DOCUMENT RESUME

ED 257 450 IR 011 683

AUTHOR Sleeman, D.
TITLE Intelligent Tutoring Systems: A Review.
PUB DATE [84]
NOTE 22p.
PUB TYPE Information Analyses (070) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Algebra; *Artificial Intelligence; Cognitive Style; *Computer Assisted Instruction; Computer Software; Learning Processes; Literature Reviews; Man Machine Systems; *Material Development; Methods; *Models; Problem Solving; Research Needs; Secondary Education; Student Behavior
IDENTIFIERS *Intelligent Tutoring Systems; *System Design

ABSTRACT

This paper presents a critical review of computer assisted instruction (CAI); an overview of recent intelligent tutoring systems (ITSs), including current perceived shortcomings; major activities of the field, i.e., analysis of teaching/learning processes, and extending and developing artificial intelligence techniques for use in intelligent tutoring systems; and a methodology for building ITSs. Examples are given from recent work on a student modelling system used to develop an ITS for algebra. A research agenda for ITS development relating to current activities in artificial intelligence (AI) and cognitive science is suggested and a list of 36 references is provided. (MBR)

ED257450

Intelligent Tutoring Systems: A Review

D. Sleeman

School of Education & Dept. of Computer Science

Stanford University

Stanford, CA 94. 5

Abstract

The paper includes a critical review of CAI, a overview of recent ITSs (Intelligent Tutoring Systems) and a methodology for building ITSs. Examples are given from the author's recent work on a student modelling system. Further the paper suggests a research agenda for the sub-field and relates this to current activities in AI and cognitive science.

1. Introduction

The pioneers of CAI in the late 60's suggested that through this medium highly individualized instruction would become common place. Two decades later we see fairly widespread use of computer-based instructional materials, but very little of it could be said to be adaptive. Why this shortfall? I believe the key problem with virtually all authoring languages including those available today, is that the author of the teaching material has to provide in advance a list of anticipated responses (and associated actions). Thus such systems are only able to deal with situations which have been prespecified, and are unable to respond appropriately to novel errors, or for that matter to brilliant insights.

Building teaching systems which are truly adaptive is a very demanding task. A decade or so ago it was realized that this would not be achieved without the use of artificial intelligence techniques, Carbonell [1970]. Subsequently Hartley and Sleeman [1973] suggested that intelligent tutoring systems would normally have four distinct databases:

- knowledge of the task domain
- a model/history of the student's behavior
- a list of possible teaching operations
- mean-ends guidance rules which relate teaching decisions to conditions in the student model

The very earliest systems to encorporate some of these databases where programs which generated tasks. For example Uhr and his collaborators implemented programs which generated simple arithmetic and vocabulary recall tasks, Uhr [1969]. Subsequently, systems were implemented which attempted to create a task which was appropriate to the student's competence in the task domain, Suppes [1969] and Woods and Hartley [1971]. These adaptive systems included the four data-bases given earlier but often in a very simplistic form. The initial version of the Leeds Adaptive arithmetic system, for instance, used a limited number of teaching operations and its student model consisted merely of an integer to indicate the level of the student's competence. On the other hand the original scholar system, Carbonell [1970], used a recently introduced representation, namely a semantic net, for the systems' domain knowledge and the student model. Nodes in the network had tags associated with them to indicate whether the concept was, or was not, known to the student. However, SCHOLAR-I had a poorly articulated teaching strategy, which was not represented as a separate data-base.

2. Review of Recent ITSs

In the interim a number of knowledgeable/intelligent teaching systems have been implemented. Figure 1 lists many of these systems and indicates their subject domains. Many of these systems have been used to provide supportive problem solving and not initial instruction as this is felt to be a more appropriate use. A further general point is that currently ITSs are an order of magnitude more expensive to implement than regular CAI (it is usually estimated that an hour of CAI material requires 100 hours of an experienced author's time to prepare), and so it is important that the implementors of these systems choose their topics well. I believe the topics listed in figure 1 are important ones. Indeed I believe that many of them are educational "watersheds", that is they represent topics which if not mastered will render further progress in the field (virtually) impossible. For example, it is highly significant if a child fails to become competent at clerk's or vertical arithmetic.

[Figure 1 about here]

As noted earlier building a truly responsive teaching systems implies solving a range of very open-ended problems. Each of the systems implemented has tended to emphasize some aspects and neglect others. In a recent overview Sleeman and Brown [1982] suggested that current perceived shortcomings include:

1. The instructional material produced in response to a student's query or mistake is often at the wrong level of detail, as the system assumes too much or too little student knowledge.

2. The system assumes a particular conceptualization of the domain, thereby coercing a student's performance into its own conceptual framework. None of these

systems can discover, and work within, the student's own (idiosyncratic) conceptualization to diagnose his "mind bugs" within that framework.

3. The tutoring and critiquing strategies used by these systems are excessively ad hoc reflecting unprincipled intuitions about how to control their behaviour. Discovering consistent principles would be facilitated by constructing better theories of learning and mislearning -- a task requiring detailed psychological theories of knowledge representation and belief revision.

4. User interaction is still too restrictive, limiting the student's expressiveness and thereby limiting the ability of the tutor's diagnostic mechanisms.

The field has subsequently concentrated most of its efforts on two major activities:

1. Understanding the nature of learning, mis-learning and teaching processes.

2. Extending and developing, AI techniques for use in ITSs.

2.1. Analysis of teaching/learning processes. If one does not have a good feel for the types of misunderstandings which occur with a particular subject domain, then it is impossible to write CAI material or implement ITSs which can deal with the domain effectively. Collins and his group have done significant work on protocol analysis within the ITS community in the subject areas of geography and meterology, Stevens, Collins, and Goldin [1982]. More recently Matz [1982] and Sleeman [in press] have analyzed students' difficulties with beginning algebra. However, workers in science education who have been studying the differences between experts and novices and those who work in the field of mental models have also contributed significantly to our knowledge of these issues, see for example Stevens & Gentner [1983], Davis, Jockusch & McKnight [1978].

2.2 AI Techniques evolved by the ITS Community

As noted earlier there are many central AI problems to be solved before powerful general purpose ITSs can be implemented. Before we review the techniques which have been implemented, I would like to note the additional stringent requirements which are met in this area (and are now being encountered in other areas of applied AI like Expert Systems). Namely for the systems to be acceptable in the field they must be robust (that is they must not crumble when they encounter a response which is out of their range), they must be able to cope with responses which are both incomplete and inconsistent, and thirdly they must respond fairly quickly. The techniques to be highlighted here include:

- Student modelling and concept formation.
- Friendly Natural Language Systems.

2.2.1 Student Modelling

A student model was one of the data-bases which Hartley and Sleeman [1973] suggested should be part of each ITS. The issue which has been addressed more recently is that of inferring such models from observing student's performance. The principal issues addressed have been developing techniques to:

- avoid the combinatorial explosion when producing models from primitive rules.
- cope with noisy data

Suppose one is attempting to model the incorrect behavior of students in a particular domain. The approach taken by the BUGGY [Brown & Burton, 1978] and LMS projects [Sleeman, 1982] is to provide the modelling system with a set of correct domain rules and associated incorrect rules which have been noted previously in protocols. If from task-analysis one knows that N (primitive) rules are necessary to solve the task, then a simple model-generating algorithm will produce N! variants; this is a prohibitively large number even for modest sized rule bases. (And of course including incorrect, or mal-rules will merely increase the number of models.) The BUGGY project overcame this problem by using a set of heuristics to determine the combinations of rules which could occur in any one model. The LMS project focused on a particular rule for each set of tasks and exploited properties of the rules, such as rule independence and subsumption to radically reduce the total number of models, Sleeman [1983].

Students do make careless slips, and it is important that these should not deflect the modeller unduly. The BUGGY project assumed that if the student's response was within a certain tolerance, that the error was a number-fact-retrieval problem and should be ignored, i.e., modelled as if the correct number-fact-retrieval had been made. LMS uses a simple statistical procedure to return the model which it believes explains the students behavior on a set of tasks.

2.2.2 Friendly Natural Language Interface

"Classical" Natural Language work, see Simmons [1970], provides parsers which are able to analyze sentences in natural language if the input is grammatically correct, which in turn implies it must be complete. On the other hand, humans regularly communicate with, and are understood when they use, incomplete and inconsistent utterances. Thus it is not surprising that ITSs have encountered a need to cope with these messier types of inputs.

The parser which Burton [1976] implemented as part of the SOPHIE system was described as being a fuzzy semantically-driven parser, and represented a pragmatic step in the building of ITSs and in turn made an important contribution to Natural

Language processing. The parser was so named as it looks for semantic classes such as measurement in the student's input and not for a particular syntactic entity, as in conventional parsers. It was designated fuzzy as it had the ability to ignore noise words -- particularly as it became more convinced that it was able to parser a complete input. The ACE system, Sleeman and Hendley [1982], has extended this technique to deal additionally with inconsistent and incomplete user explanations.

2.3 Another Perspective on the ITS and AI.

Another perspective on the intercorrelation between AI and ITSs is provided if one thinks of the sub-systems which are needed in a complete ITS (Task Selector, Problem Solver, Presenter, Response Analyzer, Student Modeller and overall strategy critic) and then considers the important AI research areas which each of these subsystems raise. I have attempted one such listing:

Problem Solver:

Representation, Search, Heuristics.

Presenter/Analyzer:

Natural Language (incomplete and inconsistent input).

Modeller:

Inference, Representation, Consistency of data-bases, Dialogue Control.

Critic:

Inference, Representation.

It appears that many of the research issues currently of central concern to AI, also appear here, and as we have noted earlier, in this context the solutions need to be robust, user-tolerant and efficient.

3. An evolving methodology for building ITSs

The implementors of an ITS frequently perform the following steps:

1. Analyze protocols for students from the target population solving typical tasks and codify their difficulties/misunderstandings. (This may involve detailed "clinical" interviews with the students).

2. Create data-bases for the ITS which includes a coding of the mal-rules observed in step 2.

3. Use the ITS with students and in particular note student errors which are not spotted by the system.

4. Carry out detailed student interviews to determine the nature of these misunderstandings; encode these as additional mal-rules.

Steps 3 and 4 are repeated until the system captures the majority of the bugs which occur with the target population.

3.1 Notes on the ITS Methodology

1. The above emphasizes creating and debugging a data-base of rules which represent the student's misunderstanding of the domain. In fact, ITSs have a variety of data-bases, as noted earlier, including domain knowledge, tutorial strategies (i.e., when to interrupt a student, how to present the essential diagnostic information etc.). Each of these data-bases needs to be articulated and debugged as indicated above.

2. Articulating the several data-bases involved in an ITS is an important contribution of this field to the theory of education. Such activities will

transform training and instruction from an <u>art</u> form to a <u>science</u>. In other areas of human activity, expert systems have been the vehicles for communicating knowledge within disciplines as disparate as Medicine and Engineering, Biology and Accounting. (Thus many people view ITSs as a sub-activity of the Expert Systems field; additionally having build an expert system many institutions are beginning to ask whether the same data-base can then be used in <u>instructional</u> settings. Clancey & Letsinger [1981], in particular, have looked at these issues).

3. The methodology for building ITSs given earlier also bears striking similarities to the knowledge extraction-refinement cycle used within the expert system's paradigm, Hayes-Roth, Waterman & Lenat [1983].

3.2 <u>A Detailed look at rule refinement in the domain of introductory algebra</u>

In 1981, I ran an experiment with 24 14-year-old algebra students to determine their competence at basic algebra tasks. Essentially this same data-base had been used earlier with 15-year-olds, and LMS had spotted a high percentage of their difficulties, Sleeman [1982]. The results of the experiment with the 14-year-olds were <u>very</u> different; a high percentage of the student's errors were <u>not</u> diagnosed by LMS. Again to verify that the difficulties noticed were not an artifact of interacting with LMS, I also administered a few months later a paper-and-pencil test. I subsequently interviewed all those students who had significant problems on the second test and those who had had problems on the first test which appeared to be cleared up before the second.

The results of the interviews can be summarized as:

1. Some students regularly solve algebraic equations by searching for solutions. Namely, substituting values for the variable to determine if the equation balances. So given an equation 2 x X + 5 = 23, many students would substitute X = 1, X = 2, X = 3, ...

2. Some students solve equations of the form:

$M \times X + N \times X = P$

as if the 2Xs are independent. I worked with a student who solved 9 tasks in a row by a weird but consistent algorithm, she solved tasks such as:

$3 \times X + 2 \times X = 12$

as $3 \times 2 + 2 + 4 = 12$

and wrote $X = 2$ and $X = 4$

(The first X was consistently taken as the value of the second coefficient of X, she then copied down the second coefficient - writing 3 + 2 + 2 this was then evaluated and the second X was obtained by subtracting this sum from the value on the RHS. For more discussion of this protocol see Sleeman [in press].

3. Some students had a range of <u>alternative</u> methods, and confided they were unclear when to apply which method. Some of these students occasionally used the correct method to solve an algebraic equation, and at other times appeared to use a mal-rule, see figure 2 for examples of protocols which include mal-rules.

4. Consistent use of Mal-Rules

Many of the students used mal-rules consistently. Just over half of the 24 students we saw mishandled precedence in equations of the form:

$2 + 3 \times X = 9$

A section for one such student is given in figure 2.I. Figure 2.II is part of a protocol produced by a student who collects all the numeric terms on the RHS of the equation irrespective of whether they are "free" integers or coefficients.

The student, who created figure 2.III, was remarkably consistent with his mal-rules over a whole range of task types. Note in particular how he handled task c which involves 3 X-terms. Having worked task h, he noticed that when he moved the 4 across to the RHS he changed the sign and so he suggested that when he move the X (associated with 2 x X) to the LHS, he should also change its sign. He then verbalized that

X - X is 0

and so the LHS became 0 and the rhs did not, and so he realized that this proposed solution was not possible. However, for good measure he also worked task i with the "revised" algorithm.

[Figure 2 about here]

5. "Saved Souls"

During the on-line session a student consistently solved problems of the form:

3 x X + 4 x X = 13 as

X + X = 13 + 3 + 4

However, during the second (pen-and-paper test) and during the interview she worked them correctly. Moreover, when presented with falations alternative, of the form given above, she was able to say clearly it was wrong and was able to explain clearly why it was wrong. This behavior was noted with several students.

3.3 Summary of the experiment

The principal observations stages were:

1. Some students have unstable "bugs", that is they have a whole range of "methods" [illegible] solving the same task, and they are unclear when to apply which method. [illegible] Sleeman [in preparation] for a more detailed discussion of this.)
2. Schema for generating mal-rules. Several students gave us a valuable insight into their "logic" when asked why the changed

$2 \times X + 3 \times X = 19$ to

$X + X = 19 - 2 - 3$

they said they were merely collecting all the Xs to one side and the numbers to the other. This "schema" enable us to correctly predict how the same students would process other tasks such as:

$5 \times X = 2 \times X + 18$

3. Differing types of student-errors

Figure 3 shows the same task being solved differently and both incorrectly by two separate students. I have classified the error of figure 3.1 as a manipulative error, as I believe the student essentially knows the rule but has made an error in carrying it out. I have classified the error of figure 3.2 as a parsing error as I believe this represents a significant misunderstanding of algebraic notation. (Additional examples of manipulative and parsing errors are given in Figure 2). This distinction is born up experimentally. During the course of interviews the student whose protocol is given in figure 3.1 was able explain the various stages in the transformation, whereas the other student asserted he went from the first line to the second line in one step (i.e. there were no intermediary steps).

[Figure 3 about here]

4. The difficulties of teaching algebra have been greatly underestimated. The nature of the misunderstandings noted here are being investigated, to see if teaching sequences can be devised to avoid some of the misunderstandings noted with these students.

5. Finally the rule set has been enhanced so that LMS would be able to handle many of the errors which it was previously unable to detect. (Because the parser errors have a different form from the manipulative mal-rules, LMS has also been somewhat enhanced).

For a discussion of current activities of this work see Sleeman [in press].

4. Concluding comments

The field of ITS currently has a major opportunity. It is now possible to run large ITSs on a personal computer where before one needed a sizeable fraction of large DEC-10, and so it is feasible to run large scale teaching experiments.

However, many of the technical issues raised earlier remain. I predict that the focus of work in the next decade will be on:

- analysis of the students' misunderstandings.
- providing more psychological explanations for the mal-rules observed, Brown & VanLehn [1980] and Sleeman [in preparation].
- providing more robust and more versatile natural language interfaces.
- implementing more robust user/student modelling systems.

To date these have been very limited in their scope and only captured the user's competence on a narrow task. To be really effective these models must include extensive information about the user, and be able to activate a body of inference rules which will enable the system to rapidly build a "crude" model from only general characteristics of the user. Examples of a general inference rule might be "if young and male assume the user is aggressive", an example of a more specific inference rule might be "if a second-year engineering student assume the user knows about thermodynamics". The more specific inference rules should have a higher credibility than the more general rules. For a more detailed discussion of user modelling see Rich [1983] and Sleeman [1984].

To those in a sub-field, progress often appears to be frustratingly slow. However, if one recalls that only two decades, CAI itself was in its infancy, then one might well be satisfied with the progress with ITSs. Moreover, there is good reason to expect the rate of progress will accelerate now that we have very much better programming environments in which to build systems and the possibility of using them in classrooms.

References

A. Barr, M. Beard & R.C. Atkinson (1976). The computer as a tutorial laboratory: The Stanford BIP project. International Journal of Man-Machine Studies, 8, pp567-596.

J.S. Brown & R.R. Burton (1978). Diagnostic models for procedural bugs in basic mathematical skills, in Cognitive Science, 2. pp155-192.

J.S. Brown, R.R. Burton & A.G. Bell (1975). SOPHIE. A step towards a reactive learning environment. International Journal of Man-machine Studies, 7, pp675-696.

J.S. Brown, R.R. Burton & J. deKleer (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE 1, 2 and 3. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp227-282.

J.S. Brown & K. VanLehn (1980). Repair Theory: a generative theory of bugs in procedural skills. Cognitive Science, 4, pp379-426.

R.R. Burton (1976). Semantic Grammar: an engineering technique for constructing Natural Language understanding systems. Bolt, Beranek & Newman report, 3453.

R.R. Burton & J.S. Brown (1982). An investigation of computer coaching for informal learning activities. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp79-98.

J.R. Carbonell (1970). AI in CAI: An Artificial Intelligence Approach to Computer-aided Instruction. IEEE Transactions on Man-Machine Systems, 11, pp.190-202.

W.J. Clancey & R. Letsinger (1981). NEOMYCIN: reconfiguring a rule-based expert system for application to teaching. Proceedings of Int. Joint Conf. on AI, 7, pp 829-836.

W.J. Clancey (1982). Tutoring rules for guiding a case method dialogue. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp201-225.

R.B. Davis, E. Jockusch & C. McKnight (1978). Cognitive processes in learning Algebra. Journal of Children's Mathematical Behavior, 2, pp1-320.

M.R. Genesereth (1982). The role of plans in intelligent teaching systems. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp137-155.

I.P. Goldstein (1982). The genetic graph: a representation for the evolution of procedural knowledge. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp51-77.

J.R. Hartley & D.H. Sleeman (1973). Towards intelligent teaching systems, *Int. J. of Man-Machine Studies*, 5, pp.215-236.

F. Hayes-Roth, D.A. Waterman & D.B. Lenat, eds (1983). Building Expert Systems, Addison-Wesley: Reading.

R. Kimball (1973). Self-Optimizing Computer-Assisted Tutoring: Theory and Practice. Stanford IMSSS Technical report 206.

R. Kimball (1982). A self-improving tutor. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp283-307.

M. Matz (1982). Towards a process model for high school algebra errors. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp25-50.

M.L. Miller (1982). A structured planning and debugging environment for elementary programming. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp119-135.

T. O'Shea (1982). A self-improving quadratic tutor. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp309-336.

E.A. Rich (1983). Users are individuals: individualizing user models. International Journal of Man-machine Studies, 18, pp199-214.

R.F. Simmons (1970). Natural language question answering systems. CACM, 13, pp15-30.

D.H. Sleeman (1982). Assessing competence in basic algebra. In *Intelligent Tutoring Systems*, eds D. Sleeman & J.S, Brown, Academic: London, pp185-199.

D.H. Sleeman & J.S. Brown (1982). Intelligent Tutoring Systems: an overview. *In Intelligent Tutoring Systems*, eds D. Sleeman & J.S. Brown, Academic: London, pp1-11.

D.H. Sleeman & R.J. Hendley (1982). ACE: A system which Analyses Complex Explanations. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp99-118.

D.H. Sleeman (1983). A rule directed modelling system. In *Machine Learning*, eds R. Michalski, J. Carbonell, & T. M. Mitchell, Tioga: Palo Alto, pp483-510.

D. Sleeman (1984). UMFE: User Modelling Front-end subsystem. Stanford HPP working paper.

D. Sleeman (in press). Basic Algebra revisited: a study with 14 year-olds. Journal of Mathematical Behavior.

D.H. Sleeman (in preparation). An attempt to understand students' understanding of Algebra.

R.L. Smith, H. Graves, L.H. Blaine & V.G. Marinov (1975). Computer-assisted axiomatic mathematics: informal rigor. In Computer Education O. Lecarne & R. Lewis (eds). North Holland: Amsterdam.

A.L. Stevens & A. Collins (1977). The goal structure of a socratic tutor. Bolt Beranek & Newman Technical report 3518.

A. Stevens, A. Collins & S.E. Goldin (1982). Misconceptions in student's understanding. In Intelligent Tutoring Systems, eds D. Sleeman & J.S. Brown. Academic: London, pp13-24.

A. Stevens & D. Gentner (1983). Mental models Erlbaum: New Jersey.

P. Suppes (1967). Some theoretical models for mathematics learning. Journal of Research and Development in Education, 1, pp5-22.

L. Uhr (1969). Teaching machine programs that generate problems as a function of interaction with students. Proceedings of the 24th National Conference, pp125-134.

P. Woods & J.R. Hartley (1971). Some learning models for arithmetic tasks and their use in computer-based learning. British Journal of Educational Psychology, 41, pp35-48.

Figure 1

Some of the Intelligent Tutoring Systems Implemented in the 1970's.

1. Symbolic integration, Kimball [1973 & 1982].

2. Solving quadratic (by a discovery method), O'Shea [1982].

3. Axiomatically-based mathematics, the EXCHECK system, Smith et. al. [1975].

4. Electronic trouble-shooting, the several SOPHIE systems, Brown, Burton & Bell [1975] and Brown, Burton & de Kleer [1982].

5. Interpretation of NMR spectra, the PSM-NMR system, Sleeman & Hendley [1982].

6. Socratic dialogue in geography and metereology, the WHY system, Stevens, & Collins [1977].

7. Medical diagnosis, the GUIDON System, Clancey [1982].

8. Informal gaming environment: the WEST system, Burton & Brown [1982], and the WUMPUS system, Goldstein [1982].

9. Program-plan debugging, the SPADE system, Miller [1982].

10. Basic programming, the BIP system, Barr, Beard & Atkinson [1976].

11. A consultancy system for users of MACSCMA, an algebraic manipulation system, Genesereth [1982].

Figure 2

Three examples of very consistently used MAL-RULES.

```
I) Pupil AB-11 on task set 7.
 a) The task given was : 4 + 2 * X = 16

    Pupil writes          1) 6X = 16
                          2) X = 2.6666
 b) The task given was : 2 + 4 * X = 14

    Pupil writes          1) 6 * X = 14
                          2) X = 2.333
 c) The task given was : 3 + 5 * X = 11

    Pupil writes          1) 8 * X = 11
      (and is told she can leave it in that form)
 d) The task given was : 5 - 3 * X = 11

    Pupil writes          1) 2 * X = 11
      (and is told she can leave it in that form)


II) Pupil AB-17 on task set 5
 a) The task given was : 2 * X + 4 * X = 12

    Pupil writes          1) * X * X = 12 - 2 - 4
                          2) X ** 2 = 6
                          3) X = ROOT 6
 b) The task given was : 2 * X + 3 * X = 10

    Pupil writes          1) * X * X = 10 - 2 - 3
                          2) X ** 2 = 5
      (and is told he can leave it in that form)
 c) The task given was : 2 * X - 3 * X = 10

    Pupil writes          1) * X * X = 10 - 2 + 3
                          2) X ** 2 = 11
      (and is told he can leave it in that form)


III) Pupil AB-18 on task sets 5, 6, 7 and 8.
 a) The task given was : 2 * X + 3 * X = 10

    Pupil writes          1) 2 * X = 10 - 2 - 3
                          2) 2 * X = 5
                          3) X = 2.5
 b) The task given was : 3 * X + 5 * X = 24

    Pupil writes          1) X + X = 24 - 3 - 5
                          2) 2 * X = 16
                          3) X = 8
 c) The task given was : 3 * X + 4 * X + 5 * X = 24

    Pupil writes          1) X + X + X = 24 - 3 - 4 - 5
                          2) 3 * X = 12
                          3) X = 4
 d) The task given was : 2 * X + 4 = 20

    Pupil writes          1) X = 20 - 2 - 4
                          2) X = 14
```

BEST COPY AVAILABLE

e) The task given was : 3 * X + 5 = 7

Pupil writes 1) X = 7 - 3 - 5
2) X = -1

f) The task given was : 4 + 3 * X = 14

Pupil writes 1) X = 14 - 3 - 4
2) X = 7

g) The task given was : 5 + 6 * X = 20

Pupil writes 1) X = 20 - 5 - 6
2) X = 9

h) The task given was : 4 * X = 2 * X + 6

Pupil writes 1) 2 * X = -4 + 2 + 6
2) 2 * X = 4
3) X = 2

Pupil then wrote 1) X - X = 2 + 6 - 4
2) 0 = 4
and QUITS.

i) The task given was : 5 * X = 3 * X + 6

Pupil writes 1) 0 = 4
and QUITS.

Figure 3

Two protocols illustrating both manipulative and parsing mal-rules.

I $\quad 6 \times X = 3 \times X + 12$

$9 \times X = 12$

$X = 12/9$

$X = 4/3$

II $\quad 6 \times X = 3 \times X = 12$

$X \times X = 12 + 3 - 6$

$2 \times X = 9$

$X = 9/2$