Int. J. Man-Machine Studies (1985) **22**, 1-10

What non-programmers know about programming: Natural language procedure specification

KATHLEEN M. GALOTTI

Department of Psychology, University of Pennsylvania, Philadelphia, Pennsylvania 16802, U.S.A.

WILLIAM F. GANONG III

Kurzweil Applied Intelligence Inc., Waltham, Massachusetts, U.S.A.

(*Received 17 April 1983, and in revised form 22 November 1983*)

Part of learning to program involves learning to use control statements. This study examined the spontaneous use by non-programmers of control statements when writing instructions. Previous work suggested that non-programmers rarely use such statements in their instructions. The present findings demonstrate that non-programmers can and do use control statements when their use is appropriate for the recipient of the instructions. Some methodological problems involved in the demonstration of basic competence are discussed.

Introduction

As the number of people who program computers grows, the question of how best to teach non-programmers to program becomes increasingly important. Recent discussion of this issue (see, for example, duBoulay, O'Shea & Monk, 1081; Gould, Lewis & Becker, 1976; Green, 1980; Miller, 1981; Soloway *et al.*, 1981) has focused on understanding the needs of the naive user. We wish to underscore the need to make appropriate assumptions about the novice programmer. In contrast to a decade ago, we can no longer assume that the would-be programmer has an appreciable background in engineering, mathematics or logic; nor that he or she will have the necessary patience and motivation to tolerate obstacles to learning, such as complicated and awkward syntax and semantics of some programming languages. If we take seriously the goal of making systems "user-friendly", then our goal must be to understand the naive user's beliefs, strategies and models of computers and computer use. The aim of this paper, then, is to provide a partial description of one component skill naive users bring to the task of programming.

An obvious way of discovering initial and basic competence is to study the performance of novices on a task analogous to programming, and to try to characterize the knowledge they use in such a task. This will help to describe: (1) what the novice already knows and (2) what the novice needs to learn. In addition, the characterization itself may provide hints of how best to design a training process. One task analogous to programming is writing instructions for other people to carry out, or more formally, "procedure specification".

This task has several features which invite study. First, it has the advantage that novices have experience both giving and following instructions, so no special experimental training is required. Any competence, knowledge or strategies employed in this task can be presumed to be a part of the repertoire of most novices. Also, instructions

0020-7373/85/010001+10$03.00/0

can incorporate various formalisms, such as numbering or labelling, diagrams or flow charts. Such devices are often found in programming languages, or used in the course of writing programs. Discovering and describing the spontaneous use of these devices will presumably help us to help the novice in using analogous devices when learning to program.

While the idea of using analogies or models in instruction is not a new one, it has recently become clear that not every analogy will be effective or even innocuous (duBoulay *et al.*, 1981; Halasz & Moran, 1981). One step toward finding useful analogies is to begin with an understanding of what analogies computer novices spontaneously make, then to evaluate the effectiveness of such models.

Preliminary work on procedure specification by Gould *et al.* (1976) examined how computer novices wrote (and followed) instructions. The tasks subjects worked on involved either assembling children's blocks, or typing a certain pattern of characters. Whether asked to write their instructions in the form of a "detailed description", a "procedure or plan", within an experimenter-supplied syntax, or with no restrictions on format, subjects were equally accurate and fast in both generating and following instructions.

A detailed study of procedure specification has been carried out by Miller (1981). Fourteen non-programmers wrote instructions for a fictitious file clerk who was to perform several clerical tasks, such as making a list of company employees who had various attributes (for example, a job rating of "superior" and a job title of "photographer") in a fictitious company. Such tasks often involved accessing and searching through several company files, and/or cross-checking information from two files. Miller's aim was to investigate the feasibility of using English as a programming language, and toward this end, he performed several detailed analyses. The one we will focus on here concerns the overall use of control statements.

Control statements (Miller calls them transfer of control statements) describe the order in which various actions are to be carried out. Rather than being instructions to perform a certain action, control statements are instruction about when or how often other instructions are to be followed. Some examples are: conditionals, such as if–then statements, which describe conditions that must be met before an action is carried out; iteration statements, which designate how many times an action or actions are to be repeated; and GOTO statements, which direct the instructee to look elsewhere in the instructions to find the next instruction to be carried out. It is worth noting that other research has examined the relative ease of learning particular syntactic instantiations of some of these statements (see, for example, Green, 1977; Sime, Green & Guest, 1973, 1977), but Miller was the first to examine the spontaneous use of these statements in a more natural, everyday sort of task.

Miller coded his texts into various content categories. The one of interest here was designated "transfer of control". His basic finding was that the coded texts used this category very infrequently. The percentage of codes in the "transfer of control" category, as a fraction of all codes, averaged about 10%. As a comparison figure, Miller cites work by Knuth that estimated that 22% of the commands in FORTRAN programs written by novices are control statements. Miller considered this difference a major one between "natural" instructions and computer programs.

It might be tempting to accept this result at face value, and to base training programs, or design new "non-procedural" programming languages upon this finding, on the

grounds that worrying about flow of control is a rather "unnatural" thing for non-programmers to do. However, there is at least one alternative explanation of the results, as Miller himself points out. What one writes explicitly and what one leaves implicit in a written document depends on one's assumptions about the pragmatics of the situation. In particular, it is bad form to belabour the obvious. Subjects in Miller's experiment may have left out control statements not because such statements are too difficult, but because they are too obvious. For example, consider typical instructions printed on shampoo bottles, which read, "Wet hair, apply shampoo, rinse, and repeat". On an exact reading, these instructions are hopelessly imprecise, yet it is highly implausible that anyone has ever iterated endlessly through the loop, forever a prisoner in a shower stall. Our actions are governed not only by the instructions, but by inferences about the instructor's intentions. Anyone who writes instructions knows this. Indeed, were the directions made more precise, as in, "1. Wet hair. 2. Apply shampoo. 3. Rinse. 4. Repeat steps 2 and 3 one time only", we might find them either laughable or insulting. Thus, one reason Miller's subjects may have not used many control statements is that they presumed that the instructee would possess a modicum of "common sense" and thus thought control statements were obvious.

The argument that people take into account a model of their listener when planning what to say and how to say it is not new with us. Grice (1975) detailed a set of maxims, or rules of co-operative conversation, which he claims speakers and listeners implicitly follow. Summarized briefly, these are: (1) be no more or no less informative than is required; (2) be truthful; (3) be relevant; (4) make your contribution easy to understand; avoid ambiguity and obscurity. Violations of these rules lead the listener to assume infelicity on the part of the speaker. For example, the following two exchanges violate the first maxim: A "Do you know what time it is?" B: (glancing at wristwatch) "Uh-huh". A: "Do you know what time it is?" B: "Eleven-seventeen and 52 seconds on 12 November 1983".

We argue that Miller's subjects may have been following the first maxim in their instructions. They did not want to include obvious information which they presumed any listener could supply.

We tested this intuition by asking a small sample of college students ($N = 5$) to write instructions telling an English-speaking Martian how to perform three simple tasks (play the card game War; collate 10 copies of a 35-page manuscript; sort a pile of records into two piles, based on several criteria). Our instructions encouraged subjects to view the Martian as lacking common sense, especially about when to start or stop doing something. The tasks were chosen with malice aforethought; we presumed they would be familiar to our subjects, and that they would elicit both repetition and conditional statements. Both intuitions proved correct. The average proportion of control statements was about 30%, a finding discrepant from that of Miller. We then tried to replicate Miller's finding, by having another group of subjects perform the task, writing this time for "a person (lacking common sense)", but still found that 34% of the statements were control statements. This last finding was troublesome (we had thought subjects would delete control structures when writing for a person), so we tried to discover factors that could account for the discrepancy between our results in this second condition and those of Miller.

Three came immediately to mind. One was that our sample of subjects differed: in particular, we had some programmers in our sample. A second difference concerned

the tasks we chose—perhaps Miller's file-searching task required fewer decisions and, hence, fewer control statements. A third concerned differences in procedure: Miller required his subjects to write individual steps that were independent and less than 80 characters each (they typed at CRT terminals), while our subjects wrote in longhand, with no restrictions. To investigate whether one or more of these differences accounted for the difference findings, we designed our present study as follows. First, only subjects completely unfamiliar with computer programming were used. Second, we had subjects write instructions for 2 tasks—(1) playing War and (2) one of Miller's file-checking tasks. Each subject wrote either for "an English-speaking Martian with no common sense" or else for "a person like yourself". Finally, half of the subjects wrote their instructions under restrictions similar to the ones imposed by Miller (at least five steps, each step no longer than 80 characters, each step independent), while the other subjects wrote according to whatever format they chose. Our aims in doing this study were two-fold: one, to see if computer novices do possess an untutored ability to use control statements and, second, to see under what circumstances this ability is best demonstrated.

Method

SUBJECTS

Thirty-two undergraduate students and staff members at the University of Pennsylvania were recruited by poster, and were paid $3·00 for a 40-min task. Subjects who signed up on the poster were telephoned and interviewed as to their academic background, hobbies, etc. In this way, we were able to be sure that no subject in our study had any background in computer programming, without directly calling to subjects' attention our interest in their computer experience. One subject had to be dropped due to failure to comprehend the instructions, and was replaced. Subjects were randomly assigned to conditions.

MATERIALS

At the start of the experiment, each subject was given a packet of materials, including instructions, a description of the tasks they were to describe in their own instructions and coding sheets on which to write their answers. The instructions stated that we were interested in how people write directions. The recipient of the instructions (either a Martian or a person) was described, and subjects were told to write carefully. (Subjects writing for the Martian were additionally given an example about the Martian trying to follow directions on a shampoo bottle, but being defeated by the ambiguity of the instructions.) Attached to the instructions were two pages, describing the tasks of playing the card game War or the task of making a list of certain employees from information in company records. (A copy of the instructions used is provided in Appendix 1.) The order of these two pages was counterbalanced.

Subjects also received several blank FORTRAN coding sheets on which to write their responses. (These sheets had no indication as to their original purpose.) It was explained that subjects needed to print their answers, one letter per box, on the sheets so that we would later be able to decipher handwriting. All subjects complied readily with this request. Half of the subjects were instructed to write their instructions subject

to the following restrictions: (1) the instructions had to have at least five steps: (2) each instructional step was to be independent; (3) each instructional step had to fit on one line of the coding sheet (80 blocks wide). The other half of the subjects were not subject to any such restrictions.

PROCEDURE

Subjects were run either singly or in groups of two. All instructions were in writing. At the end of the session, subjects were debriefed as to the purposes of the study. Later, protocols were typed up, and distributed to coders.

CODING OF RESPONSES

In order to assess the reliability of our coding scheme, we asked five graduate students and research assistants to code each sentence in each protocol. Both of the authors and one research assistant coded every protocol. The remaining four coders each scored half of the protocols. All of the coders (except the authors) were blind to the purposes of the study, as well as to the condition each subject was in. The second author was also blind to the subjects' conditions.

Our coding scheme had five major categories, consisting of:

(1) actions—the statements that refer to simple operations, either physical or otherwise that the agent should perform;

(2) modifiers—statements that describe and limit the way in which other instructions are to be carried out;

(3) declarations—statements that describe objects or current conditions;

(4) control statements—statements which describe WHEN and HOW often various actions are to be carried out; and

(5) other—statements that do not fit the above scheme.

There were also distinctions drawn within these categories, but these are not relevant for our present purposes. Coders could give as many codes to a sentence as they deemed necessary, so long as they indicated which part of the sentence belonged to which code.

Results

Overall reliability (number of agreements/number of agreements + number of disagreements) averaged only 0·51, pairwise between any two coders. However, we determined a consensus code for each sentence. This was the code or codes most frequently given by the coders. The average agreement between coders and the consensus code was 0·67. We looked next at reliability for coding a sentence or sentence fragment as a control statement. Overall, the agreement between any two coders was 0·68, while the agreement between any given coder and the consensus code, for control statements only, was 0·77. This indicates that reliability is low but satisfactory, and suggests that (1) our notion of control statements is one others can understand and that (2) the results below reporting the spontaneous use of control statements are reliable, and not an artifact of an eccentric author or two. In Appendix 2 we present some examples of protocols, with the consensus codes indicated.

For each sentence in each protocol, we counted the number of control statements (from the consensus code) as a fraction of the total number of codes. This dependent measure was used in a 2 (recipient—Martian or person) $\times 2$ (restriction—restricted or unrestricted form) $\times 2$ (task—War or list) ANOVA, with repeated measures on the last factor. Results showed, first, a main effect for the factor of recipient, with subjects writing for Martians using significantly more control statements than subjects writing for persons [$X = 21{\cdot}91$ and $12{\cdot}53$, respectively; $F(1, 28) = 18{\cdot}37$, $P < 0{\cdot}01$, accounts for 19% of the total variance by ω^2] [see Linton & Gallo (1975) for details on this measure]. There was also an effect for task. Subjects used more control statements when instructing about the card game than when writing about the clerical task [$X = 20{\cdot}41$ and $14{\cdot}03$, respectively; $F(1, 28) = 10{\cdot}06$, $P < 0{\cdot}01$, accounts for 8% of the total variance]. Finally, there was an interaction between these two factors [$F(1, 28) = 11{\cdot}49$, $P < 0{\cdot}01$, accounts for 10% of the total variance]. Table 1 presents the cell means for the interaction. Subjects instructing a person to make a list of employees use significantly fewer control statements than did either (1) other subjects writing for a Martian or (2) the same subjects instructing a person to play War. Note that this condition is the one most similar to the one used in Miller's study. No other comparisons were statistically reliable. Thus, restricting the form of the instructions subjects wrote had no appreciable effect on the spontaneous use of control structures, contradicting one of our initial hypotheses.

TABLE 1
Percentage control statements used by recipient and task

Martian—War	Martian—List
18·56	22·13
Person—War	Person—List
19·13	5·94

Pooled estimate of S.E.M. = 2·01.

We also looked at gross measures of the form of subjects' protocols. The first measure was of length—that is, number of sentences in the protocols. The $2 \times 2 \times 2$ mixed ANOVA was run on this dependent measure. The only reliable effect found was for the factor of recipient. Subjects writing for Martians had more sentences in their instructions than subjects writing for a person [$X = 14{\cdot}88$ and $8{\cdot}78$, respectively; $F(1, 28) = 15{\cdot}13$, $P < 0{\cdot}01$, accounts for 24% of the variance].

The second measure we looked at was the total number of codes given to a protocol (again, using consensus codes). We use this as a rough estimate of how much information the subject provides to the recipient of the instructions. The same $2 \times 2 \times 2$ ANOVA run on this dependent measure yielded only an effect for recipient: protocols written for a Martian received significantly more codes than did protocols written for a person, [$X = 21{\cdot}44$ and $11{\cdot}88$, respectively: $F(1, 28) = 20{\cdot}69$, $P < 0{\cdot}01$, accounts for 30% of the total variance].

Discussion

This study demonstrates that, under certain conditions, non-programmers do spontaneously use control statements, and do so in about the same proportion as do novice programmers writing FORTRAN programs. This suggests that whatever the obstacles to learning to program are, the basic idea of flow of control is not one of them. This is not to suggest that our subjects' instructions used control statements whenever needed, or in the right places. As the examples in Appendix 2 show, subjects' instructions about iteration or about what to do if a test condition is not met are often vague or unspecified. None the less, our findings do suggest that the untutored person does have some rudimentary knowledge of what a control statement is.

Second, we offer a methodological point. The demonstration of underlying competence often requires special circumstances. Communication to another person is quite unlike computer programming, since speakers and listeners often "go beyond the information given" in instructions or conversations, making inferences as to the other's beliefs, knowledge and intentions. To uncover this type of competence, researchers must take care to minimize pragmatic factors which lead the subjects to leave the obvious unstated.

The basic idea here is that performance on tasks designed to measure an underlying ability might be sensitive to other abilities or factors unrelated to the one of interest. In other words, performance on a task may not always reflect competence, or underlying ability, because of factors specific to the particular task.

We hope in future work to carry out more detailed analyses of the form of "natural" control statements, to see where and when they are used, and what form they take. We will also look at the overall structure of written instructions, examining what kinds of information is included, and where and when it is included. We hope that the study of written instructions will uncover principles that are useful in teaching non-programmers to program.

We thank Warren Stewart, Diane Bloch, Marcia Glicksman, Tim Hyrnick and Carol Smith for coding protocols, and Barbara Grosz, Bonnie Lynn Webber, Ellen Prince and Lloyd Komatsu for profitable discussions. We also thank an anonymous reviewer for his helpful comments. Results of preliminary studies were presented at the Conference on the Use of Very High Level Computer Languages and Systems, University of Pennsylvania, in February 1982. Requests for reprints should be sent to the first author, now at the Department of Psychology, Carleton College, Northfield, MN 55057, U.S.A.

References

DUBOULAY, B., O'SHEA, T. & MONK, J. (1981). The black box inside the glass box: presenting computing concepts to novices. *International Journal of Man-Machine Studies*, **14**, 237-249.

GOULD, J. D., LEWIS, C. & BECKER, C. A. (1976). Writing and following procedural, descriptive, and restricted syntax language instructions. *Memo No. RC 5943*, IBM Thomas J. Watson Research Center, New York.

GREEN, T. R. G. (1977). Conditional programming statements and their comprehensibility to professional programmers. *Journal of Occupational Psychology*, **50**, 93-109.

GREEN, T. R. G. (1980). Programming as a cognitive activity. In SMITH, H. T. & GREEN, T. R. G. Eds, *Human Interaction with Computers*. New York: Academic Press.

GRICE, H. P. (1975). Logic and conversation. In COLE, P. & MORGAN, J., Eds, *Syntax and Semantics III: Speech Acts*. New York: Academic Press.

HALASZ, F. & MORAN, T. P. (1982). Analogy considered harmful. *Proceedings of the Gaithersburg Conference on Human Factors in Computer Systems*, Gaithersburg, Maryland. Published by the Association for Computing Machinery.

LINTON, M. & GALLO, P. S., JR (1975). *The Practical Statistician.* Belmont, California: Wadsworth Publishing Co.

MILLER, L. A. (1981). Natural language programming: Styles, strategies and contrasts. *IBM Systems Journal*, **20**, 184-215.

SIME, M. E., GREEN, T. R. G. & GUEST, D. J. (1973). Psychological evaluation of two conditional constructions used in computer languages. *International Journal of Man-Machine Studies*, **5**, 105-113.

SIME, M. E., GREEN, T. R. G. & GUEST, D. J. (1977). Scope marking in computer conditionals. *International Journal of Man-Machine Studies*, **9**, 107-118.

SOLOWAY, E., BONAR, J., WOOLF, B., BARTH, P., RUBIN, E. & EHRLICH, K. (1981). Cognition and programming: Why your students write those crazy programs. *Proceedings of the National Educational Computing Conference*, Texas, June 1981 (to appear).

Appendix 1: Instructions to subjects, by condition

MARTIANS—RESTRICTED

This study investigates how people write instructions. Your task is to instruct an English-speaking Martian how to perform the two tasks shown on the following pages. This Martian can read English sentences, but has no other experience with any tasks like the one you'll be instructing him in. Also, the Martian possesses no common sense whatever. For example, if he were to read a shampoo bottle with directions like, "Wet hair, apply shampoo, rinse and repeat", he wouldn't know which step to repeat (he'd wonder if that meant he should just keep rinsing, or if he had to wet his hair again, or what) and when to stop. So, your directions must be complete and precise, and should concentrate on telling him when to do what, and when to stop doing it.

Please write your answers on the forms provided, putting one letter in each little box. Write your answer as a series of steps, each of which describes a different action. You must fit each step on one line of the response sheet.

Your answers should contain at least five steps, but you may have as many more than that as you like. If you need to make any corrections, please just rewrite the line(s) you want to change.

Take as long as you need to do this. Please also do the tasks in order, completing the first set of directions before you start the second.

TASKS USED

War

Tell your "instructee" how to play the card game, War. Assume that there is a deck of cards, one other player to play with, a table and two chairs.

List

Tell your "instructee" how to make a list of certain employees of the "Famous Frammis" company. To do this, you first need to know about some of the company files.

Each file contains several "records". Each record contains information about one employee, and is written on a 3×5 card. The personal file is arranged alphabetically by name, which is the first item on the record. Second item is employee number; third

item is age at last pay period; fourth is marital status. One "record" in this file has the form:

1. Name
2. Number
3. Age
4. Marital Status

The job file contains records organized in terms of increasing employee number, which is the first item on the record. Second item is job title; third is year in which employee was hired; and fourth is supervisor's rating of employee performance. So, one record from this file has the form:

1. Number
2. Job Title
3. Date Hired
4. Rating

With this in mind, your job is to tell your "instructee" how to make a list of all employees who are 64 or more years old and who also have 20 or more years of experience. The list should be organized by employee name.

Assume the two files are sitting on a desk, and that there are pens and pads of paper to work with.

PERSON—RESTRICTED

This study investigates how people write instructions. Your task is to instruct a person like yourself how to perform the two tasks shown on the following pages. You should therefore write your instructions so as to be easily understood and executed by a person similar to you.

(The final three paragraphs from the Martian-Restricted condition, along with a description of the two tasks, followed.)

MARTIAN—UNRESTRICTED

(The first paragraph from the Martian—Restricted condition was used.)

Please write your answers on the forms provided, putting one letter in each little box. If you need to make any corrections, please just rewrite the line(s) you want to change.

You may organize the directions in anyway you like.

(The fourth paragraph from the Martian—Restricted condition, along with a description of the two tasks, followed.)

PERSON—UNRESTRICTED

(The first paragraph from the Person—Restricted condition was used, followed by the final three paragraphs from the Martians—Unrestricted condition, and a description of the two tasks.)

Appendix 2: Example protocols with consensus codes

(Note: A = Action Statement; D = Declaration; M = Modifier; CS = Control Statement; O = Other.)

EXAMPLE 1: WAR (CONDITION: MARTIAN—RESTRICTED)

Ask the other person if he would like to play a card game with you. (A)

If he says yes (CS) then ask him to sit at the table with you. (A)

You give half of the number of cards to your player (A) and you keep 26, too. (A)

To begin, each of you places a card on the table (A) face up (picture-side down). (M)

Whoever put down the card with the highest number on it takes both cards to keep. (A)

The goal is to win all of your player's cards (O) (your player has the same goal too). (O)

The cards with the pictures of people, not numbers, are worth more than ten. (D)

Repeat the *process* of putting down a card. (CS)

Repeat the *process* of picking up cards, depending on who put the highest numbered card down. (CS)

If both players put down the same number card (CS) this is a war. (D)

To play a war, (O) both put down three cards, (A) face (of cards) down next to face-up card. (M)

Next to face down cards, both put down a face-up card. (A)

The person who put down the card with the highest number or picture wins. (D)

The winner takes and keeps all the cards on the table from the war. (A)

Continue repeating the aforementioned *processes*, til one player reaches his goal. (CS)

The player to reach goal first is the winner. (D)

EXAMPLE 2: LIST (CONDITION: PERSON—UNRESTRICTED)

Begin this list by going to the personal file. (A)

Go through these files (A) and note the name and employee number of each employee who is 64 years or older. (A)

Take this list to the job file. (A)

Since the first list will be in alphabetical order, (O) take the first name from the list and add it to his employee number card in the job file. (A)

If he has been working for the firm for 20 years or more, (CS) put his name on the top of the list. (A)

Do the same for all the names on the first list. (CS)