SESSION VI CONTRIBUTED PAPERS: PROGRAMMING LANGUAGES AND SYSTEMS

JOHN F. KING, *Illinois State University, President*

S.I.M.P.L.E. control and measurement of auditory events

DORIS AARONSON, EDWARD GRUPSMITH, and DOUG OHMAN

New York University, New York, New York 10003

A SIMPLE interactive programming language is shown to be useful for psychological research involving auditory stimuli and responses. It is helpful in recording spoken stimulus tapes with program control of event sequencing and timing; it can control on-line interactive auditory and bisensory experiments; and it can calibrate the stimulus durations and pause times for auditory events.

Psychology minicomputer users have discovered the advantages of developing general purpose, high-level, user-oriented programming languages. However, few of these high-level languages are currently being used to handle experiments with spoken stimuli or responses. We would like to discuss here some general guidelines for computer control of auditory events, and one user language that is capable of handling all phases of an auditory experiment. This particular language was originally designed to control visual displays, as is the case for most psychology computer languages. In that context, it was named S.I.M.P.L.E., Scope Interpreter for Memory, Perception, and Learning Experiments. However, the generality of its logical command structure also enables it to handle auditory events.

The SIMPLE system has three properties that are generally important to any user language: it handles a relatively complete cognitive laboratory, it meets the precision needed by experiments, and it is compatible with a variety of hardware. First, as a self-contained audio-visual lab, SIMPLE can generate and synchronize auditory and visual warning signals,

sequential displays, and feedback information, as well as collect vocal and motor response data. Second, SIMPLE meets current experimental needs. It provides the precision and reliability required by many experiments using auditory events, particularly those using speech. For memory and language studies, SIMPLE can generate single-channel or dichotic sequences with stimulus duration, inter-stimulus pause time, and between-channel synchronization controlled to a few milliseconds. In comparison, when natural speakers generate stimuli, not only are they highly variable, but the durations of their words and pauses are systematically confounded with each other and with their serial position (Hutton, 1954). On the response side, SIMPLE can record latencies and durations of subjects' spoken words with millisecond accuracy. This would enable the user to obtain a fuller picture of response patterns than is usually obtained without automated equipment. Third, consider compatibility. SIMPLE is logically compatible with or easily adaptable to a wide variety of laboratories. This is because the individual user commands are natural functional units for many experimental paradigms and for many hardware configurations, and because these command units can be sequenced with few constraints other than those of the experimental design. Hence, when SIMPLE is implemented in other labs, the assembly language reprogramming needed to meet particular device specifications will be confined to the few subroutines that directly control each device.

In this paper, we will first describe SIMPLE's hardware facilities, and give a brief overview of the

This research was supported in part by Grant MH-16,496 from the National Institute of Mental Health. This paper was presented at the Fourth Annual Meeting of the National Conference on On-Line Computers in Psychology, November 20, 1974, held at Northeastern University, Boston, Massachusetts. Steven Brauth helped in developing the SIMPLE language. We thank Stephen P. Cohen for helpful comments on an earlier draft. Requests for reprints should be sent to D. Aaronson, Psychology Department, 4 Washington Place, Room 858, New York University, New York, New York 10003.

Figure 1. PDP-8/I computer system used with SIMPLE at New York University.

programming commands available to the user. More extensive descriptions of the hardware and software, oriented toward the visual display capabilities, have been published elsewhere (Aaronson & Brauth, 1972; Brauth & Aaronson, 1972). Then we will discuss three ways in which SIMPLE can control and measure auditory events. Together, these three ways enable the user to handle all aspects of a large class of auditory psychology experiments.

SIMPLE HARDWARE

Figure 1 is a diagram of the SIMPLE hardware at N.Y.U. SIMPLE runs on a PDP-8/I computer with 8,192 core locations, Teletype, crystal clock, and high-speed reader/punch. The PDP-8/I is interfaced with a Computer Communications Corp. CC-30 communications station. This includes a programmable visual display on a standard TV monitor (Sony, with P-4 phosphor), an ASCII-coded hardware character generator (20 lines of 40 characters each), with its associated 1,024-word memory buffer, and an alphanumeric response keyboard. In addition, a relay buffer was added to activate other types of external displays. These 12 relays can be set and cleared to operate lights, tape recorders, tactile displays, tone and noise generators, movie or slide projectors. SIMPLE senses and operates on the human subject's responses from the CC-30 typewriter-style photoelectric keyboard. Also, switch-closure responses are sensed from homemade special-purpose buttons and keys substituted into the CC-30 keyboard interface and from a 12-bit data collector (i.e., sense switch

register) directly connected to the PDP. Hence, many response sources are possible, including voice-operated relays, telegraph keys, special pens, and pressure-sensitive writing boards. SIMPLE records these events, as well as their latencies, from the onset or offset of any specified stimulus display or from a preceding response.

SIMPLE has seven classes of user commands, as illustrated in Figure 2. *Buffer* commands operate the I/O devices to read input into or punch output from a temporary buffer core storage. When an experimental trial has started, *Display* commands select the position, the content, and then illuminate the scope based on three sources: the text buffer, input with the buffer command, the user's program, and the subject's responses. Alternatively, *List Processing* commands permit the user to select display contents from a stored list in systematic or random orders and to select the starting scope positions for these displays. To activate auditory stimuli, *Interface* commands set or clear patterns of relays to gate-in tones or to operate tape recorders. *Keypress and Timing* commands wait for the subject's response to these stimuli, record the response and its latency from the display or from a preceding response, and branch contingent on the response data. *Arithmetic and Logic* commands alter the flow of program control by executing jump statements based on the results of arithmetic operations or on the contents of a counter or other special registers. *Read* commands read

Figure 2. SIMPLE's seven classes of user commands.

information into those registers from the input text buffer or from the user's program. Further, *Load* commands access the subject's keypresses and RTs from a special keypress array to permit tests on the data in conjunction with the arithmetic variables. Thus, these three classes of commands are used together to determine branching patterns for providing feedback and subsequent stimulus displays.

Now let us look at some examples of these commands in Figure 3. Each command consists of a single letter mnemonic, which is sometimes followed by a symbol or a number. *Buffer*: B reads from the punched paper text tape the scope display contents and the variable values needed by the program. B* punches onto tape the subject's data and variable values generated during the experiment. *Display*: Dnnnn displays from the input text buffer the number of text blocks specified by the nnnn value. D displays the subject's alphanumeric responses until he presses a "return" key. *Interface*: Innnn sets the 12-bit relay buffer according to the binary equivalent of the octal nnnn value. For example, 3776 octal equals 011 111 111 110 binary, and would close the middle relays, leaving the two ends open. *Keypress and Timing*: K+ waits up to 4 sec for a keypress. Then either the first, second, or third command down is executed, depending on whether no response occurred during that time, a response occurred but didn't match a special keypress variable, or the response did match. Tnnnn times a program delay for nnnn msec, and records all responses and RTs during that time. Wnnnn waits for nnnn msec, and is often used to determine audio or visual display durations. *Arithmetic and Logic*: This group contains three subclasses: jump, counter, and arithmetic commands. For example, Jnnnn is an unconditional jump to line nnnn in the program. Cnnnn increments a counter by 1 each time it is executed. When the counter equals nnnn, the next command is skipped, otherwise program sequencing continues in order. Contents of the eight arithmetic registers can be negated, added together, transferred between each other, compared to each other, set from the program,

BUFFER	B B*	READS PUNCHED TAPE OF DISPLAY CONTENTS & VARIABLE VALUES. PUNCHES ONTO TAPE S's DATA, AND VARIABLE VALUES.
DISPLAY	Dnnnn D	DISPLAYS ON THE SCOPE nnnn BLOCKS FROM THE TEXT BUFFER. DISPLAYS S's KEYPRESS RESPONSES.
INTERFACE	Innnn	SETS THE RELAY BUFFER. 3776 OCTAL = 011 111 111 110 BINARY & WOULD CLOSE THE MIDDLE RELAYS, LEAVING THE ENDS OPEN.
KEYPRESS & TIMING	K+ Tnnnn Wnnnn	WAITS FOR S's KEYPRESS & BRANCHES BASED ON ITS VALUE. TIMES nnnn MSEC & RECORDS RESPONSES & RTs DURING THAT PERIOD. WAITS nnnn MSEC; USED FOR CONTROLLING STIMULUS DURATIONS.
ARITH. & LOGIC	Jnnnn Cnnnn AN: mmm	JUMPS TO LINE nnnn OF THE PROGRAM. INDEXES A COUNTER & SKIPS WHEN IT REACHES THE LIMIT. nnnn. SETS N-TH ARITHMETIC REGISTER TO THE NUMBER mmm.

Figure 3. Some examples of SIMPLE commands.

Figure 4. The equipment (top) and library tape (center) needed to prepare a spoken stimulus tape (bottom).

from the text buffer, or from the subject's keypress array. For example, An:mmmm sets the n-th arithmetic register to the number mmmm.

SAMPLE SIMPLE PROGRAMS

We will illustrate three ways in which SIMPLE commands can be used to control auditory events: to generate tape recordings of spoken stimuli, to conduct a bisensory recognition memory experiment, and to do temporal calibrations of spoken responses.

SAMPLE Preparation of Spoken Stimuli

Suppose that we want precision control of word durations, pauses between words, and hence, overall stimulus rate. To prepare temporally well-controlled tape recordings, we must first prepare a *library tape* and then use computer control to rerecord words one at a time onto a *stimulus tape*. The top of Figure 4 shows the equipment needed to generate the stimulus tapes: a minicomputer, its relay buffer and sense switch register, a voice key, and two remotely controllable tape recorders, for the library and stimulus tapes. First, consider the audio circuits. The speech channel of the library tape recorder is gated via Relay 4, to the stimulus tape-recorder input. The tone-pulse channel of the library tape recorder goes to the voice-key input, which in turn signals Computer Sense Switch 2. Three additional computer relays are needed to operate the tape recorders. Under program control, Relay 8 sends the stimulus recorder in and out of record mode; Relay 2 sends the library recorder in and out of play mode; and Relay 1 rewinds the library tape until the end-of-tape sensor signals Sense Switch 1.

Now let us assume that the library tape in the middle of the figure will contain 11 stimulus items: a warning tone plus 10 digits. First, a highly intelligible

COMMANDS	COMMENTS	
A0:0	INITIALIZE	INITIALIZE STIMULUS COUNTER
B		READ TEXT TAPE WITH ITEM CODES
C*		INITIALIZE LIBRARY-TAPE COUNTER
RC		SET LIBRARY COUNTER TO ITEM CODE
T		START CLOCK; ENABLE SENSE SWITCHES
I0002	POSITION LIB.TAPE	ACTIVATE RELAYS; START LIB.TP.RECORDER
T+		WAIT FOR TONE PULSE AT SENSE SW.2
T*		TURN OFF CLOCK; DISABLE SENSE SWITCHES
C/		PROPER POSITION ON THE LIBRARY TAPE?
J(-5)		NO, GO BACK AND WAIT FOR NEXT TONE
W0400	RECORD STIM.TAPE	YES, WAIT BEFORE STARTING STIMULUS TAPE
I0016		START STIMULUS TAPE; LIBRARY TP. CONTINUES
W0200		RECORD DIGIT OR WARNING TONE
I0012		GATE OUT LIBRARY TAPE AUDIO OUTPUT
W0300		RECORD INTER-STIMULUS INTERVAL
I0000	REWIND	STOP BOTH RECORDERS (ALL RELAYS OPEN)
I0001		REWIND LIBRARY TAPE
T		START CLOCK; ENABLE SENSE SWITCHES
T+		WAIT FOR END-OF-TAPE SIGNAL AT SENSE SW.1
T*		TURN OFF CLOCK; DISABLE SENSE SWITCHES
I0000	END-TEST	STOP REWIND
A+:1		ADD 1 TO STIMULUS COUNTER
A/:8		CHECK IF LAST STIMULUS IN TRIAL
J2		NO, DO ANOTHER STIMULUS
F		YES, FINISHED

TEXT TAPE: 1/\$ 8/\$ 2/\$ 7/\$ 3/\$ 9/\$ 6/\$ 5/\$
RELAYS: 1=REWIND; 2=PLAY; 4=SPEECH GATE; 8=RECORD

Figure 5. A SIMPLE program to record a warning tone and one seven-digit trial of a stimulus tape.

speaker would record each digit 10 to 50 times on a *prelibrary* tape. Then we would select one example of each digit, so as to equate intensity and duration for the total set. In our example, each digit is 200 msec. We might slightly clip the digit onsets and offsets to provide sharp transients. Peak or average intensity can be measured with a voltmeter, and digit duration can be calibrated on the computer, as we will discuss later. The final *library tape* will have a speech channel and a tone-pulse channel to signal the computer to prepare for each word. Hence, the time between the onset of each tone pulse and its associated word must be constant, 400 msec in our example. Further, the ordinal position of each tone pulse will serve as the item code for the following word. The item codes, shown on the figure, are used by our program to locate the specific elements to be rerecorded from the library tape. For example, an item code of "1" indicates that the warning tone is called for; an item code of "5" indicates a "3" on the library tape.

For our present example, the final stimulus tape will contain 100 trials, each having a warning tone followed by a sequence of seven digits. On the stimulus tape, the tone and digits will each last 200 msec and the interstimulus interval will be 300 msec, yielding a 2 digit/sec stimulus rate. To produce the final stimulus tape, we will rerecord items from the library tape one at a time. This procedure insures that all occurrences of a given item will be acoustically identical, regardless of presentation rate, serial position, or trial number.

Figure 5 displays a SIMPLE program to record a warning tone and one seven-digit trial of our stimulus tape. For 100 trials, we would just add an additional

counter loop. To simplify this program for the purposes of illustration, we have assumed that all pieces of equipment (e.g., tape recorders) have activate and deactivate times of 0 sec. In reality, these times would be measured, and slight corrections would be made in the values of the programmed Wait statements (Wnnnn). The left column on the figure contains the actual SIMPLE commands. The center and right columns contain comments so that we can more easily interpret the program for the present discussion.

The program for preparing stimulus tapes can be divided into five parts:

(1) *Initialization.* The A0:0 command sets the A0 register to zero. This register will be used as a counter to determine how many stimulus items have been recorded. The B command reads the paper text tape with the item codes into the computer. C* initializes a counter which will be used to determine the location of the appropriate stimulus item on the library tape. RC sets the upper limit of the counter related to the C* command, based on the item codes from the text tape. T enables the computer to pick up impulses from the sense switches. In this case, the sense switches are connected to a voice key.

(2) *Library-Tape Positioning.* The next set of commands runs the library tape recorder until the proper stimulus is found. I0002 sets the computer relays to start the library tape recorder and to complete its audio output circuit. T+ halts the program and waits for a tone from the pulse channel of the library tape, via the voice key. T* disables the sense switches once a tone has come through. Then C/ counts the number of tones received from the pulse channel of the library tape. When the number of tones equals the item code, previously set by RC, the next command is skipped. If it is not yet equal, a J(-5) is executed which goes back to T and waits for another tone.

(3) *Stimulus tape recording.* The tone-pulse for the proper library digit or warning tone has now been located. The program waits 400 msec (W0400) for the library tape recorder to reach the onset of the associated digit or warning tone, connects the speech channel, and starts the stimulus tape recorder by closing the appropriate relays (I0016). The stimulus item is recorded for 200 msec (W0200), and then the audio connection between the two tape recorders is broken (I0012). A silent section of tape is recorded for 300 msec (W0300) to provide the interstimulus interval, and both recorders are then stopped (I0000).

(4) *Rewind.* The I0001 command closes the relay which rewinds the library tape. The end-of-tape sensor on the library tape recorder is connected to the sense switches and enabled by T. The T+ waits for an end-of-tape signal and I0000 stops the rewind.

(5) *End test.* The A+:1 command adds 1 to register A0 and A/ checks to see if A0 equals 8. If not, we must record more stimulus digits, and we start

again by jumping to the second line of the program (J2). If the warning tone and seven digits have all been recorded (A0=8), the program halts (F).

With minor modifications, this type of program could easily be used to record dichotic tapes with word pairs on the two channels closely synchronized. The first channel would be recorded as we have just described, except that the subject's warning tone preceding each digit series would initially be recorded across both tape channels. Then this tone could be used as a signal to the computer for synchronizing both channels when the channel No. 2 digits are recorded on a second pass.

SIMPLE Control of an Audio-Visual Experiment

Figure 6 contains a SIMPLE program for a bisensory recognition memory experiment. As shown on the top of the figure, the subject first listens to the spoken stimulus tape that we have just discussed. After each trial, a test digit will be visually displayed for a YES/NO recognition response. We might be interested in the subject's RTs or errors as a function of serial position, presentation rate, time between stimulus and test items, or S/N ratio of the stimuli. Let us now discuss the command structure for this experiment. Note the ease with which a tape recorder and a tone generator can be controlled, and the ease with which auditory and visual displays are intermixed within a trial. Again, this program is for a single trial, and a counter-loop must be set up to cycle around for a 100-trial session.

Figure 6 shows a single trial of this experiment.

Figure 6. The time course of events in a recognition memory experiment (top). A SIMPLE program to run that recognition memory experiment (bottom).

Figure 7. The equipment, the course of events, and a SIMPLE program for obtaining temporal calibrations of auditory stimulus tapes.

First, the B command reads the text tape into the input buffer. Then the D command displays the word "Ready" on the scope until the K+ command senses the subject's keypress to begin the trial. Next, the I and W commands hold closed a computer relay for 4 sec while the tape-recorded stimulus sequence is presented. Following the auditory sequence, RK reads into the keypress array the character corresponding to a correct recognition response, in this case the letter Y. Then the D1 command displays the visual recognition test item, a 7 in this case. The K+ command then waits for the subject's keypress. If the subject's keypress does not match the preset keypress variable, the second statement after the K+ is executed; if it does match, the third statement down is executed. Thus, if the subject responds in error, the feedback tone is skipped, but if he is correct, a tone is activated by a computer relay (I0002) for 100 msec (W0100). Finally, the last two commands output on the punched tape the subject's response and reaction time, and terminate the program.

SIMPLE Time Measurements of Spoken Stimuli or Responses

A SIMPLE means of time calibration for auditory stimulus tapes is valuable, if not absolutely critical, to many types of research on speech perception and memory. For example, the time available during or between spoken stimuli for "perceptual coding" or for "memory decay" may be important to the theoretical

conclusions of the research. Also, there is a need for SIMPLE response calibration procedures. Recent research on memory retrieval and speech production places a heavy theoretical emphasis on temporal factors in cognitive processing. The subject's RTs are taken to index cognitive processing time. Much current research deals with clusters or chunks in the recall of word lists, or with phrase structure in sentence production. However, very few studies report response times for fluent speech containing more than a single word! This is because such temporal measurements are tedious without automated equipment.

Figure 7 presents an exceedingly short and SIMPLE program to obtain temporal data for stimulus or response tape recordings of spoken sequences. On the top of the figure is an equipment schematic. The tape-recorded stimuli are played into a voice key. From the voice-key relay, the common and normally open contacts are wired to Computer Sense Switch 1, and the common and normally closed are wired to Sense Switch 2. Our particular sense switches are designed to signal the computer only on the leading edge of an input. Thus, for a sequence of three words, the signal pattern in the middle of the figure would result. The onset of Word 1 would close the voice key and trigger Sense Switch 1. When Word 1 is finished, the voice key would open and trigger Sense Switch 2. This would yield the ON TIME for Word 1. The onset of Word 2 would again close the voice key and trigger Sense Switch 1. The time delay between the offset of Word 1 and the onset of Word 2 is registered by the computer, yielding the appropriate OFF TIME between Words 1 and 2. This pattern could continue as long as we desired. Now, let us look at the bottom of this figure to consider the program commands needed to obtain these time data.

In this program, C* clears the counter which will later indicate the number of pulses to be received. The T commands wait for and record the latency of the pulses. The C6 command counts six pulses, i.e., three spoken words, and decides when to output the time data with a B* command and terminate the program (F).

This is but one type of program for obtaining speech ON and OFF times. In this case, we expect exactly three words on every trial, as might be the case for calibrating a particular stimulus tape. A more general program would let the researcher type into a variable register the number of words expected on each trial. Another possibility is to let the program loop around until the researcher presses a key. This is useful when there is no a priori way to determine the number of speech stimuli. The researcher might signal the computer when the subject's response tape recording has been silent for some prolonged interval or when the subject responds "FINISHED."

In summary, we have provided an overview of a very flexible, general purpose programming language for the cognitive psychology lab. Although this language was designed around the visual-display scope, we have seen how it can be easily adapted to measuring and controlling auditory events. Hopefully, some ideas have been provided concerning the manner in which other programming languages might be similarly adapted. We have illustrated three important ways in which such auditory control can be helpful in cognitive research: It can be used to prepare spoken stimulus tapes with precision control over the temporal parameters; it can be used to control on-line interactive audio and bisensory experiments; and finally, it can be used to measure the temporal parameters of spoken stimuli and responses. Indeed, experimental methodology in auditory memory and psycholinguistics could be markedly improved by the more frequent use of minicomputers in these ways.

REFERENCES

- AARONSON, D., & BRAUTH, S. SIMPLE guidelines for developing a computer-based laboratory. *Behavior Research Methods & Instrumentation*, 1972, 4, 257-264.
- BRAUTH, S., & AARONSON, D. SIMPLE: Scope Interpreter for Memory, Perception, and Learning Experiments. *Decuscope*, 1972, 11, 2-7.
- HUTTON, C. L. *A psychophysical study of speech rate*. Doctoral thesis, University of Illinois, 1954.