step, to which we expected many participants would attach no interpretation. After seeing this example some participants were asked to perform difficult multiplication problems for either a short of long period, while other participants were given no multiplication to do. All participants were then asked to write a procedure to accomplish a related goal, and then to indicate what role (if any) the extra step in the original example had. When participants assign no role to this step they can be classified as superstitious or rational according to whether they retain the extra step in their generated procedure.

The results did not support the prediction. The proportion of superstitious responders was .17 (n=12) for no multiplication, .21 (n=14) for short multiplication, and .09 (n=11) for long multiplication. The differences in proportion of superstitious responders are not significant.

Dependence of generalization on domain. One of the questions raised by the EXPL work is the extent to which analysis and generalization are processes conditioned by knowledge or assumptions about a given domain, or should be seen as obeying principles largely independent of domain. For example, as discussed in Lewis (1988a), it could be that the identity heuristic is based on assumptions that are plausible for analyzing the behavior or artifacts, but that would not be accepted for natural systems. To address this question we presented isomorphic generalization problems in settings taken from computer operating procedures, a vaguely-specified industrial machine, a chemical reaction, and an animal breeding experiment. We were interested in possible differences, or lack of differences, among the settings, that might clarify the effect of domain.

The results obtained are confusing, and call for further investigation. For one of two generalization problems the computer setting was the only one in which participants produced the generalization expected by EXPL, while for a second the computer setting was the only one for which participants did *not* give the expected generalization. We suspect that these results may reflect item differences arising from the rewording of the problems to suit the various settings; a further study using a larger number of problems, with more than one rewording for each setting, might clarify this.

A related issue concerns the assumptions underlying generalization, and whether acceptance of these depends upon domain. As developed further in the discussion of theoretical work below, and in the report by Lewis, Hair, and Schoenberg (1989) which is included here, generalizations can only be justified by reference to some assumptions of regularity in the domain being analyzed. We asked participants to

choose between explanations of situations according to which various candidate assumptions were or were not violated, where different isomorphs of the situations were worded to place them in the four domains just mentioned: computers, machine, chemistry, breeding.

As with the generalization results just described, no clear pattern emerged. Some of the assumptions, such as that any outcome of a process must be controlled by some input, were treated differently in the artificial and natural domains: in this case the assumption appeared to be accepted for natural domains but rejected for artificial ones. A study in which the wording of situations is varied to dilute possible item effects, and in which more than one situation is used to test acceptance of a given assumption might help to clarify the picture. Protocol studies might also be useful in suggesting the basis for any differences that may emerge.

Theoretical Efforts.

Along with the development of the EXPL model, and the collection of empirical data bearing on it, the project has also tried to strengthen our theoretical grasp of analysis and generalization processes. Lewis (1988a) presents some of the results: defining a class of "analysis-based" generalization methods, including the so-called "explanation-based" methods, analogical generalization, and synthetic generalization, in which new procedures are produced by recombining elements of example procedures; and differentiating "superstitious" and "rationalistic" generalization processes.

More recent work has aimed to clarify the relationship between the kind of analysis of examples performed by EXPL and causal attribution. While earlier presentations of EXPL talked loosely about causal analysis, and commented on the apparent connections between EXPL's heuristics and heuristics seen in causal attribution, it proved unexpected difficult to pin down the relationship exactly.

One vexing issue served to bring this problem into focus, and drove our efforts to find a resolution. EXPL's "loose ends" heuristic says roughly that an unexplained cause can be linked to an unexplained effect. Mill's Method of Residues, a causal attribution heuristic, says that when all effects of some causes have been deducted from a situation, the remaining effects must be due to the remaining causes. Are these the same heuristic or not?

Attempts to settle this question revealed the inadequacy of our formulations of the analysis problem EXPL was trying to solve. In search of clarification we explored the philosophical literature, concluding, as discussed in Lewis, Hair and Schoenberg (1989), included with this report, that there is a serious mismatch between the philosophical notion of cause and the idea of causal connection assumed in the EXPL model, and needed to support the sort of generalizations it produces. Philsophical analysis treats events as *wholes*, and causal connections connect events. The relationships EXPL tries to discover and exploit instead link *aspects* of events. To avoid confusion, Lewis, Hair, and Schoenberg replace the term "cause" by

"control", where control relationships connect aspects of events rather than events as wholes.

The control framework clears up the relationship between loose-ends and Mill's method of residues: they are closely related, but different. Whenever both heuristics apply they give the same result, but they rest on different assumptions about regularities in the domain being analyzed, and hence have different applicability conditions.

Besides clarifying this specific question regarding EXPL's connection to causal attribution the control framework made it possible to reframe Mill's analysis of causation in terms of control. All of Mill's methods are recast as heuristics for identifying control relationships, and could be used compatibly with EXPL's heuristics whenever their applicability conditions are met.

A second area of theoretical work since Lewis (1988a) has been learnability analysis. Traditional inductive learning methods have a large literature analyzing formally classes of problems that can or cannot be solved within given performance constraints. But the recently-emerged analysis-based methods lack such an analysis. Thus we cannot characterize problems to which explanation-based generalization (for example) can or cannot be successfully applied, nor do we understand what issues determine this.

Lewis (1988b) attacks this problem for analogical generalization as performed by Anderson and Thompson's (1986) PUPS system. The paper shows that while for some simple forms of analogy there is a limited class of problems which have appropriate analogical structure, and to which analogical generalization can successfully be applied, for PUPS there is no such limited class: all problems can be solved using analogical generalization, given appropriate background knowledge. Thus (for example) no matter how seemingly inconsistent a computer command language appears, it can always be given an analysis under which it can be generalized completely using analogy.

This result is disappointing: it means that there is no way to distinguish analogical strtucture from unanalogical structure intrinsically: such structure resides not in the domain being analyzed but in the domain together with associated knowledge and interpretation. Thus to design a command language that can be generalized by analogy one cannot rely on any simple structural criterion for guidance, but instead must worry about what users will know about the language, or what they can learn about it. On the face of it this seems a much harder problem than characterizing structural regularities.

Subsequent work has shown that this analysis can readily be extended to other methods of analogical generalization. For example in structure mapping (Gentner 1983) "analogicalness" depends not on any structural property of a domain but rather on the relationships attached to it. It remains an open question whether similar results obtain for other analysis-based methods.

Summary of Main Results and Open Areas.

Thus far the EXPL project has succeeded in clarifying the role of understanding in learning, by demonstrating how analysis of examples supports generalization, which is an essential element in non-trivial learning in the procedural domain. Exploration of the relationship between the EXPL model and other generalization techniques led to recognition of the class of analysis-based methods. Exploration of the relationship between EXPL's analysis methods and causal attribution led to development of a rigorous framework within which methods of causal analysis can be defined and compared. EXPL's heuristics are seen to be new, though closely related to already-noted heuristics. Some progress has been made towards understanding the limits, or lack thereof, on what can be learned by analysis-based methods.

Many important areas remain to be better understood. The basis for the identity heuristic, the most robust of EXPL's heuristics, remains unclear. Is it based on conventions of communication, or is it a reflection of a widespread regularity in the world? Is identity itself the relevant cue, or is an identity simply a variety of coincidence, any of which would trigger analysis? This is related to the question of the domain dependence of generalization methods, discussed above as needing further study.

Despite some efforts, the role of analysis-based methods in real learning remains in doubt. Studies that compare learning with and without examples, as suggested above, may shed light on this.

Learnability analysis for analysis-based methods is needed. The results obtained for analogical reasoning need to be explored for other methods, and the issue of limitations on the analysis process, as well as the generalization process, need to be considered. This involves getting insight into the relationships between background knowledge and analysis, and background knowledge and generalization, hardly attacked in this project.

Finally, current work in human-computer interaction is building on Kintsch's construction-integration model (Kintsch 1988, Mannes and Kintsch 1988), which uses largely associative processes rather than the symbolic rule processes seen in EXPL. There are interesting prospects of integrating EXPL's learning approach into this associative framework, but the means of doing this are unclear as yet. It is possible that Kintsch's associative model will permit a successful attack on one of the original goals of the EXPL project: to model explanations as satisifying a constellation of constraints, rather than as the result of discrete, orchestrated heuristics.

References.

Anderson, J.R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard.

Anderson, J. R. and Thompson, R. (1986). Use of analogy in a production system architecture. Paper presented at the Illinois Workshop on Similarity and Analogy, Champaign-Urbana, June, 1986.

Casner, S. and Lewis, C. (1987) Learning about hidden events in system interactions. In *Proceedings of CHI'87 Conference on Human Factors in Computer Systems*. New York: ACM, pp. 197-203.

Gentner, D. (1983). Structure mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170.

Kintsch, W. (1988) The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95, 163-182.

Lewis, C.H. (1986a) A model of mental model construction. In *Proceedings of CHI'86 Conference on Human Factors in Computer Systems*. New York: ACM, pp. 306-313.

Lewis, C.H. (1986b) Understanding what's happening in system interactions. In D.A. Norman and S.W. Draper (Eds.) *User Centered System Design: New Perspectives on Human-Computer Interaction* . Hillsdale, NJ: Erlbaum.

Lewis, C.H. (1988a) Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science*, 12, pp. 211-256.

Lewis, C.H. (1988b) Some learnability results for analogical generalization. Technical Report CU-CS-384-88, Department of Computer Science, University of Colorado, Boulder.

Lewis, C., Casner, S., Schoenberg, V., and Blake, M. (1987) Analysis-based learning in human-computer interaction. In *Proceedings of INTERACT'87*, Elsevier Science Publishers.

Mack, R.L. (1984) Understanding and learning text editing skills: Evidence from predictions and descriptions given by naive people. Research Report RC103330, IBM, Yorktown Heights, NY.

Mannes, S.M. and Kintsch, W. (1988) Action planning: Routine computing tasks. In *Proc. 10th Annual Meeting of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum, 97-103.

Polson, P.G. and Kieras, D.E. (1985) A quantitative model of the learning and performance of text editing knowledge. *Proceedings of CHI'86 Conference on Human Factors in Computing Systems*. New York: ACM.

Learning from a Demonstration: Encoding, Analysis, and Generalization

Victor Schoenberg
Clayton Lewis
Mitchell Blake

Department of Computer Science and
Institute of Cognitive Science
Campus Box 430
University of Colorado
Boulder CO 80309
(303) 492 6657

9.7.86

Abstract. Computer operations are often learned from demonstrations. Effective learning requires not just seeing and remembering what was done but being able to modify what was done to meet the requirements of new tasks. The EXPL model (Lewis, 1986a) provides an account of how causal analysis could be used to support this kind of generalization. Lewis (in preparation) presented results from artificial paper-and-pencil tasks that suggested that learners used a superstitious generalization process in which features of an example that were not understood were retained in new procedures based on the example. In this paper we investigate learning from a more realistic video demonstration. The results suggest that a *different, less superstitious generalization process* is applied to the results of the analysis than was seen in the artificial tasks. The results indicate some limitations of demonstrations as learning aids.

Acknowledgements. This work was supported by grants from the Office of Naval Research (Contract No. N00014-85-K-0452), the Institute of Cognitive Science, and AT&T.

Introduction.

Learning from a demonstration requires an analysis of what the parts of the demonstration do, so that one can make appropriate changes to what was shown in tackling one's own tasks. For example, if we are shown how to delete a file we must try to determine how the identity of the file was specified in the demonstration, since almost certainly the file we want to delete is not the one deleted in the demonstration. The EXPL model (Lewis, 1986a) gives an account of how learners could use a small set of heuristics to establish causal links between user actions and system responses in a demonstration, and how these links could be used by a generalization process to accomplish novel tasks. Lewis (in preparation) obtained data using artificial, paper-and-pencil examples that were

participants' recall of the steps of the demonstration, and then asked them to perform a task with the actual system that was similar to what they had seen. We hoped the results would clarify the mode of generalization learners might employ, and would indicate the extent to which the EXPL framework could usefully be applied to more realistic learning situations. We also hoped to shed some light on the strengths and weaknesses of demonstrations as learning aids. Demonstrations may avoid some of the pitfalls of learning by instruction or learning by exploration, and may offer economic advantages over more sophisticated tutorial techniques.

Method.

Participants. Five university students, with a wide range of computer experience, served in the experiment. One participant, a graduate student in Computer Science, had used a Macintosh computer, the machine used in the study. None of the other participants had used the Macintosh and no participant had seen or used the STELLA(TM) program.

Materials. A four-minute videotape was prepared which showed the use of STELLA(TM), a system for graphically building systems dynamics models, to construct the diagram shown in Figure 1a. The task required a series of actions with the mouse and mouse button, and some keying, to construct, place, and label a collection of graphical objects. The tape showed the Macintosh display in the upper two-thirds of the video screen, with a view from above of the mouse and the operator's fingers in the lower third of the screen. A microphone attached to the mouse picked up the sounds made by the mouse as it rolled and by the mouse button as it was pressed and released. No other sound track was included, and no description of the STELLA(TM) program or its purpose was provided to participants.

Two versions of the tape were used. Because the first two participants had trouble determining when the mouse button was pressed and released, versus when it was pressed and held, a revised version of the tape with exaggerated finger movements was prepared and used with the last three participants.

Procedure. Participants watched the videotape once straight through. They were not allowed to stop the tape or review any portion of it. When they had viewed the tape they were asked a series of questions about the content of the demonstration, including the order of operations and the specific steps required for particular operations. Their responses were recorded on audiotape and transcribed for later analysis. Following this questioning participants were given the diagram shown in Figure 1b and

pointer within the object from which the link was to run, even though the demonstration showed the head only (in one case) or the head and tail (in another case) within the to-be-linked objects. The appearance of the directed links was such that the origin of a link resembled the tail of the pointer, while the head of a link resembled the head of the pointer. It is possible that this similarity suggested the use of the tail rather than the head to establish the origin, and that participants either failed initially to encode the details of the interaction that contradicted this suggestion, or failed to recall these details during performance. In either case this appears to be an instance in which a plausible but false expectation about the interface dominated participants' actual observation.

One participant was distracted for a moment during the demonstration, and did not observe how on of the objects was produced. Clearly learning-by-demonstration is vulnerable to this kind of interference, probably more than alternative approaches such as guided exploration.

Participants generally seemed to recall the function of icons without difficulty. But two participants had trouble with the "dynamite" icon used to delete objects. Since "dynamite" was used only to correct an apparent error in the demonstration, it is tempting to speculate that participants may have failed to encode this portion of the demonstration, just as a person trying to transcribe a conversation verbatim may have trouble including speech errors and corrections in the transcript.

Analysis. STELLA(TM) makes heavy use of intermediate feedback and identity cues (Lewis 1986a,b), in which user actions and system responses are linked by the occurence of identical or similar objects or attributes. So we expected that assigning causal connections within the demonstration would not be difficult. To test this we prepared an EXPL encoding of the video demonstration, and extended the EXPL analysis system to handle relationships that had not arisen in the simpler examples discussed in Lewis (1986a). The extensions included the ability to establish an identity connection between an object and its location, and the ability to represent hierarchical groups of items.

The encoding of the demonstration comprised 74 events, including 31 user actions, which formed a complex subgoal structure. EXPL's causal attribution heuristics had trouble with this subgoal structure, often linking system responses to much earlier and unrelated user actions. We attacked this problem by adding features to EXPL to permit it to subdivide a long example into episodes, following the work of Newtson and colleagues on event perception (Newtson, Engquist, and Bois, 1977;

eight major operations in the demonstration, changing only the names assigned to the parts, before needing to deviate from the sequence of operations shown in the demonstration. Superstitious generalization would dictate this course, since the order of operations constitutes an attribute of an example not to be tampered with without some reason. Participants in Lewis' (in preparation) paper-and-pencil study were in general unwilling to reorder steps in an example, even when they gave no reason order was important. By contrast, mechanistic generalization will freely reorder operations unless there is some logical dependency between them.

Four out of five participants placed themselves in the mechanistic camp, by reordering the operations in the demonstration. This does not appear to be an encoding or recall failure, since all participants recalled the order of operations in the demonstration correctly before the test task. The reorderings differed among the participants. Two participants worked left-to-right and top-to-bottom. One seemed to be deferring more difficult operations by working on easier ones first. Another placed the major shapes in the same order as in the demonstration, but varied the order in which names and links were added.

Interestingly, the STELLA(TM) interface includes some prerequisite relationships between operations that are not apparent unless they are violated. By using mechanistic generalization, and varying the order of operations, three of the five participants got into trouble. For example, the conspicuous flow object, visible in Figures 1a and 1b as a cloud with an arrow emerging from it, cannot be placed in a diagram unless there is already a stock, the rectangular object at the end of the arrow, present for it to be attached to. In the demonstration the stock was placed first, so there was no difficulty in placing the flow. When participants tried to place the flow first they found that the flow would disappear when they tried to place it.

A second hidden order constraint concerns assigning a name to an object. When an object is newly created it is in a selected state in which its name may simply be typed in. Once other operations have been performed the object is no longer selected, and a special operation must be carried out to select it again so that its name can be typed. In the demonstration, objects were always named right after creation, so the special selection operation was never needed and never shown. When participants changed the order of operations and tried to assign names later they were therefore in trouble. In this case, as with placing the flow object, superstitious generalization would have been more successful than mechanistic.

References.

Anderson, J.R. and Thompson, R. (1986) Use of analogy in a production system architecture. Paper presented atthe Illinois Workshop on Similarity and Analogy, Champaign-Urbana, June, 1986.

DeJong, G. and Mooney, R. (1986) Explanation-based learning: An alternative view. *Machine Learning*, 1.

Ericsson, K.A. and Simon, H.A. (1984) *Protocol Analysis: Verbal Reports as Data* Cambridge MA: MIT Press.

Gentner, D. (1983) Structure mapping: A theoretical framework for analogy. *Cognitive Science*, 7, pp. 155-170.

Kedar-Cabelli, S. (1985) Purpose-directed analogy. In *Proceedings of the Cognitive Science Society Conference*, Irvine, CA.

Lewis, C.H. (1986a) A model of mental model construction. In *Proceedings of CHI'86 Conference on Human Factors in Computer Systems* New York: ACM, pp. 306-313.

Lewis, C.H. (1986b) Understanding what's happening in system interactions. In D.A. Norman and S.W. Draper (Eds.) *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Erlbaum.

Lewis, C.H. (in preparation) Why and how to learn why: Analysis-based generalization of procedures.

Mack, R.L. (1984) Understanding and learning text editing skills: Evidence from predictions and descriptions given by naive people. Research Report RC103330, IBM, Yorktown Heights, NY.

Mitchell, T.M., Keller, R.M. and Kedar-Cabelli, S.T. Explanation-based generalization: A unifying view. *Machine Learning*, 1.

Newtson, D., Engquist, G. and Bois, J. (1977) The objective basis of behavior units. *Journal of Personality and Social Psychology*, 35, pp. 857-862.

Newtson, D., Rindner, R., Miller, R., and LaCross, K. (1978) Effects of availability of feature changes on behavior segmentation. *Journal of*

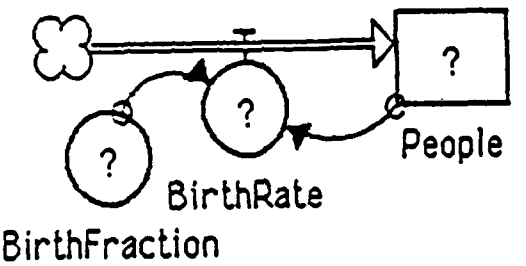


Figure 1a. The diagram whose construction was shown in the video demonstration.

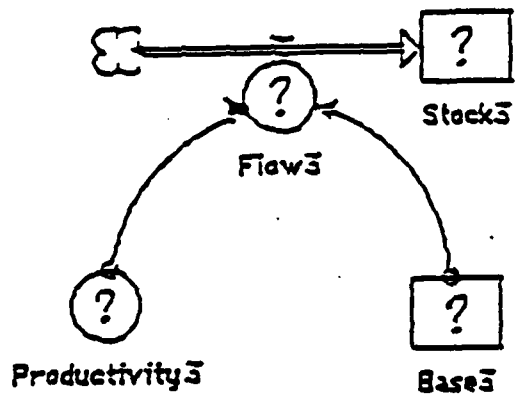


Figure 1b. The diagram participants were asked to construct during test.

Learning from a Demonstration

Clayton Lewis
D. Charles Hair
Victor Schoenberg

Department of Computer Science and
Institute of Cognitive Science
Campus Box 430
University of Colorado
Boulder CO 80309

Address correspondence to the first author at the above address

(303) 492 6657

Abstract

Learning to use a computer system is difficult, and existing training methods are costly and often ineffective. Demonstrations might provide an inexpensive and effective way to solve this problem, but little is known about the effectiveness of demonstrations as training tools, or about what features of a system do and do not lend themselves to presentation in demonstration form.

In this paper we report the results of a study of learning to use a spreadsheet from a 15 minute video demonstration. Learners who viewed the video performed less well than other learners who worked through a hands-on tutorial manual showing the same operations. Results from other participants who were asked simply to report what they saw when viewing the video, and a theoretical analysis using the EXPL causal attribution model, suggest that difficulties in learning from the demonstration center on noticing critical events in the demonstration.

Word count: 3003

Keywords: learning, demonstrations, tutorials

Topic: Psychological models of user learning and performance

Learning from a Demonstration

Clayton Lewis
D. Charles Hair
Victor Schoenberg

Department of Computer Science and
Institute of Cognitive Science
Campus Box 430
University of Colorado
Boulder CO 80309

Abstract
Learning to use a computer system is difficult, and existing training methods are costly and often ineffective. Demonstrations might provide an inexpensive and effective way to solve this problem, but little is known about the effectiveness of demonstrations as training tools, or about what features of a system do and do not lend themselves to presentation in demonstration form.

In this paper we report the results of a study of learning to use a spreadsheet from a 15 minute video demonstration. Learners who viewed the video performed less well than other learners who worked through a hands-on tutorial manual showing the same operations. Results from other participants who were asked simply to report what they saw when viewing the video, and a theoretical analysis using the EXPL causal attribution model, suggest that difficulties in learning from the demonstration center on noticing critical events in the demonstration.

Acknowledgements.

This research was supported by the Office of Naval Research, Project NR702-009, with additional contributions of funds or equipment from Hewlett Packard Corporation, AT&T, and the Institute of Cognitive Science. The authors wish to thank Mitchell Blake for assistance in programming.

Introduction.

Learning is a critical issue in human-computer interaction, and a good deal of research has been directed toward the design of training manuals (for example Black, Carroll and McGuigan, 1987) and learning by exploration (for example Carroll, Mack, Lewis, Grischkowsky, and Robertson, 1985). But little attention has been paid to learning from demonstrations, despite the possible economic advantages of demonstrations, as compared with carefully prepared written material, and despite the fact that users often express a preference for being shown how to do things by a co-worker rather than using manuals.

To explore the effectiveness of demonstrations as a learning tool, we prepared a

video recording, about 15 minutes in length, of a demonstration of the EXCEL spreadsheet program (Microsoft, 1985). We asked experimental participants to view this video, and then to perform a similar task without referring to the video or any other materials. Other participants did not view the video, but instead worked through a hardcopy manual which presented a hands-on tutorial covering the same material, and then undertook the same test task. Comparing the performance of the two groups of participants gives an indication of the effectiveness of a demonstration in presenting this materal.

Since the effectiveness of a demonstration depends in part on participants noticing the key events in the demonstration, we asked a third group of participants to view the video demonstration and simply report what they saw. By comparing the completeness of these resports with the difficulty of performing corresponding subtasks during the test, we can attempt to determine which difficulties can be attributed to problems in noticing what is happening in the demonstration.

We also analyzed the demonstration using the EXPL analysis program (Lewis 1986; Lewis, Casner, Schoenberg, and Blake, 1987). This program uses causal attribution heuristics to determine causal connections between user actions and system responses in a demonstration. These connections can be used to generalize the operations seen in the demonstration to accomplish novel tasks. This causal analysis is one kind of understanding of the demonstration. If features of the demonstration that EXPL cannot analyze are the same features that give learners trouble, this would suggest that problems in learning from the demonstration are associated with problems in understanding it, and that the EXPL model gives a good account of these problems.

Method: Comparing video demonstration and tutorial manual.

Participants. Participants were fourteen students in an undergraduate psychology course who earned course credit for their participation. Participants were not screened for computer experience; experience estimated by the participants ranged from 2 to 500 hours, with a median of 30 hours. No participant had experience with the EXCEL program used in the study, but four had experience with the Macintosh computer and two had experience with other spreadsheet programs.

Materials. A portion of the tutorial section of the EXCEL manual (Microsoft, 1985), covering basic spreadsheet operations, was selected for use in the study. This tutorial presented step-by-step instructions for carrying out a task scenario, including starting the program, entering data and formulae, manipulating format, and saving results. A video demonstration, showing all of the operations in this scenario, was prepared. The soundtrack of the video demonstration included brief explanations of some user actions that would be difficult or impossible to detect from a visual presentation alone, such as the mouse button actions required to make a menu selection. The soundtrack included no information about the purpose or effect of any of the operations shown. Participants were told that the software was intended to support

operations of the kind accountants perform. The video demonstration was about 15 minutes in length.

A test task, which could be accomplished using variations of the actions included in the task scenario, was prepared. Participants were directed to use the EXCEL program to build a spreadsheet, to include headings and data provided to them in rough tabular form, to enter some simple formulae, to copy a section of the spreadsheet, and to save their work.

Procedure. Participants were assigned to the Video or Manual groups in alternation. Participants in the Video group were shown the video demonstration, and then given up to 45 minutes to work on the test task. Participants were not permitted to replay any portion of the video demonstration, either during their initial viewing, or during the test task. Participants were asked to think aloud during the entire procedure, including the viewing of the demonstration and their work on the test task. An audio recording was made of the entire session, and a video record was made of their work on the test task.

Participants in the Manual group were given up to 30 minutes to work through the selected excerpt from the EXCEL manual using the Macintosh. They then were given the balance of an hour to attempt the same test task as the participants in the Video group. As with the Video group, the manual group was asked to think aloud, and similar recordings were made of their performance, except that a video record was made of the entire session for these participants.

Participants in both groups were free to ask the experimenter for assistance. Assistance was provided when, in the experimenter's judgement, participants would be unable to deal with their difficulty in a reasonable amount of time.

Method: Reporting events in the video demonstration.

Participants. Participants were ten students in a summer session of an undergraduate psychology course who earned course credit for their participation. Participants were not screened for computer experience; most participants indicated they had 'some' computer experience. No participant had experience with any spreadsheet program, and three had experience with the Macintosh computer.

Materials. The same video demonstration used in the comparison with the written tutorial was used.

Procedure. Participants were asked to view the video demonstration, and to report aloud the events that they saw or heard in the video. They were permitted to stop the tape but not to review any portion of it. An audio recording was made of their reports.

Analysis and results.

Demonstration breakdown. For purposes of analysis the scenario shown in the demonstration was broken down into a series of significant events, as judged by the experimenters. These events, in turn, were grouped into 26 operations, each including a meaningful user action, or series of closely-related actions, and associated system responses. Examples include selecting a row, entering a single data item, and selecting a specific item from a menu.

Test task breakdown. Since participants were free to choose their own methods and order of operations for the test task, steps in the test task do not correspond in a simple manner to steps shown in the demonstration. For purposes of analysis the test task was broken down into seven subtasks, each in general including more than one of the simpler operations used in analyzing the demonstration. These subtasks are described in Table 1.

Table 1

Subtask	Description
Start	turning on the machine, inserting diskettes, starting program
Enter data	entering numbers and headings in spreadsheet
Widen	widen a column as necessary to accomodate a long heading
Format	arrange for numbers to appear as dollar amounts
Formula	enter specified formula in spreadsheet
Copy	duplicate the amounts in one column in another column
Save	store the spreadsheet on diskette with a specified name

Success rate. Table 2 shows the number of participants judged to have successfully accomplished each subtask, in the Video and Manual groups. Since not all participants attempted each subtask, either because of lack of time, or because they found an alternate way to accomplish the required result, the number of participants who attempted each task, and the proportion of those who attempted it who were successful, are also shown.

Table 2.

	Video Group			Manual Group		
Subtask	Number of Successful Ps	Number of Ps Attempting	Success Rate	Number of Successful Ps	Number of Ps Attempting	Success Rate
Start	7	7	1.00	7	7	1.00
Enter data	7	7	1.00	7	7	1.00
Widen	3	7	0.43	6	7	0.86
Format	1	5	0.20	7	7	1.00
Formula	1	7	0.14	7	7	1.00
Copy	2	5	0.40	2	5	0.40
Save	5	5	1.00	4	4	1.00

These results indicate that the manual presentation was somewhat more effective in enabling participants to accomplish some of the critical operations in the test task. Fewer participants in the Manual group attempted the last subtask, but since different amounts of time were allotted for the test phase for the two groups, and since participants were helped when they bogged down, not much weight should be given to this difference. Two participants in the Manual group did not finish working through the manual in the allotted 30 minutes, and they did not even attempt to perform those subtasks of the test task that they had not seen in the manual.

Reproduction of demonstrated operations. In addition to investigating how successful participants were in carrying out the various subtasks of the test task, we can ask how well participants were able to reproduce the actions they were shown in the video or in the manual. This is a different measure from success, since participants might be able to reproduce part of a procedure but not enough to complete a subtask; equally, participants might complete a subtask successfully, but not by using a method shown in the video or manual. For each of the 26 operations in the breakdown of the demonstration described above, we counted the number of participants in each group whose test task performance included any instance of this operation, regardless of whether this reproduction formed part of a successful subtask or not. We divided these counts of participants by the number of participants who might reasonably have used the operation in question. For example, participants who did not reach the stage of saving their work were not counted in assessing the rate of reproduction of the operations required for saving work. The resulting reproduction score ranges from 0, indicating that no participant carried out the given operation during the test phase, to 1, indicating that all participants who attempted a pertinent part of the test task carried out the given operation at least once. The median score for the Video group is .76, and for the Manual group, 1.00. The difference in reproduction is significant by a sign test: of 12 operations whose reproduction differed, 10 had higher scores for the Manual group than for the Video group, $p<.05$.

The correlation of reproduction scores for the two groups is very high, and significant: rank sum correlation = .826, $p<.01$. This indicates that while overall reproduction is higher for the Manual group, the relative reproduction score for different operations is not affected much by presentation mode.

Reporting rate for operations. Aspects of a demonstration can differ in how salient the critical events are, whether user actions of system responses. To get an idea of how the operations shown in the video demonstration compare in these respects, we scored the verbal reports produced by the participants who viewed the video and reported what happened by tallying the events we judged significant in each operation that were or were not clearly mentioned in a participant's report. To obtain a summary measure of reporting for each operation, taking into account the fact that different operations involved different numbers of events, we calculated for

each participant the proportion of significant events mentioned, and then took the median of this proportion across the 10 participants. The resulting reporting rate score ranges from 0, indicating that the median participant mentioned none of the significant events in an operation, to 1, indicating that the median participant mentioned all of the significant events. The median reporting rate for the 26 operations is .45.

Relation between reporting rate and reproduction. The rank correlation of reporting rate and reproduction by the Video group is .654, which is significant at the .01 level. This indicates that operations that were poorly reported were poorly reproduced by other participants, consistent with the idea that problems in noticing the critical events in the demonstration were a source of trouble in reproducing the operations shown. It is possible that other factors than noticing, such as the comprehensibility of events, also affect reporting rate. We return to this point in the discussion.

EXPL analysis. We manually encoded the events in the demonstration and submitted them to the EXPL analysis program, which attempts to place causal links between user actions and resulting system responses. We judged that this analysis was accurate for 11 of the 26 operations, and questionable or clearly deficient for the remaining 15. If the EXPL model gives a good account of the comprehensibility of parts of a demonstration, and if problems in comprehensibility caused participants trouble in reproducing what they saw, the reproduction scores should be higher for operations EXPL can analyze than for operations it cannot analyze. In fact, the median reproduction scores are 1.00 and .57 for these classes of operation, but this difference is far from significant (rank-sum test, $z=1.14$).

Comprehensibility could affect reporting rates if participants were less (or more) likely to report events whose significance they could not determine. If EXPL provides a good account of comprehensibility, reporting rates would then differ between operations EXPL can analyze and ones it cannot. Median reporting rates for these classes of operations are .33 and .39 respectively; this difference does not approach significance (rank-sum test, $z=.104$).

Discussion.

By itself, the video demonstration did not perform as well as the tutorial manual on which it was based. Participants who used the manual were more successful in accomplishing some key tasks, and more often reproduced operations shown in the tutorial. We can advance some reasons for the poorer performance of the video demonstration, based on the analyses presented above, and on the details of what participants did in attempting the test task.

The pattern of results obtained points to noticing as a likely problem in learning from the video. The EXPL analysis, which at least attempts to characterize the comprehensibility of the demonstration, does not account for the pattern of difficulty participants encountered. On the other hand, there is a strong relationship between how completely some participants reported the key events in different parts

of the video and how often other participants reproduced those same parts. Thus seeing what is happening appears to be a more powerful factor, at least in this demonstration, than understanding what is happening.

One subtask of the test that illustrates this point is widening a column. As Table 2 shows, Manual subjects were more often able to perform this subtask than Video participants. A critical feature of the correct widening procedure is placing the cursor on a specific part of the boundary of the column to be widened. Of the four Video participants who had trouble with this task, only one placed the cursor at the right spot. This is consistent with the idea that participants had not noticed exactly where the cursor must be placed. The presentation of this procedure in the manual not only describes where the cursor must be placed in text, but uses an arrow to indicate the key spot in an accompanying diagram. It is plausible that these devices made it more likely that Manual participants noted this key aspect of the procedure.

Another subtask on which the Manual participants did better points to another difference between the demonstration and the manual, at least as they were employed in this study. Manual participants could, and did, look back at the manual while working or. the test task, while Video participants could not review the video. This may have been an important difference in the 'formula' subtask: four of the Manual participants did refer back to manual during this subtask.

Examination of participants' attempts at the Format subtask, the remaining subtask for which the Video group did poorly as compared with the Manual group, suggests that in this case the comprehension processes addressed by the EXPL model may play a role. This subtask requires two main steps, selecting a region of the spreadsheet to which a new format will be assigned, and selecting a new format. The EXPL analysis of the demonstration revealed that there is no cue in the demonstration to indicate that these two steps, though performed consecutively, are linked. Of the four Video participants who attempted the Format subtask and failed, all four selected a new format but neglected to select a region to which it would apply, exactly as would be expected from the EXPL analysis. The presentation of the Format subtask in the manual clearly groups the region selection with the format selection, so the role of the region selection is much clearer.

Care is needed in interpreting these findings broadly, in view of the specific conditions under which we compared the video and manual presentations. Our video was an almost purely visual presentation, with very limited information added in the soundtrack. Participants were not permitted to review it, while they were permitted to review the manual, since that is part of normal usage of manuals. The Manual group did hands-on work during learning, while the Video group simply watched the demonstration.

Our findings point to specific steps that might be expected to improve the performance of the video, steps that would change some of these conditions. First, the soundtrack could be used to call attention to key aspects of the operations shown, such as the critical placement of the cursor in the Widen subtask, so that they would

more often be noticed. Second, means could be provided to permit learners to review portions of a demonstration if desired. This presents new challenges in design: There are standard, familiar ways of structuring and indexing text, and learners have a lot of experience in looking back for information in a book. How does one provide like facility for a video presentation? Finally, missing connections, such as that pointed up by the EXPL analysis of the Format subtask, could be indicated in the soundtrack of a demonstration.

References.

Black, JB, Carroll, JM, and McGuigan, SM. (1987) What kind of minimal instruction manual is the most effective? In proceedings of CHI+GI 1987 (Toronto, April 5-9). ACM, New York, pp. 159-162.

Carroll, JM, Mack, RL, Lewis, CH, Grischkowsky, NL, and Robertson, SP. (1985) Exploring exploring a word processor. *Human-Computer Interaction*, **1**, pp. 283-307.

Lewis, CH. (1986) A model of mental model construction. In proceedings of CHI 1986 (Boston, April 13-17). ACM, New York, pp. 306-313.

Lewis, CH, Casner, S, Schoenberg, V, and Blake, M. (1987) Analysis-based learning in human-computer interaction. In proceedings of Interact 87, 2d IFIP Conference on Human-Computer Interaction (Stuttgart, September 1-4).

Microsoft Corporation. (1985) *Microsoft Excel User's Guide.*

Generalization, Consistency, and Control*

Clayton Lewis, D. Charles Hair, and Victor Schoenberg
Department of Computer Science and
Institute of Cognitive Science
Campus Box 430
University of Colorado
Boulder CO 80309
clayton@sigi.colorado.edu (303) 492 6657

Address correspondence to the first author.

September 26, 1988

Abstract. Easy learning of a user interface depends in part on users being able to generalize successfully about it. Philosophical doctrine, and some recent work in human-computer interaction, argues that *causal analysis* of interactions can support generalization. But neither the philosophical literature nor the HCI literature provides a rigorous theory of causal analysis adequate for problems in human-computer interaction. We propose ~~such~~ a rigorous theory ~~here~~, and show how it accounts for two robust generalizations, using certain general assumptions. We then present evidence that these assumptions are accepted by people. Finally we compare this theory with other treatments of consistency.

Consider the following three commands from a fictitious system:

(E1) foo baz: deletes the authorization table
(E2) blee baz: deletes the terminal assignment table
(E3) foo bar: prints the authorization table

What command would you issue to print the terminal assignment table? Probably you will say "blee bar" (of a sample of eleven computer scientists ten gave this response.) But why?

An answer in general terms is easy to sketch. Comparison of the examples suggests that "baz"causes deleting and "bar" causes printing in the examples, while "foo" specifies the authorization table and "blee" the terminal assignment table. Mill's Methods of Induction (Mill, 1900) could be employed to organize this analysis. The causal connections obtained can be used to support generalizations to novel cases: if I conclude that "bar" *causes* printing, as opposed to being associated with it perhaps coincidentally, then using "bar" in the future should produce printing. Generalization of interactions using causal analysis in this way is discussed in Lewis (1986), Lewis, Casner, Schoenberg, and Blake (1987), and Lewis (1988).

*Revised version to appear in "Proc. CHI'89 Human Factors in Computing Systems", New York:ACM.

On examination this sketch has problems. Do we really want to say that "bar" or typing "bar" *causes* printing? Typing "bar" by itself certainly does not. Typing "bar" preceded by an invalid name does not cause printing either. We could perhaps try to say that typing "bar" causes printing when the system is in certain states, but such close reasoning seems absent from our original confident generalization.

Consultation of the philosophical literature (for example Mackie 1974) confirms the impression that our concepts of causation, as analyzed by philosophers, are not well adapted to reasoning about the sort of problem that is typical in human-computer interaction. Philosophical discussion concentrates on causal connections among events considered as wholes, while we need to know how the *constituent parts* of events determine the constituent parts of later events. The troublesome "bar" and "printing" are examples of these constituents whose role we need to understand in generalizing.

To avoid confusion we suggest the terms "control" and "specify" to replace "cause" in describing the connections among these constituents. Thus we will say that in the three examples above "second word of command" *controls* "operation performed", and that the value "bar" *specifies* "printing". Here is how we define these terms.

The domain from which examples are drawn and within which we wish to generalize consists of a collection of *cases*. In the present discussion a case will represent an example command together with some description of its outcome. The terminology is kept abstract so as to permit applications of this framework to domains other than commands and their outcomes, such as direct manipulation interactions, sentences and their meanings, and others. Though space does not permit a discussion here, Mill's Methods of Induction can be derived within this new framework by considering cases consisting of collections of antecedent and consequent circumstances.

A case has a collection of *aspects*, which are simply functions from cases to other domains. For example, "second word of command" would be an aspect which maps cases into words, while "operation performed" maps cases into operations. Aspects need not always be defined, as might happen when we attempt to print a nonexistent file. Aspects are divided into *antecedent* and *consequent* aspects: antecedent aspects in our discussion will be aspects of commands, while consequent aspects will be aspects of their outcomes. So "second word of command" would be an antecedent aspect and "operation performed" a consequent aspect. We assume that the antecedent aspects of cases are sufficient to determine the consequent aspects, so that two cases with the same values for all antecedent aspects must have the same values for all consequent aspects (situations in which identical commands give different results would be dealt with by including system state information among the antecedent aspects.)

Control is a relationship between antecedent and consequent aspects. Intuitively, saying that "second word of command" *controls* "operation performed" means that by changing the second word of a command we can change the operation it performs. Rigorously, the definition is this, using bold lower case letters for aspects,

capitals for cases, and the notation e(C) for the value of aspect e at case C:

Aspect a *controls* aspect c *at case* C iff

(1) a is an antecedent aspect of C and c is a consequent aspect, and

(2) both a and c are defined for C, and

(3) there is a contrasting case C' for which
(3a) a(C') is defined and not equal to a(C), and
(3b) e(C')=e(C) for all other antecedent aspects e, and
(3c) c(C') is defined and not equal to c(C).

The requirements that aspects be defined are intended to exclude situations in which an invalid value for an antecedent aspect might affect a consequent aspect that is really independent. For example, changing a file name from a valid to an invalid value may make the operation performed by a command undefined, but one would not wish to say that the file name controls the operation.

Using this definition we can say something about the examples E1-3 discussed above. Let us use "first word of command" and "second word of command" as antecedent aspects, and "thing operated on" and "operation performed" as consequent aspects. Applying the definition of control to case E1, using E2 as the contrasting case, we see that "first word of command" controls "thing operated on"at E1 (and similarly at E2.) Examining E1 using E3 as the contrasting case reveals that "second word of command" controls "operation performed" at E1, and similarly at E3.

This is progress, but we are unable to base any generalizations on these control relationships. There are two reasons for this. First, control relationships are defined purely locally: the fact that "first word of command" controls "thing operated on" at E1 and E2 does not permit us to assert that "first word of command" controls "thing operated on" at E3, as we would wish to.

A second gap in our reasoning so far is that knowing control relationships among aspects does not tell us how the *values* of those aspects will behave. Even if we know that "first word of command" controls "thing operated on" at E3 we do not know that replacing "foo" by "blee" in E3 will produce a case whose "thing operated on" is "terminal assignment table".

Strong, but plausible, assumptions are needed to fill these gaps. First we will assume that the control relationships we find at one case will hold at others. Formally, we assume

A1 *Consistency of control*: If a controls c at some case C, and a and c are defined at another case C', then a controls c at C'.

To deal with the behavior of values we will assume that whenever one aspect controls another, the values of the aspects are associated in the same way. Formally, we assume

A2 *Consistency of consequent values*: If a controls c at cases C and C', and a(C')=a(C), then c(C')=c(C).

This definition supports the usage of saying that a value of an antecedent aspect *specifies* the associated value of a consequent aspect when one aspect controls the other. Thus in E1 we can say that "foo" *specifies* "authorization table" and "baz" *specifies* "deletion", since "first word of command" controls "thing operated on" and "second word of command" controls "operation performed" at E1.

Armed with these powerful assumptions we can almost derive the generalization we want from E1-3. But there is still a gap: we need to know that the consequent aspects "thing operated on" and "operation performed" are *defined* for the command "blee bar". The rest of what we want would then follow: "first word of command" would control "thing operated on" at "blee bar", and the value "blee" would then specify the consequent value "terminal assignment table"; similarly "second word of command" would control "operation performed", and "bar" would specify "print". But none of our assumptions ensure that these consequent aspects are defined for "blee bar", and if they are not all bets are off: we cannot then even be sure that "first word of command" controls "thing operated on" at "blee bar".

This difficulty is a real one, and not just a glitch in the formal machinery. In a real system it could easily happen that "blee bar" is an invalid command, perhaps because the terminal assignment table cannot be printed. Just because something can be deleted, as in E2, does not ensure that it can be printed, or that it can be printed in the same way as other objects. Yet it is plausible to assume that such irregularities do not occur, as we implicitly do in arriving at "blee bar" as the command we need. We apparently assume that values which are valid for an aspect in one situation will be valid in others, or equivalently, that an invalid value in one context is invalid in another. Formally, we can define an invalid value as follows.

v is an *invalid value* for a and c at C iff

a is an antecedent aspect and c a consequent aspect of C, and
a controls c at C, and
there is a case C' such that a(C')=v, and
e(C')=e(C) for all other antecedent aspects, and
c(C') is undefined.

Now we can state the assumption

A3 *Consistency of invalid values*: if v is an invalid value for a and c at some case C, and a(C')=v for any case C', then c(C') is undefined.

We now have enough assumptions to deal with E1-3. If "thing operated on" were undefined for "blee bar", then "blee" would be an invalid value for "first word of command" and "thing operated on" at E3, since replacing "foo" by "blee" in E3 would make "thing operated on" undefined. But then the use of "blee" in "blee baz" would have to make "thing operated on" undefined in E2, which is false. Similarly "operation performed" must be defined for "blee bar", since otherwise the use of "bar" in E3 would not work. As worked out above, it follows from these consequent aspects being defined for "blee baz" that their values are "terminal assignment table" and "printing", as required.

Just as assumptions are needed to derive the obvious generalization from E1-3, natural interpretation of other examples requires further assumptions. Consider the following commands and outcomes.

E4: z n f archives the current audit file with classification "HOLD".
E5: k n d transmits the customer feedback file with classification "HOLD".

What part of the first command specifies the classification? Eleven of eleven computer scientists gave the obvious answer: "n". The assumptions stated so far, however, are not adequate to justify this conclusion. Space does not permit a complete analysis, but we need the further assumption that there is some antecedent aspect of E4 that controls the classification. We can state this in general as

A4 *No free consequents*: Every defined consequent aspect of a case is controlled by some antecedent aspect.

We also need any of the following three further assumptions:

A5 *No aliasing*: if a controls c at C and C', and c(C')=c(C), then a(C')=a(C).

A6 *No extra baggage*: Every defined antecedent aspect of a case controls some consequent aspect.

A7 *One consequent per antecedent*: Any antecedent aspect of C controls at most one consequent aspect.

If we are willing to accept violations of A5-7 then the following analysis of E4-5 would be OK: "z" and "k" are both aliases for "classification HOLD", "n" does nothing, and the third letter controls both the operation and the file operated on. Any one of A5-7 suffices to rule out this interpretation and any other in which "n" does not specify the classification.

One further assumption is needed to justify the obvious interpretation of A5-7: we need to rule out the possibility that some other aspect of the examples, in addition to the middle letter, also controls the classification. We assume directly

A8 *No multiple control*: Any consequent aspect is controlled by at most one

antecedent aspect.

The arguments just given show that natural generalizations depend on strong but plausible assumptions about the domain. If people use these assumptions we can explain the generalizations they make. But the evidence about the assumptions is indirect.

To seek direct evidence, we asked eleven computer scientists (faculty and graduate students in computer science at the University of Colorado) to choose between contrasting explanations of eight groups of example commands. For each assumption A1-8 we devised examples for which we could provide one explanation that was consistent with all of the assumptions, and a second which violated the selected assumption and satisfied the remaining ones. Figure 1 shows the item devised for A8. Participants were asked to say which explanation they regarded as "more likely".

The results are summarized in Table 1. As can be seen, most participants rejected explanations violating A1, Consistency of control, A2, Consistency of consequent values, A5, No aliasing, A6, No extra baggage, and A8, No multiple control. Assumptions A4, No free consequents, and A7, One consequent per antecedent, were favored but some participants attached no weight to them. The p-values shown are for the sign test.

Assumption A3, Consistency of invalid values, was not strongly supported by the participants. Participants were asked to explain their preferences, and only one provided a justification for choice on the A3 item which resembled A3. It is possible that people arrive at the common generalization from E1-3 using some alternative to A3, such as assuming directly that values of consequents are defined, in the absence of indications to the contrary.

Though generalizations based on the assumptions seem natural and robust, any of the assumptions can be and are false in some situations. The participants just discussed were aware of this: when asked whether a rejected explanation was actually impossible, only three participants indicated that any of the explanations were impossible. Only the violation of A2, Consistency of consequent values, was called impossible by more than one participant. But even this assumption is quite often violated in real systems. For example, the same operation symbol, "+", denotes different arithmetic operations for different data types in most programming languages. Of course in this case users often think of all of the different operations as "addition", but in Ada it is possible to assign unrelated operations to the same identifiers, with the choice of operation dependent on the types of parameters.

Since people are aware that violations of the assumptions are possible, reliance on the assumptions has only heuristic value. People must therefore be prepared to deal with violations of the assumptions, but how they respond to violations remains to be explored. We do have evidence that people will sometimes work to avoid confronting violations, by reinterpreting apparently deviant examples in such a way as to preserve the assumptions. Robert Mack (personal communication) found that

participants who were asked to view and then report on demonstrations of the use of a text editor sometimes said that they must have missed an action that selected insert mode, though in fact the system in question was always in insert mode and no such action was shown. These participants apparently saw "insert mode" as a consequent aspect in need of a controlling antecedent aspect. In our own work we asked participants to recall and explain sequences of screen contents and commands that included those shown in Figure 2a and 2b. Six of thirty participants explained the sequence in Figure 2a by providing a role for the seemingly superfluous "Z". Five of the six proposed that the "Z" helped to identify which instance of "P" was to be deleted, specifying the "P" that follows a "Z". Seven of a different thirty participants who saw the sequence in Figure 2b managed to connect the apparently excess consequent, the deletion of "P", to the antecedent value "Z". Six of these described the operation as removing Z and any following letters.

Since users nearly always must develop generalizations of the kind we have been discussing in using a system, designers should strive to support them. Clearly the designer of the system from which E1-3 are drawn would be in trouble if the command to print the terminal assignment table were anything other than "blee bar". Design to support generalization goes under the heading "consistency", and so our framework can be seen to provide a specific analysis of what consistency means. Designers can produce consistency by honoring assumptions A1-8: if these assumptions are not violated users' generalizations will be successful.

This analysis of consistency differs from previous ones in connecting consistency directly to the generalization process it is intended to support. Reisner's (1981) use of formal grammars in analyzing interfaces, work on task-action grammars (Payne and Green, 1983; Payne, 1984) and the Kieras and Polson use of productions to measure the overlap in required knowledge among tasks (Kieras and Polson, 1985; Polson, Muncher, and Engelbeck 1986) do not illuminate how the regularities they promote could be used actually to construct generalizations.

Assumption A5, No aliasing, is an example of an assumption whose status is clarified by the present analysis. The earlier analyses would suggest that aliases would be undesirable only if users learned to use them: only then would extra grammar rules or productions be needed to describe alternative commands. But our analysis suggests that aliases are more broadly difficult. First, learning from examples that include aliases is hard. Second, even the *possibility* of aliases can complicate the analysis of any example, even if aliases *are not actually involved*, as seen in E4-5.

Our direct evidence about the assumptions is drawn from a sophisticated population. Are the assumptions relevant to designing for more naive users? Indirect evidence argues that they are. Recall that the assumptions were developed to support certain robust and natural generalizations: anybody, sophisticated or not, who makes these generalizations needs some such assumptions. No doubt the assumptions are reinforced for experienced users by the fact that they are often (though by no means always) found to be true. But we expect to find that even very naive users will implicitly rely on the assumptions.

Table 1

Assumption	Prefer explanation consistent w. assumption	Prefer explanation violating assumption	No preference	p
A1: Consistency of control	9	2	0	.1
A2: Consistency of consequent values	11	0	0	.01
A3: Consistency of invalid values	6	4	1	--
A4: No free consequents	8	1	2	.05
A5: No aliasing	10	1	0	.05
A6: No extra baggage	11	0	0	.01
A7: One consequent per antecedent	7	2	2	.25
A8: No multiple control	9	2	0	.1

Figure 1: Item used to assess acceptance of A8, No multiple control.

Here are three commands in ExPox and two explanations of them:

ghu fli wef

nhe fli wef

ghu cve wef

Explanation 1: The first command deletes the file wef. The second command prints the file wef. The third command archives the file wef.

Explanation 2: The first command deletes the file wef permanently. The second command deletes the file wef provisionally. The third command locks the file wef permanently.

Which explanation do you think is more likely?

Figure 2a: Sequence to recall and explain (see text).

Contents of screen: DZP

Command: remove ZP

Contents of screen: DZ

Figure 2b: Sequence to recall and explain (see text).

Contents of screen: DZP

Command: remove Z

Contents of screen: D

Acknowledgement

This research was supported by the Office of Naval Research under Contract No. N00014-85-K-0452.

References

Kieras, D.E. and Polson, P.G. (1985) An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22, pp.365-394.

Lewis, C.H. (1986) A model of mental model construction. In *Proceedings of CHI'86 Conference on Human Factors of Computing Systems.* (New York, ACM) pp. 306-313.

Lewis, C.H., Casner, S., Schoenberg, V. and Blake, M. (1987) Analysis-based learning in human-computer interaction. In *Proceedings of INTERACT '87.*, Elsevier Science Publishers, pp. 275-280.

Lewis, C.H. (1988) Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science*, 12, pp. 211-256.

Mackie, J.L. (1974) *The Cement of the Universe.* Oxford: Clarendon Press.

Mill, J.S. (1900) *A system of logic.* London: Longmans.

Payne, S.J. and Green, T.R.G. (1983) The user's perception of the interaction language: A two-level model. In *Proceedings of CHI'83 Conference on Human Factors of Computing Systems.* (New York, ACM) pp. 202-206.

Payne, S.J. (1984) Task-Action Grammars. In *Proceedings of INTERACT'84*, Elsevier Science Publishers, pp. 139-144.

Polson, P.G., Muncher, E., and Engelbeck, G. (1986) A test of a common elements theory of transfer. In *Proceedings of CHI'86 Conference on Human Factors of Computing Systems.* (New York, ACM) pp. 78-83.

Reisner, P. (1981) Formal grammar and design of an interactive system. *IEEE Transactions on Software Engineering*, 5, pp. 229-240.

Distribution List

Dr. Susan Chipman
Office of Naval Research
800 N. Quincy Street
Arlington VA 22217

DTIC
Cameron Station
Alexandria, VA 22304-6145

Dr. Max Irving
Department of the Navy
University of New Mexico
Bandelier Hall West
Albuquerque, NM 87131

END

6-89

DTIC