DOCUMENT RESUME

ED 108 601 IR 002 112

AUTHOR Follettie, Joseph F.
TITLE Contingent Instructional Advance: Implications for IDCMS.
INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO SWRL-TN-1-72-05
PUB DATE 23 Feb 72
NOTE 17p.

EDRS PRICE MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS *Branching; Computer Assisted Instruction; Computer Graphics; *Computer Programs; Computer Storage Devices; *Information Processing; Instructional Design; Instructional Improvement; Instructional Systems; *Instructional Technology; *Interaction; On Line Systems; Optional Branching; *Research Design; Research Tools
IDENTIFIERS IDCMS

ABSTRACT

An illustration of modestly-interactive instruction of moderate length--to be administered to six students who start off together--is presented. Implications for IDCMS regarding number of audio programs and video files and extent of looping are preliminarily deduced on the basis of characteristics of the illustrative instructional program. (Author/SK)

ED108601

SOUTHWEST REGIONAL LABORATORY
TECHNICAL NOTE

DATE: February 23, 1972

NO: TN 1-72-05

CONTINGENT INSTRUCTIONAL ADVANCE: IMPLICATIONS FOR IDCMS

Joseph F. Follettie

ABSTRACT

An illustration of modestly-interactive instruction of moderate length--to be administered to six students who start off together--is presented. Implications for IDCMS regarding number of audio programs and video files and extent of looping are preliminarily deduced on the basis of characteristics of the illustrative instructional program.

CONTINGENT INSTRUCTIONAL ADVANCE: IMPLICATIONS FOR IDCMS

Joseph F. Follettie

Student-system interactive instruction or research requires decisions to advance S or to loop off the mainline contingent upon some or all of the information reflected in prior performance. The SWRL Instructional Development Control and Monitoring System (IDCMS) in time will be appreciably used to establish characteristics of effective-efficient interactive instruction. The design of IDCMS in Version 1 hardware configuration now is appreciably completed. The design was premised on a general view concerning what the system will be asked to do and cost considerations. Developing views on how the system in Version 1 form desirably will be used now make entertainable the proposition that slight modifications in hardware design may be warranted. Although perhaps a less imperative matter, parallel comments seem applicable to software to be provided by the contractor. The particular slight modifications in the contracted system that may be warranted can only be identified and evaluated in consequence of increasing specificity regarding desired general system functions. Among system functions, the interactive one presently invites closest scrutiny; many research functions are subsumed by the system's interactive function. The scenario to be presented illustrates an interactive structure in sufficient detail to permit qualified staff to evaluate the contracted IDCMS configuration against functions inherent in the interactive structure. Such an approach is only as useful as its illustrative argument is compelling. Hence, the possible executive decisions that system "deficiencies," so defined, invite are: a) If costs are acceptable, modify contracted design for the system to achieve illustrative functions. b) Modify the interactive requirement to correspond to the contracted system. A compromise decision falling between these alternatives also is possible.

A <u>mainline</u> instructional program is one that S will negotiate to a program-defined exit if the instructional control decision at each decision point falling along the mainline is positive (+). That is, if at each point tested S reveals a criterion level of proficiency for the program skill(s) (PS) tested, then he will advance along the mainline to a predetermined succeeding lesson until, at last, he exits through a terminal decision point for the program. However, whenever his performance warrants a negative (-) instructional control decision, S will <u>loop</u> off of the mainline. That is, S will loop off of the mainline wherever a test reveals achievement of less than a criterion level of proficiency for tested program skill(s).

If a test warrants a negative instructional control decision--that is, reveals subcriterion proficiency--then two major causes for this decision can be discerned: a) prior relevant mainline instruction or the conditions of its administration--e.g., pacing--are ineffective for this particular S or b) prerequisite skills not taught on the mainline are deficient. Two sorts of prerequisite skills--distinguished on the basis of the point in the instructional progression wherein their proficient employment is required--can be discerned: a) entry skills (ES), whose proficient employment is required during earliest lessons of the instructional program and b) enroute prerequisite skills (MS), whose proficient employment is required at later intermediate points in the instructional sequence. The point at which such skills become relevant would not be particularly important as a basis for distinguishing the two sorts of prerequisite skills if skills of both sorts became relevant one at a time. However, it tends to be true that several prerequisite skills become relevant at the outset of instruction, whereas prerequisite skills that become relevant later in instruction do so one at a time. Illustrative interactive instruction initially will make use of this difference in temporal concentration of entry and enroute prerequisite skills. However, later remarks will place ES and MS in the same set (EMS).

A fundamental assumption underlying the structure to be illustrated is that student-system interaction should reference to moderately extensive instruction under the condition of multiple usership (e.g., 6 Ss) Some of the problems we will encounter disappear or are considerably lessened if instructional extent is taken as one-third of what we will show. When elsewhere we examine time-pressured retrieval-transmission from stores containing much larger numbers of "entities," such a lessening of extent will bound the effort.

Moderately extensive instruction is here normatively defined as 15 30-minute lessons, such that an S whose rate of acquisition (referenced to the illustrative instruction) is average will complete the program by working 25 minutes per day on normative instruction and supplementing instruction as needed. Figure 1 provides a diagrammatic view of the illustrative instruction. Open circles reflect mainline instruction; closed circles, the (identified) prerequisite skills. (A foreseeable outcome of interactive research is that it will invite +H hypotheses concerning relevance of formerly identified prerequisite skills.)

Level 1 lessons of Figure 1 may be interpreted either as addressing single program skills without subordinate skills structure or as addressing a program skill that is superordinate to a set of subskills. The former view yields the simplest possible illustrative program. Since the program already has been extensively oversimplified, the second interpretation is used here. We characterize Level 1 lessons of Figure 1 as reflecting a program skill that integrates two subskills previously taught in the same lesson.[1]

---

[1]Particularly where rule learning and generalization are required--as in phonics instruction--Figure 1 dramatically oversimplifies the situation by failing to reflect buildup of the rule set over lessons.

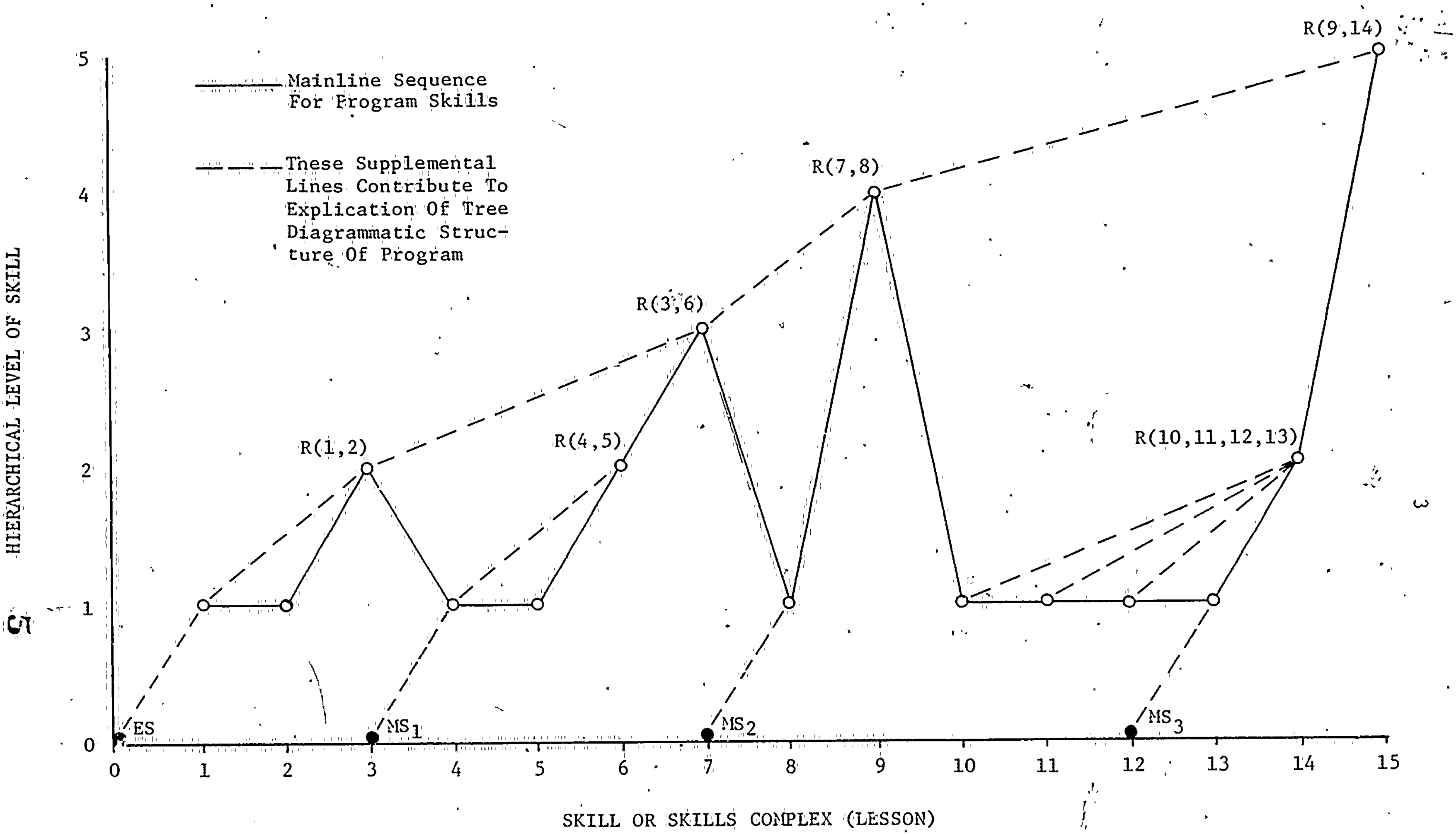

Figure 1. An Illustrative 15-Lesson Program, Showing Sequence, Skills - Integrative Structure (R = Integration of Skills), Entry Skills (ES), and Enroute Prerequisite Skills.

Entry Skills (ES)

Any instructional program is predicated on one or more entry skills. These skills are required during earliest lessons of the instructional program. However, they are taught only if evaluation reveals that S performs below criterion proficiency level for them. It is typical to class as entry skills only those skills for which it can be assumed that at least half of the entering Ss will be criterion proficient when tested at the outset of instruction. Earliest lessons of given instruction typically will require S to be criterion proficient in several such skills. Because so many entry skills require evaluation at the outset of instruction, we assume that an initial test, consisting of a perfunctory subtest for each entry skill, will either clear S for initial instruction or will yield one or more hypotheses to the effect that E may be deficient in one or more entry skills. Should S be cleared for mainline instruction (+ES), tenability of one or more hypotheses of form H:-ES_i still might be established later in consequence of diagnostic characteristics of the evaluation system. Should a hypothesis H:-ES_i be entertained in consequence of administering the perfunctory entry skills test, then deficiency in ES_i might be further evaluated using a more extensive test EV_1-ES_i (see Figure 2).

We assume that skills analysis referencing to given instruction should be extended downward no further than one level below the level of the entry skill. In consequence, if H:-ES_i is found tenable, then one level of formal instruction culminating at the ES_i level of a proficiency hierarchy can be designed. We denote such instruction IN-ES_i. Effectiveness of such instruction can be tested using a second version of the test for criterion proficiency in ES_i--denoted EV_2-ES_i. Should this test also reveal tenability of H:-ES_i, then one's options would be either to administer versions of IN-ES_i and EV-ES_i as required to raise S to ES_i or to administer informal instruction and conduct informal evaluation culminating in the decision +ES_i. We show the second of these options in Figure 2. Which option would prove most apt probably will depend on the particular characteristics of ES_i and, in research setting, of E and of the system that supports student-system interaction.

Where this is allowed, various system-burdening requirements emerge. First, the system must store a rather large number of novel rule words that permit testing for rule mastery in all applicable intraword positions and in all applicable word contexts. Second, so many alternate diagnostic tests can be imagined that on-line composition of tests becomes required, which may entail complex coding of word items. Were we to admit this complication into the illustration, then the lengthy paper that lies beyond would not be required. The system simply could not handle extensive rule learning and generalization within the context of moderately extensive instruction.

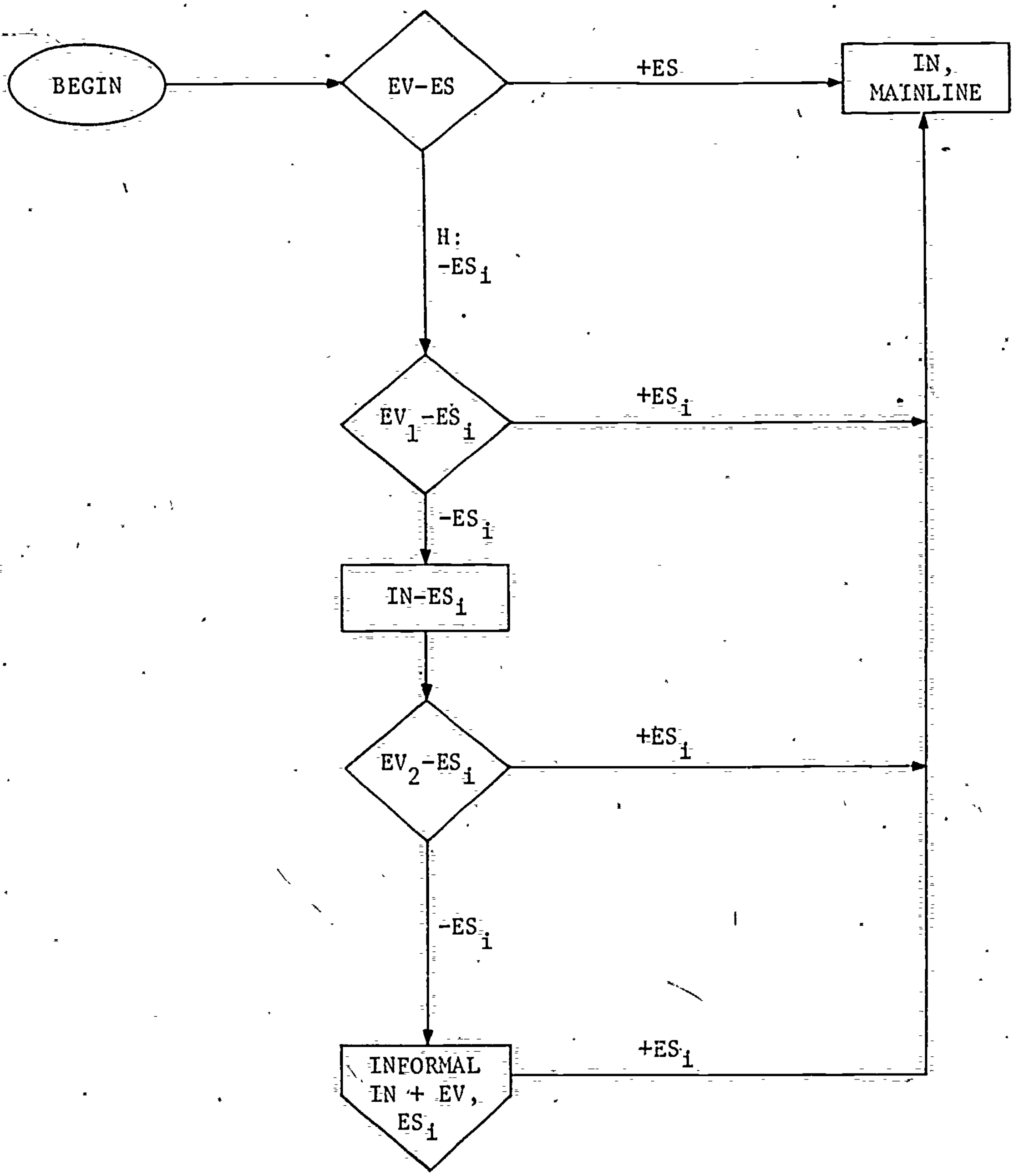

Figure 2. Illustrative Flowchart for Entry Skills (ES) And An ith Entry Skill (ES_i) In Particular For Which S Is Unacceptably Proficient, Treated Interactively. H = Hypothesis, EV = Evaluation, IN = Instruction, Minus Sign = Unacceptable Proficiency Level, Plus Sign = Criterion Proficiency

Enroute Prerequisite Skills (MS)

Enroute prerequisite skills differ from entry skills only with regard to the point in instruction wherein they become germane. These skills--denoted MS--become prerequisite to instruction at intermediate points during instruction, rather than during earliest lessons. Typically, such prerequisite skills will become germane one at a time. That is, unlike entry skills--several of which will prove germane at or very nearly at the outset of instruction--enroute prerequisite skills will warrant consideration singly at different points in instruction. In consequence, it is not necessary to employ an overall perfunctory test--EV-MS--for these skills. Rather, at the point wherein a given such skill MS_i becomes germane, a definitive test EV_1-MS_i may be administered (see Figure 3). Again, one's options are a series of versions of IN-MS_i and EV-MS_i which formally instruct and evaluate S until the condition +MS_i is reached or informal instruction and evaluation culminating in +MS_i after preliminary formal instruction and evaluation. Again, we show the second option in the flowchart.

As with entry skills, categorization of S as +MS_i need not be irrevocable. If later evaluation is made suitably diagnostic, then it remains possible at some later point in instruction again to entertain H:-MS_i.

Program Skills (PS)

The instructional design-development effort assumes that not all Ss will require administration of instruction addressing ES and MS; however, it is assumed that all Ss will require administration of every program skill.[2]

Figure 4--an oversimplification to be corrected in later remarks--begins with first instruction for some program skill PS_i falling on the mainline; this instruction is denoted IN_1-PS_i. The illustration assumes entertainability of three hypotheses: a) H0:+PS_i, b) H_1:-PS_i with cause -MS_i, c) H_2:-PS_i with cause ineffective IN_1-PS_i (ineffectiveness defined on S). In consequence of S's EV_1-PS_i performance, we either will accept H_0--or, for the benefit of those who do not accept hypotheses, strongly entertain it--or will entertain H_1 or H_2. If H_1 or H_2 exhaust the possible causes of -PS_i, then it is not necessary that EV_1-PS_i provide a definitive basis for choosing between H_1 and H_2 (although

[2]Should empirical effort reveal this assumption not to be tenable, then instructional redesign and redevelopment would be required consonant with the proposition that all Ss will require administration of every program skill.

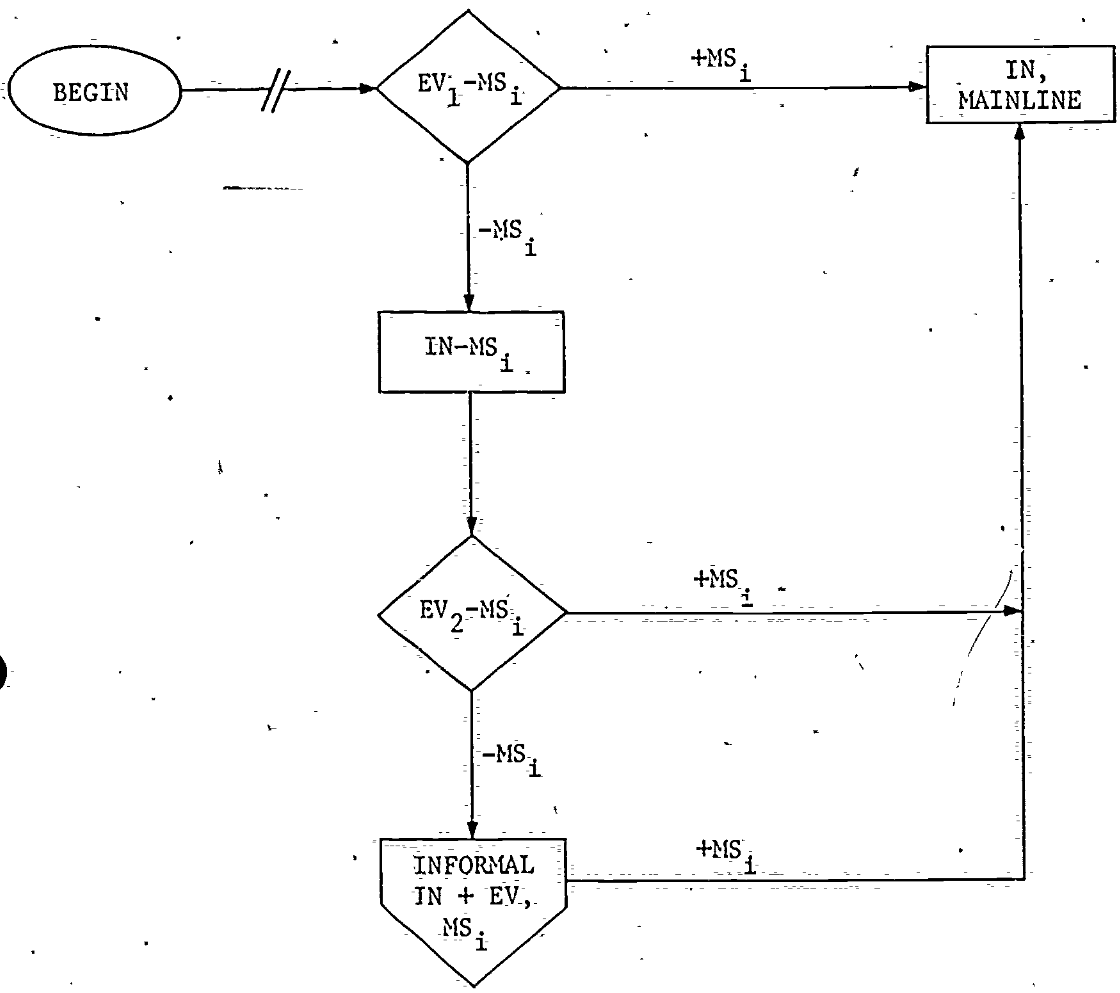

Figure 3. Illustrative Flowchart For an ith Enroute Prerequisite Skill (MS_i) Treated Interactively. EV = Evaluation, IN = Instruction, Minus Sign = Unacceptable Proficiency Level, Plus Sign = Criterion Proficiency.

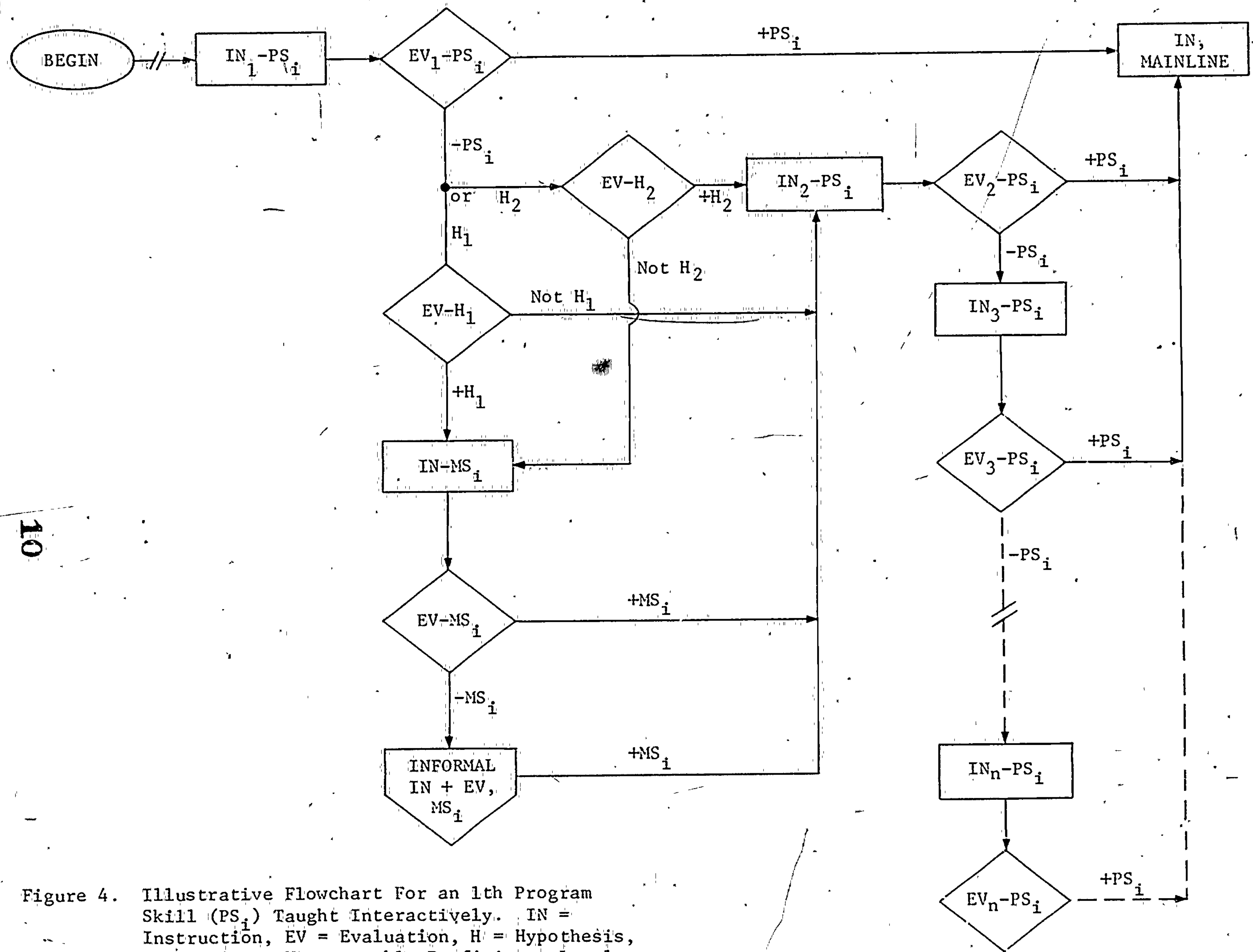

Figure 4. Illustrative Flowchart For an 1th Program Skill (PS_i) Taught Interactively. IN = Instruction, EV = Evaluation, H = Hypothesis, Minus Sign = Unacceptable Proficiency Level, Plus Sign = Criterion Proficiency Or H Is Acceptable.

instructional efficiency will be higher if the test is definitive regarding cause). Figure 4 has us evaluate either H; if it is rejected, then the other H is taken as tenable and prescriptive instruction is administered accordingly. If H_2 is found tenable, then one alternative is to treat PS_i in greater instructional depth (or to elaborate on it, to use alternate exemplars, to make its logic more explicit, etc.) An extended series--IN_2-PS_i + EV_2-PS_i through IN_n-PS_i + EV_n-PS_i--might then be devised to take instruction to whatever level of elaboration experience proves useful. Alternatively, at some point early in such a series, one might switch over to informal instruction and evaluation. While Figure 4 terminates the prescriptive series that is consonant with accepting H_2 on formal instruction and evaluation, it often will be true that an acquisition problem can only finally be overcome in consequence of informal intervention by E. Such an exit from prescriptive instruction is uncritical here because it will not be counted in the count of program elements that burdens the system.

A Closer Look at Prerequisite Skills

The distinction between ES and MS probably can be defended on grounds of evaluative efficiency when instruction is extended. However, the present illustration probably will not suffer if we collapse the two--denoted collectively EMS. Let EMS = 8. We will assume that it makes sense to test any H_1 (that a prerequisite skill is deficient) using a cursory screening test EV-EMS. If, as we will, we then assume that it is necessary to follow up on the evidence provided using a restricted test EV-EMS_i, we are assuming in effect that the diagnosis based on the cursory test may be in error. I believe such an outcome should throw S back to a second version of EV-EMS. Let us assume that the system stores four such versions EV_1-EMS through EV_4-EMS. These are stored as separate, randomly accessible segments because if retrieved as a sequence we will need tell the system where in the sequence we wish it to transmit.

Since EMS = 8, we require 8 EV_1-EMS_i and 8 EV_2-EMS_i. Letting each of these constitute one segment, then 18 such evaluative segments must be stored. Consonant with earlier remarks restricting formal instruction of defective prerequisite skills to one treatment, which we interpret as two segments, storage of an additional 18 segments will be required. The prerequisite skills storage, in segments, then is 36--4 EV-EMS, 16 EV-EMS_i, and 16 IN-EMS_i.

If EV-EMS yields $+H_1$ (or EV-H_2 yields not $-H_2$) then it will be required that one of eight 4-segment sequences be retrieved and presentation to S be initiated. Such sequences have the form EV_1-EMS_i + IN-EMS_{ia} + IN-EMS_{ib} + EV_2-EMS_i. While S may exit from the sequence following EV_1 or EV_2, if he does not exit at EV_1, then he will continue through the sequence in fixed order. Hence, the illustrative instructional program requires that four randomly accessible one-segment program elements and eight randomly accessible 4-segment program elements be stored for purposes of evaluating and instructing prerequisite skills.[3]

Segment Characteristics

For present purposes, we will define segments on time. Let the segment consist of 2-20 items, where an item may be audio only, video only, or audio + video as such characteristics can be discerned by an uncritical review of audio program and video frame storage for IDCMS. (We say uncritical because number of storable audio programs may be at issue.) Let the segment be two minutes long on the average, with number of items--2, 20, or an intermediate number determined by mean item duration. What will vary is duration of the audio message. We assume that every audio message space will be preceded by an 11-bit digital code[4] whose average duration will be just in excess of 1.5 seconds and whose average separation from the audio message space--measured from the end of the code to the beginning of the message space--will be 1 second. We assume further that a stop interval lying beyond the audio message space will use 1.5 seconds of tape. Thus, if the segment contains 20 items, then an item will occupy 6 seconds of audio tape and the audio message space can only be 1.5 seconds long (because code, code separation, and stop intervals use 4.5 seconds). On the other hand, a 2-item segment on the average will feature 55 second audio messages. Wherever one instructional sequence (without testing interruption) features a series of instructional segments, then the instructional sequence, together with the test that terminates it, can be stored as a sequenced program element. Program element storage for prerequisite skills is four 2-minute program elements and eight 8-minute program elements--or 12 elements, 72 minutes.

---

[3]While it is likely that program elements referencing to higher-numbered prerequisite skills will not need be stored throughout instruction, a continuing requirement to store 12 elements presently appears sufficiently slight that it would not be worth the trouble to simulate the slightly lower requirement that would characterize a best day requirement for six Ss. Worst day would require that all elements be stored.

[4]The system as contracted contemplates using 7-bit codes. These suffice to random access to 128-track files--which requires only four bits. If seven bits really are required just to do this, then perhaps we require 14 bits rather than 11 to be able to get to any track number of video storage. While we do not require random accessing to individual tracks for purposes of serving illustrative instruction, the requirement will go beyond a capability for accessing to the starting point of any of just 16 128-track video files. Rather, we may need to be able to access to as many such files as we define program elements.

A Closer Look at Program Skills

Foregoing remarks have dispensed with H_1. H_0 accepted, S continues on the mainline. H_0 rejected and H_1 rejected, H_2 must be accepted (unless one hypothesizes some previously-unidentified prerequisite skill--the signal to start over). Just what H_2 entails turns upon lesson level in the skills hierarchy (see Figure 1). Level 1 skills complexes are independent of other skills complexes. If H_1 is rejected following rejection of H_0, then subcriterion performance detected by a Level 1 test limits the search for defective instruction to that for the particular lesson in which the Level 1 test occurs . look at the Level 1 mainline to clarify where one can go in ... that a Level 1 test detects subcriterion performance.

We have asserted that every Level 1 lesson (Figure 1 shows nine) will feature two subskills that are taught and then integrated. The integrated program skill then has the subscript of the lesson's number. Each mainline treatment of a Level 1 lesson, then will feature three IN-EV sequences. For present purposes we assume that any such sequence consists of two instructional segments and one evaluative segment. Sequences, or program elements, for a Level 1 lesson are:

1. $IN_1\text{-}PS_{i1a} + IN_1\text{-}PS_{i1b} + EV_1\text{-}PS_{i1}$

2. $IN_1\text{-}PS_{i2a} + IN_1\text{-}PS_{i2b} + EV_1\text{-}PS_{i2}$

3. $IN_1\text{-}PS_{ia} + IN_1\text{-}PS_{ib} + EV_1\text{-}PS_{i}$

Each sequence terminates on an EV which, detecting subcriterion performance, will jerk S off the mainline. Hence, it makes sense to view Level 1 mainline storage as 9 (lessons) x 3 (sequences) 3-segment program elements, or 27 6-minute program elements.

If H_0 and H_1 are rejected in consequence of administration of $EV\text{-}PS_{i1}$ or $EV\text{-}PS_{i2}$, then there seems just one place to go--to an alternative version of $IN\text{-}PS_{i1}$ or $IN\text{-}PS_{i2}$ (or to an alternative version, with alternative procedure--e.g., regarding pacing). Were we to follow Figure 4 literally, then we could overload any system at this point, simply by requiring many versions of each sequence. For present purposes, let us require just one alternative version per mainline instructional sequence at Level 1.

If H_0 and H_1 are rejected in consequence of administration of $EV\text{-}PS_i$, then the difficulty could reside at any of three addresses. Either instruction is ineffective for one of the two subskil or it is ineffective for integration of these subskills. While this could necessitate a longer-than-one-segment $EV_1\text{-}PS_i$, its follow-on implications for alternative sequences need not differ from those for subskills. That is, again we will require just one alternative version per mainline instructional sequence that integrates subskills at Level 1. However, if the difficulty is identified as being at the subskill level and an appropriate alternative sequence is negotiated, then we wouldn't wish to have S negotiate new instruction at the integrative level just to be able to negotiate a new version of EV-PS_i. Hence, let us require a single additional version of EV-PS_i, to be used just to evaluate the program skill at the integrative level. In consequence, Level 1 looping to alternative versions of mainline instruction will require (if we circumvent EV-H_2 testing) 27 3-segment and nine 1-segment program elements. This amounts to 27 6-minute and nine 2-minute elements.

H_0 and H_1 rejected at higher levels, the source of difficulty is a more complex matter. Lessons 3, 6, and 14 of Figure 1 are at Level 2. Assuming that higher-level instruction has only the structure implied by Figure 1--that is, higher-level open-circles embrace no structure of their own--being simply integrative with respect to the lower-level skills that they subsume--then a Lesson 3 or Lesson 6 failure on an appropriate test for skills integration may be due either to the failure of Level 2 instruction or to Level 1 failure referencing to either of two subsumed skills. Lesson 14 failure, according to the same reasoning, may be due to five sources of failure. Let us assume that mainline Level 2 instruction will contain as many instructional segments as there are subsumed Level 1 skills. Thus, the Level 2 mainline sequences are:

Lesson 3: $IN_1\text{-}PS_{3a} + IN_1\text{-}PS_{3b} + EV_1\text{-}PS_3$

Lesson 6: $IN_1\text{-}PS_{6a} + IN_1\text{-}PS_{6b} + EV_1\text{-}PS_6$

Lesson 14: $IN_1\text{-}PS_{14a} + IN_1\text{-}PS_{14b} + IN_1\text{-}PS_{14c} + IN_1\text{-}PS_{14d} + EV_1\text{-}PS_{14}$

Particularly when a lesson has as many potential program skills sources of difficulty as Lesson 14 does, we either need to put EV-H_2 into the instructional system to aid pinpointing of the source or we need to increase length of the mainline test. For present purposes, we do the latter. Let EV_1-PS_{14} give way to the two segments EV_1-PS_{14a} + EV_1-PS_{14b}. Hence, the Level 2 mainline storage requirement is two 3-seg-program elements plus one 6-segment program element, or two 6-minute and one 12-minute program elements.

We can continue to live with the decision to employ only two alternate versions of Level 1 sequences if we are willing to assume that Level 1 sources implicated at Level 2 must arise simply due to "forgetting" or a need for refresher instruction. This convenient assumption makes it possible then to recycle to one of the earlier versions even though both may formerly have been used. However, if the source is found to be at Level 2, then an alternative version of the Level 2 sequence will be needed. This requirement is identical to that for Level 2 mainline instruction (see preceding paragraph).

Only Lesson 7 is Level 3. It integrates the subsumed skills PS_3 and PS_6. Treating it as we have Level 2, then mainline and looping instructional requirements each will be on the order of what we have shown at Level 2 for Lessons 3 and 6. Thus, the overall storage requirement is for two 3-segment program elements, or two 6-minute elements.

Only Lesson 9 is Level 4. Since it too integrates two skills--PS_7 and PS_8--it also imposes an overall storage requirement of two 3-segment program elements, or two 6-minute elements. Only Lesson 15 is Level 5. It also integrates two skills--PS_9 and PS_{14}--and so also imposes an overall storage requirement of two 3-segment program elements, or two 6-minute elements.

Like Level 1, Levels 2-5 each will require one alternative test per lesson that can be used independently of instruction following recycling to lower-level instruction. This adds five 1-segment (2-minute) and one 2-segment (4-minute) EV program elements to the storage inventory. Table 1 summarizes the overall storage requirement.

Table 1

Number of Program Elements, Element Lengths, and Normal-Play Minutes for Mainline and Supplemental Instruction

	Lessons	No. Program Elements	Element Length Segments	Total Normal-Play Minutes
Mainline				
EMS	0	4	1	8
Level 1	9	27	3	162
Levels 2-5	6	5	3	30
		1	6	12
Subtotals		(37)		(212)
Supplemental				
EMS	0	8	4	64
Level 1	9	27	3	162
		9	1	18
Levels 2-5	6	5	3	30
		5	1	10
		1	6	12
		1	2	4
Subtotals		(56)		(300)
Totals	15	93		512

Since the illustration is an absolute bare-bones one, guaranteed to minimize storage-retrieval requirements when several Ss receive interactive instruction of moderate extent, we further assume that a single pace of instruction is employed, with variation in instructional duration occurring in consequence of differential supplementation of mainline instruction across Ss. Thus, each S will receive 14 minutes of mainline instruction per day (212/15) on the average. This instruction will feature presentation of 2.5 program elements per day (37/15) on the average. Allowing 30 seconds for switching from one element to the next will consume an additional minute on the average. Assume now that the S having the highest acquisition rate will manifest a mainline to formal looping ratio of 4:1 and that the S having the lowest acquisition rate will manifest a 1:1 ratio. In consequence, a child whose acquisition rate is average with respect to the illustrative program will manifest a 2.5:1 ratio if the distribution of rates is symmetrical. All Ss will average 15 minutes per day in mainline instruction. An S having average acquisition rate for the program will spend 6 additional minutes in formal looping instruction. Let us further assign to this S 4 minutes per day of informal instruction and evaluation consonant with his going beyond the limited formal instructional materials that the illustrative program makes available for supplemental purposes. Thus, we define the 30-minute session for an S of average rate as consisting of 15 minutes of mainline instruction, 6 minutes of formal supplemental instruction, 4 minutes of informal supplemental instruction, and 5 minutes of break time (positioned by E in the day's instructional sequence on the basis of his perception of S's needs). The consequence is that an S having an average rate will complete the program in 15 30-minute periods.

The highest-rate S will complete as much as the S of average rate in 21.25 minutes. If he also uses 5 minutes per day for breaks, then he will complete the program in 12.75 days. The lowest-rate child will complete as much as the average rate child in 40 minutes. If he also uses 5 minutes per day for breaks, then he will complete the program in 24 days. Accordingly, the 13th day will be a worst case day for program storage. The highest rate child will complete the program that day, while the slowest rate child will be only half way through at the beginning of the day.

The program is so devised that it is theoretically possible that any of the supplemental program elements might be required at any point in instruction--a condition I feel typically will prevail in interactive instruction. Hence, only mainline storage of program elements can be deleted on the worst case day. At the beginning of Day 13, the slowest rate child should have completed approximately 106 minutes of instruction. Completion of Lesson 7 signifies negotiation of 108 minutes of the mainline treatment. This removes 18 3-segment program elements from the Table 1 inventory. What remains as the worst day storage requirement is shown in Table 2.

Table 2

Worst Day Storage Requirement

	No. Program Elements	Element Length, Minutes	Total Normal-Play Minutes
	18	2	36
	1	4	4
	46	6	276
	8	8	64
	2	12	24
Totals	75		404

In consequence of an extreme accommodation to the system, we have managed to bring the worst case audio program storage requirement down to a level that is consonant with the contractor's view of audio storage as 96 programs. All that we need to do with this simplified illustration to make it exceed 96 programs on a worst case day is adopt the view that supplemental materials should reflect two alternative versions to the mainline version, rather than one. Removing supplemental EMS elements, this adds 48 program elements. Added to the 75 for a worst day, the system becomes overburdened in light of the contractor's view of 96-program audio storage, for the worst day requirement now becomes 123 program elements.

Whatever may be true for audio is truer still for video. The 16-file view of video storage simply will not do.

The Looping Requirement

It remains to determine how the modest looping requirement inherent in the oversimplified illustration compares with the contractor's naive view that "first-level branching" epitomizes interactive instruction. The simplest (although probably not the most effective) approach to contingent instructional path specification for the illustrative program has H_0 evaluated first. $+H_0$ steps S along the mainline. $-H_0$ leads to evaluation of H_1. $+H_1$ leads S into supplemental EMS instruction with exit to a second (off-mainline) version of the instructional-evaluative program element for which $-H_0$ was obtained. $-H_1$ is interpreted as $+H_2$, leading S into the supplemental second version of the program element for which $-H_0$ was obtained. If system software will not support this degree of contingent advance, then that portion of the software that addresses the looping requirement simply will be useless.