Explanation and Learning in Procedural Skills

Clayton Lewis

CU-CS-436-89 April 1989

University of Colorado at Boulder

DEPARTMENT OF COMPUTER SCIENCE

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

Explanation and Learning in Procedural Skills

Clayton Lewis

CU-CS-436-89 April 1989

Department of Computer Science
Campus Box 430
University of Colorado,
Boulder, Colorado, 80309

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 Cognitive Science.

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.