A brief history and comparison of several variants of the ACT (automated contingency translator) language

C. T. MICKEY

BRS/LVE, 5301 Holland Drive, Beltsville, Maryland 20705

This paper presents a brief history of the automated contingency translator (the ACT language). It will also present a rather detailed comparison of ACT-N to ACT III.

The commercial rights to ACT I for the Digital Equipment Corporation PDP-8 family of computers was purchased by Lehigh Valley Electronics from the originator, Dr. Jock Millenson, about 1968. The language consisted of a Compiler and a separate Executive and was designed to work on a 4K machine; this language, as such, was never marketed. Members of Lehigh Valley Electronics wrote an Interpreter for the original NOVA to permit code generated by the Compiler in the PDP-8 to be entered and executed on the NOVA. This system was displayed at trade shows in 1969, but was never marketed to end users.

Lehigh Valley Electronics then rewrote ACT I, combining the Executive and Compiler simultaneously in core and adding several important features such as Stores and Command Mode. This language variant became known as ACT II, and several systems were sold to end users. Several of these systems are still in use. ACT II was written for 8K of core on the PDP-8 system. Other relevant changes made at that time were: (1) Command letters or codes were changed from a completely arbitrary letter to the first letter of a word that could logically describe the action to be taken such as CTRL A for access rather than CTRL E, CTRL K for kill rather than CTRL V, CTRL R for run or restart; CTRL H for hold or halt and CTRL C for clear. (2) Special characters were added to permit a tape of the program being compiled to be reused for subsequent recompilation. This program is somewhat different in ACT II than in ACT III because the PDP-8 does not have interrupt masks for each class of device as does the NOVA. (3) The space character was permitted but not required anywhere.

Almost simultaneously with this further development, the language was written for the Data General Corporation NOVA series of machines incorporating all of the then existing features of ACT II and making additional changes to further enhance the language. Variables were increased from 8 to 64, and the third type of transition was changed from one based on the count of state exits to one based on the comparison

of variables and constants. Other changes have been made in ACT III over the years to further enhance its usefulness. These changes include: (1) servicing the power monitor system, (2) including all possible characters in literal print statements, (3) adding control characters to permit compiling from a high-speed paper-tape reader and outputting data to high-speed paper-tape punches or alternate teleprinters, (4) outputting to an internal buffer, (5) transferring the value of the latest response into a variable, and (6) transitions based on analog values of response.

Simultaneously with these developments, Dr. Millenson rewrote a variant of ACT I for the PDP-9 computer series. The author has no knowledge of this variant other than its existence.

Subsequent to this variant, Dr. Millenson created still another version of the language for the NOVA known as ACT-N. A number of these systems have been marketed in Europe. A modification of this version will be marketed by BRS/LVE as ACT-NIII.

There are only minor differences in the conception of the procedure and subsequent entry into the computer between ACT III and ACT-N. These differences are in regard to operations with variables, outputting of data, naming states, and specifying stimulus values.

In ACT III there are only two phases to the program. The Procedure Phase, which includes manipulation of variables and data recording statements, and the Record Phase, that specifies only which data is to be recorded.

In ACT-N there are three major phases: (1) the Procedure Phase, in which only the states and transitions are specified, (2) a Basic Phase, in which operations with variables and print statements are specified, and (3) a Record Phase, in which the data to be recorded are identified.

ACT III deals only with integer variables, and there are 64 of them available (eight letters each having eight postscripts). For modification of variables, there are two arithmetic operators (+ and -). For transitions based on variables, there are six relational operators (>, <, >=, <=, >=, <=).

<=, =, and <>) and two arithmetic operators (+ and -).

ACT-N has 15 integer variables (A through O), two string variables (U and V), with a maximum length of 127 each, and four real variables (W, X, Y, and Z). For modification of variables, there are five arithmetic operators (+, -, X, /, and SQRT), and for transitions based on variables, there are three relational operators (>, <, and =). Computed values are not allowed in transition statements. Not all operators are permitted with all classes of variables, but that could constitute a 15-min talk in itself.

In ACT III, the record section can only be used for specifying what data is to be recorded. The data identified, such as response counts and time durations, cannot then be moved to variables for effecting transitions or for subsequent calculations.

In ACT-N, the record section is treated as a class of variable, known as Q variables, and data so specified can be moved unidirectionally into real or integer variables for subsequent calculations, for effecting transitions, and for data recording. In addition to response counts and state durations, count of any given state entrance can be specified.

ACT III time transitions may be an expression including hours, minutes, seconds, and units.

ACT-N time transitions must be expressed in terms of a single identifier: either minutes, seconds, or units, but none of these combined.

ACT III permits one nonprobabilistic time transition per program state and any number of probabilistic time transitions per program state.

ACT-N permits only one time transition per program state; if probabilistic, there may be up to four probability statements.

In ACT III, the order of servicing transitions is: variables first, time second, and response third.

In ACT-N, the order of servicing transitions is variables or response first, time second.

In ACT III, nesting is allowed to a depth of four in three planes.

In ACT-N, nesting is allowed to a depth of seven in six planes.

In ACT III, it is possible to undo the current line or the entire program compiled to that point.

In ACT-N, in addition to the two listed for ACT III, a CTRL W will undo the current plane.

In ACT III, states are named with a plane identifier (U, V, and S) followed by an octal number indicating the stimulus value. A period is used as a separator of the stimulus value and an octal number from 0 to 77. This number permits 64 states, having the same stimulus value in each plane.

In ACT-N, states are named sequentially using a single letter and a decimal number from 1 through 127. An octal number separated by a space defines the stimulus value.

Other features added in ACT-N that have not previously existed in any other variant are: (1) The capability of specifying in each print statement which device is to receive that data. This currently includes the console teleprinter, an alternate teleprinter, or a printer such as a Versatec or T1733 KSR/ASR, and a high-speed punch or similar device. (2) Mathematical manipulations of four real variables including multiplication, division, and square root as well as addition and subtraction. (3) Use of 6 planes per station and 62 stations. These are language limits, but the hardware limit is still eight stations. Computer update time may also be closer to eight stations with a 100-Hz clock. (4) Operations with string variables for performing pseudorandom schedules, printing of histograms, or similar applications. A dimension statement associated with string variables is also added. (5) Two teleprinters or consoles may be utilized for accessing the Compiler or Command Mode. (6) In time transitions, the letter T is recognized as an acceptable alternate to the letter A.

THE I/O STRUCTURES OF ACT III AND ACT-N

Interrupts

In ACT III, input-output (I/O) is handled completely by flag checks, where all devices so equipped are maintained in the masked out condition. The only devices expected to interrupt, therefore, are the power monitor and the ACT Clock. One of the constraints this places on the system is that all input or output devices must be capable of asynchronous operation from zero to their maximum frequency.

Early systems that did not service power monitor maintained the interrupts off throughout the executive routine. A patch tape permitted modification of the language to service the power monitor, which requires the Executive to run with the interrupts enabled. All systems currently being delivered include power monitor in the software.

The interrupt service routine saves all accumulators, carry, and location zero, it checks the power monitor flag (halting if set), and then it checks the "ACT Clock" flag. If neither flag has been raised, it indicates an unauthorized interrupt such as noise or an unknown device, which causes E-1 to be printed on the console teleprinter with the interrupts turned off. Following this condition, the system timing is synchronized with the beginning of the next ACT Clock tick, and processing is then continued from where it was interrupted. If the clock tick occurs while servicing the executive routine, it indicates that update time was exceeded which causes the E-O message to be queued, for printing with highest priority in back-

ground, and processing is continued from where it was interrupted. If the clock tick occurs in background, the necessary information is saved and the Executive routine is begun. If no interrupt of the Executive routine occurs, the processor restores the previously saved information and returns to Background. The power monitor may interrupt in Background or Executive. In either case, the necessary information is saved and the processor is halted. Upon restoration of power, the saved information is restored, timing is synchronized with the next ACT Clock tick, and processing is picked up where it left off.

In ACT-N, interrupts are handled much as described under ACT III, as far as the clock and power monitor are concerned. However, certain higher speed I/O devices are allowed to interrupt the Executive. Slow-speed devices, namely teletype-writers, are only allowed to interrupt in background. When an I/O device interrupts, a character is transferred between the device and its buffer with the interrupts turned "off." This should not present a problem since service time for each peripheral is significantly less than 100 microsec. However, it can add the service time of some I/O devices to the Executive timing, thereby making it more likely to get an update-time-exceeded message. In ACT-N this occurrence has the disastrous result of killing the station program being updated at that moment.

ACT-NIII allows all I/O devices to interrupt at any time, and services them as described above. However, in the event of an E-O, the station being updated is simply removed from the run list and placed on a hold list. From there, it can be restarted, or, if it is found to have an excessively long or infinite Basic loop, it can be killed.

This manner of handling I/O devices has the major advantage of maintaining an I/O device running at its maximum or slow rate. This is extremely important in the case of such devices as magnetic tape or disk, as well as some high-speed paper-tape readers/punches.

Background

In ACT III, background circulates through several routines, namely Monitor, which echoes all keys struck on the keyboard or sent via the teleprinter reader and looks for special control characters (A, B, D, E, and P). Monitor calls Readin, which first checks for operating errors, E-O through E-4, then reads a character, ignoring Escape and all characters bracketed by Escapes. If Readin is unable to obtain a character because the input device is busy or idled, it calls Datalook which checks for data pointers and processes data output. When this program is unable to output a character, because the device is busy or idled, it returns to Readin. When

a control character is encountered, monitor transfers control to the appropriate program, such as Command or Compile. These programs, while they are active, call Readin to obtain their characters, and when finished, return to Monitor. Thus, operating errors are sensed each time a Readin is attempted and take control of the console printer in precedence to any other program which may be using it. Program input has next priority on a single teleprinter system because data output is suppressed while compiling, and data type has the lowest priority (or, on multiple output device systems, equal priority with the compiler).

A flag is raised each time Readin obtains a character from an input device. This flag and a message flag controlled by timeout is used to determine when it is appropriate to output a special character (Escape on some systems and CTRL F on others). When this special character is sensed in Readin, all subsequent characters until the next occurrence of the special character are ignored. This routine permits the user to generate a tape on-line of a program being compiled for later recompilation from a tape reader. A CTRL B causes a modification of the Readin program such that in addition to checking the teleprinter input flag, it also checks the input flag of an alternate device (normally a paper-tape reader) and suppresses timeout to the console teleprinter. Upon sensing an \$ in the Record Phase, the suppression program is removed. Upon sensing a CTRL E in Readin, the Readin program is restored to the original condition of looking only to the teleprinter for input. Changes were made early in 1975 concerning when these control characters could be sensed.

Timeout for the alternate I/O devices does not produce the special character since all output to these devices is assumed to be unidirectional, i.e., from the processor to the device. Input is not accepted from any of these devices except via the CTRL B operation.

ACT-N Background

In ACT-N, all keys struck on either of two keyboards are ignored until a CTRL E is struck. When a CTRL E is typed, that teleprinter is masked off the interrupt system, and a Reading program is entered. This program reads characters via flag checks and determines whether the user is attempting to access a busy station or a free station. If a busy station, conversational mode is entered where variables can be queried or changed, the station can be killed, or the station can be restarted. If CTRL E is struck on the alternate keyboard, while this program is active, WAIT will be sent to the alternate printer. If a free station is accessed, the compiler

is entered. While in the Compiler, the alternate keyboard may be used in conversational mode. If an attempt is made to access a free station from the alternate keyboard, the message TRY LATER will be sent to the alternate printer. When the compiler is entered, an IN code is requested. This IN code specifies what device will furnish compiler input, and whether or not typing this input will be suppressed. If either keyboard is specified, characters are read via flag checks as required in the Compiler. If a high-speed paper-tape reader (PTR) is specified, characters are read via the interrupt system into a buffer, and the compiler takes its input from the buffer. Tapes used in the PTR may not have characters that would cause a Compiler error, or any extra characters not required by the Compiler (with the exception of ignored codes such as nulls, rubouts, and line feeds). When Reading or the Compiler are held up for want of input characters, a data-type program is entered. This program places all output characters into the specified buffer and enables the appropriate output device. When this program is finished or held up, the Kill program is entered. This program checks a Kill list, performs the Kill function when so directed, and returns to Readin. Programs are automatically started as soon as they are compiled and cannot be stopped except via Kill.

ACT-NIII Background

ACT-NIII reads both keyboards via flag checks and echoes all keys struck until a CTRL A occurs. It then determines whether a free or busy station is being accessed. If busy, Command Mode is entered. While in Command Mode, WAIT is sent to the alternate printer for all keys struck on that keyboard. Command Mode (Conversational Mode) performs the same functions as ACT-N, except that CTRL C (clear) and CTRL H (halt) have been added.

When a free station is accessed, an IN code is requested. The IN code, as in ACT-N, specifies the input device and suppression of timeout if desired. All input is taken via the interrupt system into a buffer, and the Compiler always takes its input from that buffer, ignoring Escape and all characters bracketed by Escapes. While compiling, the Escape (CTRL F) code is generated similar to ACT III. However, it differs in that the tape is neither device- nor station-specific. Recording should not be started until the first statement of the Procedure Phase. All subsequent input should then be recorded. The other background functions are identical to ACT-N.

In ACT III, if systems are equipped with nothing but a single teleprinter for all input and output (the condition assumed by the ACT III language tape furnished with new systems), a compiler flag is checked and no data typing is permitted while this flag is set. Patch tapes are provided to modify the language to or from single teleprinter and multiple I/O device systems. On a multiple device system, output can be directed to any device but the console teleprinter.

In ACT-N and ACT-NIII, single teletypewriter systems, data is sent to the console printer, if so directed by a print statement. If compiling is being performed with inhibit, this does not present any problem. However, if compiling from the keyboard without inhibiting, the data will be injected between input characters (with the Escape character placed appropriately). This could increase appreciably the length of tape to be used for recompilation. Output specified for any device not being used currently in the Compiler will not have Escape characters generated.

In the ACT III multiple I/O device systems, timeout to the alternate devices occurs sequentially and is determined by the CTRL D operation. Each character is sent first to the highest speed device (an internal buffer), if that device was selected. The same character is then sent to the next-highest priority device, if selected, and finally to the lowest priority device, if it was selected. Only then is the next output character obtained by the background program.

In ACT-N and ACT-NIII, any single output device is specified in each print statement.

In ACT III, the data buffer is an area in memory which the Data-Type program treats as a peripheral device. Data, once entered into the buffer may be retrieved by outputting the entire data buffer to its I/O device, under program control (TAPE), via control of the console (CTRL T), or by virtue of the fact that it has become full (automatic CTRL T). The buffer is packed two ASCII characters/word and can be readily extended to any desired length. The output driver program assumes that it will be working with a teleprinter kind of interface, and thus unpacks the buffer, sending it one character at a time to device code 51. This also could be readily changed to send the data by word over the data channel to a bulk storage device. The simplest method of accomplishing this would be to branch from ACT to the Output Driver program, ignoring any clock ticks that may occur during this normally very short period of time. The alternative to this is to modify the

Interrupt Service routine of ACT III to keep the bulk storage device masked off the interrupt line until the buffer is full or ready to be dumped. This program would be relatively easy to write, and sufficient portions of the listings are available for any user to perform this function for themselves.

The Stores routine permits a user to save in core, data specified in the Record Phase that may recur at higher rates than their output device can accept them. Each store command places one piece of information in a four-word block of memory along with a station and plane identifier and a link to other information on this list. The major advantage of this routine is that the storage takes place in the Executive and is, for practical purposes, instantaneous. The major disadvantages are: (1) It is rather inefficient use of core. (2) The output is very inflexible. (3) Literals cannot be stored.

In ACT-N, these data may be saved in string variables. The major advantages of this method are: (1) the most economical use of core, (2) unrestricted output formatting, (3) practically instantaneously occurrence, (4) data can be nondestructively retrieved, (5) the data can be retrieved for computation and/or program control as well as for output.

The major disadvantages are: (1) The maximum length of the list is 127 pieces of information with two such lists available per station, provided none of the string variables are required in the Procedure Phase. (2) Literals cannot be put into string variables.

In ACT-N and ACT-NIII, the space character is looked for as the terminator of noise words, and thus is a required character in some illogical places. Also, it is not ignored and may cause Compiler error codes if used in other places.