Student Models:
The Genetic Graph Approach

by

Barbara Jane Wasson

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-85-10
May 1985

Student Models: The Genetic Graph Approach

by

Barbara Jane Wasson

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, 1985

ABSTRACT

The widespread use of computers as educational tools is aided by the rapidly expanding market of instructional and diagnostic systems. Computer-aided instruction (CAI) systems are used to teach or tutor many different subjects. Traditional CAI systems have many limitations. Attempts at overcoming these limitations resulted in the development of more sophisticated systems often referred to as intelligent-CAI (ICAI) systems.

A major component of an ICAI system is the student model. We identify certain criteria which an ideal student model must satisfy and we use these criteria to evaluate existing modelling techniques. One particularly promising approach, due to Goldstein, is to use a genetic graph as a framework from which the student model is obtained. Goldstein's method is described and its potential as a base for the student model is examined in detail. We propose extensions to the genetic graph which are necessary to fulfill the requirements of the ideal student model.

One advantage of using the genetic graph approach is the flexibility of the design to support ICAI in diverse domains. To illustrate this feature and to demonstrate that the proposed extensions to Goldstein's model are useful, two radically different domains, subtraction and elementary ballet, are illustrated as genetic graphs.

ICAI systems would be greatly improved if the student model were dynamically expandable. We discuss the necessity and feasibility of this expansion, and eventually argue that this is an attainable goal.

Acknowledgements

First of all, I would like to thank Dr. Marlene Jones for her suggestion of topic and supervision of this thesis. Her enthusiasm for the subject provided motivation and her constant encouragement made the thesis exciting to work on. I cannot thank her enough for her friendship and caring.

I would like to thank my readers Dr. David Poole and Dr. Robin Cohen. David provided a great sounding board for many of the ideas and his enthusiasm provided encouragement during dry periods of the design. Robin's input on the final version of the thesis helped to polish the definitions and grammer. Thank you very much. I would also like to thank Dr. Neil Ostlund for allowing me to use his lab to produce my diagrams. Thanks to the guys in the lab for all their assistance.

I would also like to mention the members of the bullpen and the others who spent many wonderful times together during the past year and a half in Waterloo; the baseball and Friday dinners made the time some of the best in my life (not to mention the heated discussions during lunch in the office).

The acknowledgements would not be complete without thanking Tim for his friendship, encouragement and love during the duration of this thesis (thanks for introducing Mar and I). This thesis could not have been completed without his endless attempts to get the typesetting to work properly.

Finally I would like to thank my family, Dana, Jane and Brad (I'll mention Dee too mom). I thank my father for introducing me to Computer Science and my mother for her constant encouragement all of my life. Thanks to Brad for always being proud of his big sister.

Financial support for this thesis come from three sources; NSERC, Bell-Northern Research and the Department of Computer Science at the University of Waterloo and I would like to thank them very much.

to my parents

Table of Contents

Chapter 1

CAI Systems

Computer-assisted instruction (CAI) systems were designed to provide a teaching environment where a student interacts with a computer program. Early CAI systems were concerned with [21], [29]:

1. providing individualized one-to-one instruction,
2. teaching as well as a good human teacher, and
3. trying to avoid copying human frailties and limitations while imitating human abilities.

Unfortunately these goals were not attained by the early script-based CAI systems. These were essentially drill-and-practice units which could only provide information, generate questions and evaluate a student's answer based upon relatively primitive criteria. For a general discussion of CAI systems see [14], [28]. Compared to human teachers, the majority of CAI systems suffered from [16],[27]:

1. an inability to conduct conversations with the student in the student's natural language,

2. an inability to understand the subject being taught and hence being unable to accept unanticipated responses,

3. an inability to decide what should be taught next,

4. an inability to anticipate, diagnose and understand the student's mistakes and misconceptions, and

5. an inability to improve or modify current teaching strategies or learn new ones.

Since the 1970's, Artificial Intelligence (AI) research in areas such as natural language understanding, knowledge representation, planning, expert systems and learning, has had an impact on the quality of more recently developed CAI systems. Attempts at overcoming the above limitations resulted in the development of more sophisticated systems often referred to as intelligent-CAI (ICAI) systems; Chapter 2 describes ICAI systems and their components.

One important component is the student model. Chapter 3 lists criteria for the ideal student model and we use these criteria to evaluate existing modelling techniques. One modelling technique, the genetic graph, which is due to Goldstein [19], looks particularly promising. Chapter 4 explains why and describes proposed extensions to Goldstein's model.

To illustrate the feasibility of the genetic graph approach, two diverse domains, subtraction and ballet, were chosen for implementation. Chapter 5 looks at these two domains and shows that the genetic graph method is appropriate.

There is a great deal of research necessary to achieve the goals of ICAI systems and make them an attractive educational tool. Chapter 6 provides conclusions and possible directions for future research in the area.

Chapter 2
ICAI Systems

2.1. What is ICAI?

Advancements made in AI have led to the development of ICAI, knowledge-based CAI or AICAI systems, which attempt to overcome the limitations of the early CAI systems. An ICAI system provides what Brown [4] refers to as a "reactive" learning environment, which allows the student to be actively engaged with the system while the system analyzes a wide range of student responses by using an embedded domain-expert. Either the student or the system can initiate interactions. The student's search for knowledge and his/her misunderstandings drives the dialogue with the tutor. These enhancements to traditional CAI systems provide more flexibility and sensitivity as well as allow the learner's preferences to be considered when deciding what to teach next. These new features are steps towards what Goldstein [19] refers to as a "learner-based" rather than "expert-based" paradigm. He cites three areas of research which are contributing to this new paradigm:

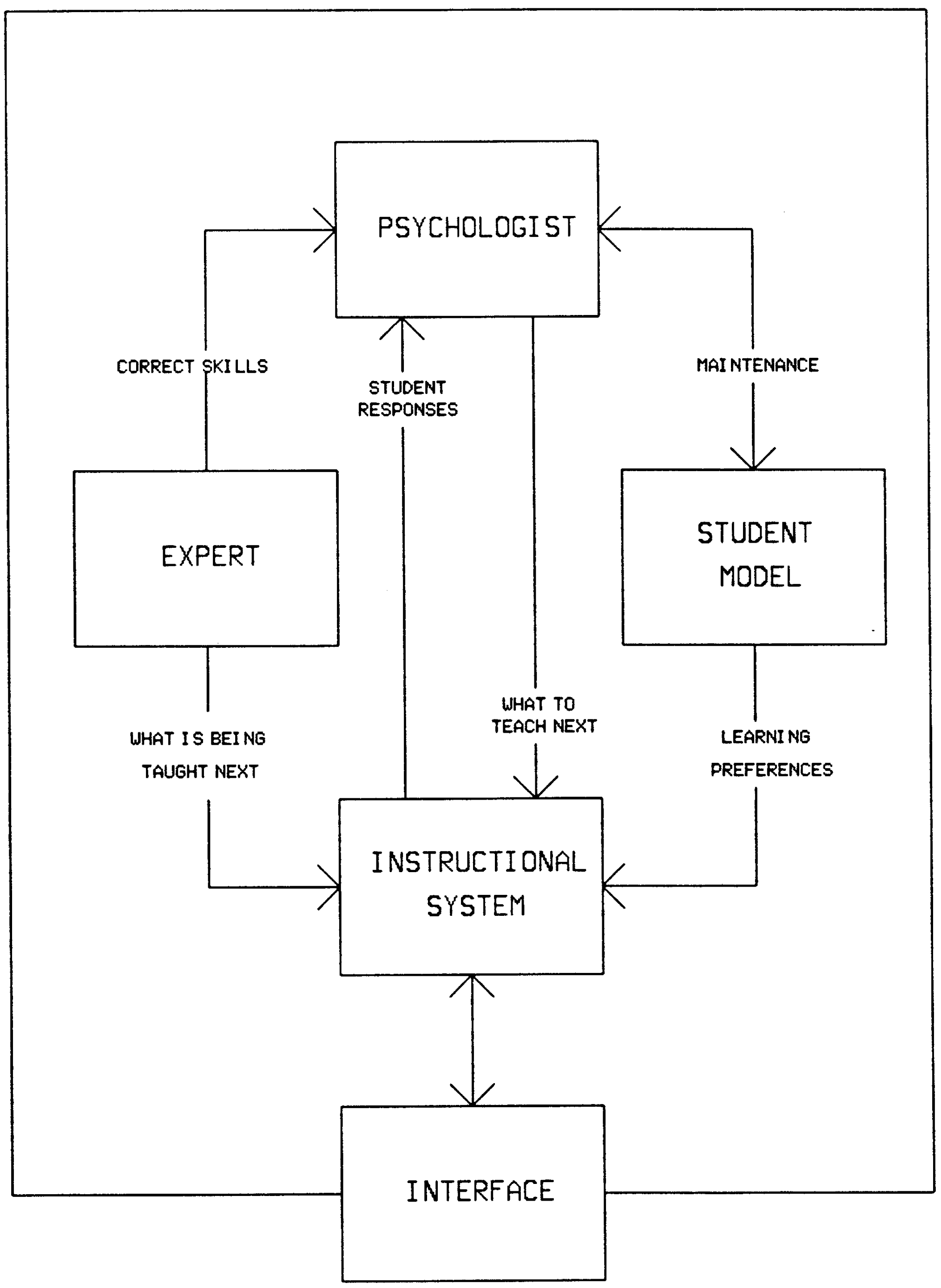
1. more sophisticated modelling of the student's knowledge and learning style [5], [6], [12],
2. widening the communication channel from student to teacher via natural language interfaces [9],
3. developing a theory of teaching skills [17], [27].

2.2. ICAI Components

An ICAI system must be able to analyze a wide range of student responses, model the student's current knowledge state including misconceptions, teach in a variety of ways, diagnose and/or determine what and when to teach and be able to engage in appropriate interactive conversations. Five major components are needed to handle these requirements. These components are the expert, the student model, the psychologist (named by Goldstein [19]), the instructional module and the interface. Figure 2.1 illustrates the interaction of these components.

The Expert

The role of the expert in an ICAI system is to contain the domain-specific knowledge. This will take the form of facts, descriptions, relationships, information regarding problem-solving skills and/or procedural information. The expert embodies knowledge of a particular application area, which combined with inference mechanisms enables the ICAI system to employ this knowledge in problem-solving situations [20]; this may be necessary in order



AN IDEAL ICAI SYSTEM

Figure 2.1

to determine correct answers to questions generated by the instructional component. The inference mechanisms could be in either the expert module or in the psychologist. For purposes of discussion herein, we assume they are in the psychologist. The psychologist uses this knowledge to perform diagnoses and evaluate responses, and the instructional component uses this knowledge to generate questions and to determine what to teach next.

Student Model

The student model is the representation of the current knowledge state of a student. The model must be able to represent knowledge, concepts and/or skills which the student has acquired, as well as those to which the student has been exposed and has demonstrated some partial understanding. There must be a mechanism by which to represent misconceptions, bugs or erroneous information which the student has demonstrated or it is suspected he/she has. The model should be sophisticated enough to indicate the student's learning preferences. For example, does the student learn best by being told, by manipulating objects, through examples or through the use of analogies. (For examples of related AI research being performed in the area of "learning" see [11], [18], [34], and references therein). The tutor, in conjunction with the psychologist, obtains this information from the student's answers and implicit problem-solving behaviour. Specific questions directed to either the student or the teacher can provide relevant historical information regarding the student's past experience. All this information is analysed and incorporated into the student model by the psychologist.

Psychologist

The psychologist is the main driving component of an ICAI system, and unfortunately the least developed module to date. The psychologist must

1. maintain the student model by determining

   -when a skill is mastered

   -errors in skills

   -what skills are being learned

2. determine what is to be taught next

To maintain the model there must be sophisticated algorithms to determine when a skill has been mastered and another set of algorithms to decide which skills caused errors to arise in the student's responses. Carr and Goldstein [12] discuss algorithms to determine when a skill has been acquired. In some systems it might be beneficial to include an entire educational diagnostic expert within the psychologist. For example, if the domain being taught was subtraction, the DEBUGGY or IDEBUGGY system [6] for diagnosing subtraction bugs could be included. If the domain being taught was reading, the diagnostic system developed by Colbourn [16] for diagnosing reading disabilities could be included.

In order to maintain the student model, the psychologist must have access to the domain expert, which contains the correct skills. The psychologist uses a correct skill to determine the state of the student with respect to learning that skill. The psychologist accesses the student model to obtain information on what to teach next. This information is then passed to the instructional module.

Instructional Module

The Instructional module, the fourth component of an ICAI system, is often referred to as the coach or tutor. Self [31] cites three types of knowledge that are required by instructional systems:

1. knowledge of how to teach,
2. knowledge of what is being taught,
3. knowledge of who is being taught.

This information comes from both external and internal sources. The expert component provides the correct form of what is being taught, and the student model provides information about who is being taught. After the psychologist has determined what and when to teach, the instructional module must determine how to teach it. Teaching in this sense includes educating the student with new information as well as testing the student on specific information that has previously been presented.

Every student is unique and one characteristic of a good teacher is his/her ability to vary the teaching methodology for different students and topics. The ideal instructional module should have this capability. Therefore, there must be a variety of instructional techniques within this component. The teaching strategies which are most commonly employed in an ICAI system are [16];

1. coaching the student within a particular activity such as a game playing situation. The intent is to manipulate the environment and the coaching such that particular skills or general problem-solving ability is acquired.

2. questioning the student to encourage reasoning about current knowledge and to modify or formulate concepts. This may involve simulations or games in which the child can discover facts or laws.

3. providing tasks for the student and evaluating the responses in order to detect the student's misconceptions.

The ICAI system would be greatly enhanced if this module had the ability to improve its instructional ability over time. In particular, it would be beneficial to be able to detect individual student preferences and instructional techniques which are successful with a number of students. Stumbling blocks, which are common to a many of students, need to be observed so that they can be avoided in the future. Recent work in the area of self-improving systems has been done by O'Shea [27].

The tutor is also responsible for deciding when to intervene with explanations, hints and further examples. There are several factors to take into account when making this decision. For example, what is the level of expertise of the student? It might be wise to allow a longer struggle for an advanced student than for a novice. Which explanation strategies have the most beneficial results with this particular student? This can be determined from the student model. Does the student appear to get bored or lose interest easily? Does the student like to explore the domain on his/her own? All these questions need to be handled within this module and the flexibility to deal with diverse situations needs to be built-in.

It is a complicated and difficult task to decide when to teach a particular skill. This is where information regarding prerequisite skills or knowledge is needed. It may be advantageous to include planning strategies in the teaching module. For recent advances in this area see [29]. Past experience in introducing new material can also be obtained from the current student model.

Interface

The last ICAI component, the interface module, is responsible for the interaction and dialogue between the student and the system. Its goal is to enhance the dialogue between the student and the system by means of a natural language interface which attempts to understand whatever the student provides to the system. The important properties of the interface are effectiveness, efficiency and friendliness. There is an enormous amount of relevant research being done within the area of natural language understanding; the interested reader should consult [33]. With respect to specific ICAI systems, the reader may wish to examine the interface which has been incorporated into the SOPHIE system [8].

Chapter 3
The Student Model

3.1. What Is Really Needed

A student model represents, at any given time, the learner's knowledge state and understanding of the domain being taught. The ideal model includes both facts and procedural knowledge about what the student has or has not mastered/grasped. There must be a mechanism for representing misconceptions and deviations from the expert's knowledge and a means of indicating corrections of these.

The representation scheme for a student should be appropriate to many domains. It should be flexible enough to include either an incredible amount of detail or a significantly small amount, depending on the domain and the learner.

It would be beneficial to recognize whether a particular student seems to learn in a particular way. For this reason, the model must include information about the style of learning that the student follows by providing information about how he/she progresses through the domain. Given this information, the tutor could then suggest remedial work that would be most profitable for the student.

This chapter surveys existing ICAI systems which model the learner and evaluates them based on these criteria.

3.2. BUGGY and the Skill Lattice

Burton and Brown [6], [10] proposed "the BUGGY model" in 1978 to explain student misconceptions or errors which surfaced in the simple arithmetic task of subtraction. Subtraction can be viewed as a set of procedural skills or subskills and a bug occurs when a variation of the correct procedure is employed by the student. Three modifications of a correct procedure are cited as causes of "bugs". They are: the deleting of part of the correct procedure, adding incorrect sub-procedures and the replacing of correct sub-procedures by incorrect ones.

Burton [10] defines a subskill as an isolatable part of a skill that is possible to mislearn. These subskills are placed in a skill lattice which is the hierarchical relationship between them. Figure 3.1 shows the skill lattice and Table 3.1 defines the fifty-eight corresponding subskills. The skill of possessing all correct subtraction subskills is the maximal element, and the absence of all subtraction subskills is the minimal element.

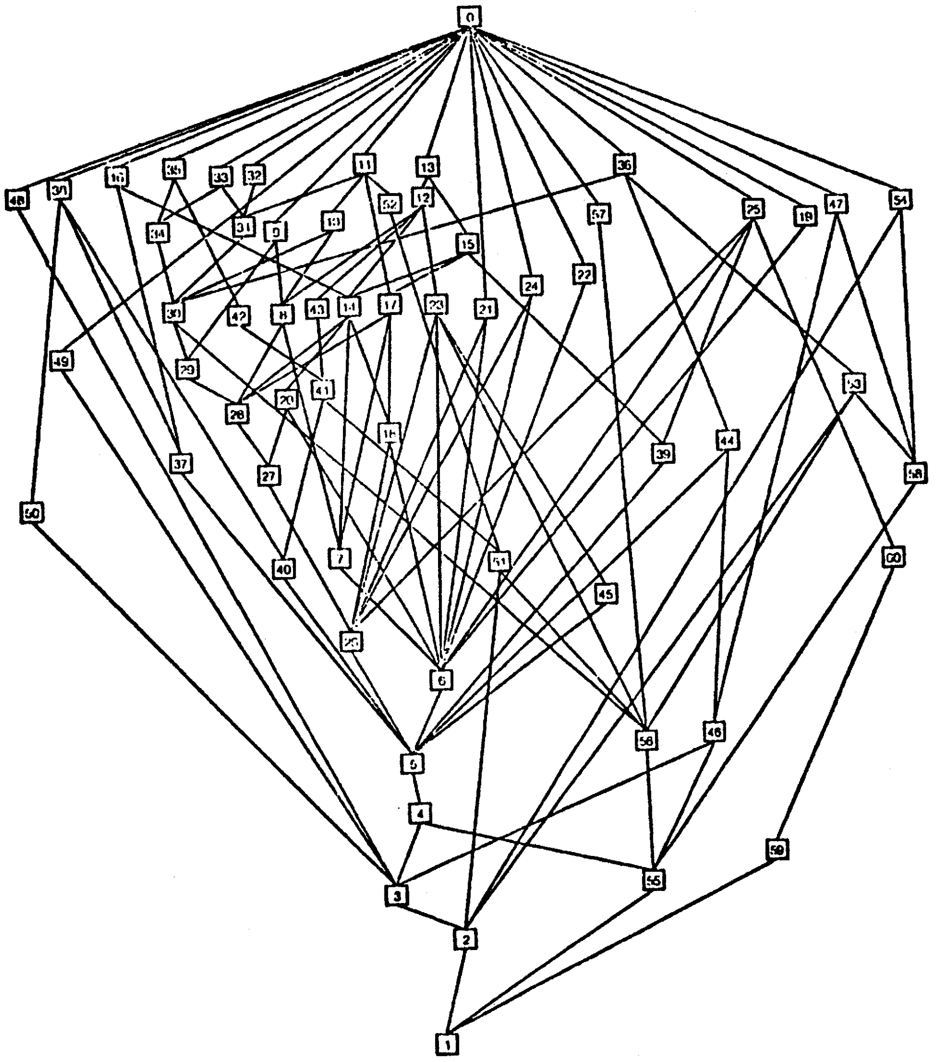


Figure 3.1 [10]

0	**Correct skills**
1	**No subtraction skills**
2	**Subtract 0 from a number whose right digit is 0**
3	**Subtract 0 from a number**
4	**Subtract columns which are n-n or n-0**
5	**Subtraction without borrowing**
6	**Borrow from somecolumns with a digit on the bottom**
7	**Borrow from columns whose answer is the same as their bottom digit**
8	**Borrow from a column with a 9 or a larger number on top**
9	Borrow from a column with a larger top digit
10	Borrow from a column with zero on top when blank on bottom
11	Borrow from a column with zero on both top and bottom
.2	Borrow from a column unless one is on top and a non-zero digit is on the bottom
13	Borrow from a column unless a one on top and bottom
14	Borrow in two consecutive columns
15	Borrow from columns with the same digits on top and bottom
16	Borrow from columns that have nine as the bottom digit
17	Borrow from columns with the same non-zero digits on top and bottom
18	Borrow from columns which have an answer of zero
19	Borrow from the leftmost column when it has a non-blank in the bottom
20	Borrow more than once per problem
21	Can borrow then not borrow
22	Borrow from columns with two on the bottom
23	Borrow from leftmost columns or columns that have a non-one on top
24	Borrow from columns with top digits smaller than the bottom digits
25	Bo:row from columns that have a top digit one less than the bottom digit
26	Borrow from the leftmost column
27	Borrow once in a problem
28	Borrow from columns that have the top digit larger than the bottom
29	Borrow from a column with the top digit greater than or equal to the bottom
30	Borrow from a column with a zero on top
31	Borrow from a middle (not leftmost) column with a zero to the left
32	Borrow from leftmost column of the form one followed by one or more zeroes
33	Borrow from a column with a zero on top and a zero to the left
34	Borrow from or into a column with a zero on top and a zero to the left
35	Borrow into a column with a zero on top and a zero to the left
36	Borrow from a column with a zero on top and a blank on bottom
37	Borrow into a column with a nonzero digit on top
38	Borrow into a column with a one on top
39	Borrow when difference is 5
40	Subtract columns with a one or a zero in top that require borrowing
41	Subtract columns with a zero in the top number that require borrowing
42	Subtract a column with a zero on top that was not the result of decrementing a one
43	Borrow into a column with a zero on top when next top digit is zero
44	Borrow from a column with a blank on the bottom
45	Borrow from a column with a one on top
46	Subtract numbers of the same lengths
47	Subtract a single digit from a large number
48	Subtract columns unless the same digit is on top and bottom
49	Borrow all the time
50	Subtract when one is in top
51	Subtract when neither number has a zero unless the 0 is over a 0
52	Subtract columns when the bottom is a zero and the top is not zero
53	Subtract when a column has a zero over a blank
54	Subtract numbers which have a one over a blank that is not borrowed from
55	Can subtract a number from itself
56	Subtract numbers with zeros in them
57	Subtract problems that do not have a zero in the answer
58	Subtract numbers when the answer is no longer than the bottom number
59	Subtract leftmost columns that have top and bottom digits the same
60	Subtract columns that have top and bottom digits the same

Table 3.1 [10]

The BUGGY System

The BUGGY system is based on the belief that a student's arithmetic errors are not random but rather are consistent discrete modifications of the correct arithmetic procedure. The system is in the form of a procedural network [30]. The domain expert contains information regarding common arithmetic bugs or procedural errors. In the case of subtraction, 110 primitive bugs are included as well as 20 common compound bugs. The results of all 130 bugs are compared with the student's answers. Based on these comparisons, the system selects a subset of bugs, each of which explains at least one of the student's wrong answers. The elements of this initial set of bugs are then combined to generate additional hypotheses for the particular student in question. Now the system starts eliminating hypothesized/proposed bugs; for example, one rule employed in this process is to remove bugs which are subsumed by other bugs and form combinations of the remaining bugs. Ultimately, each of the remaining proposed bugs is classified according to how well it explains the student's answers. This classification procedure takes into account the number of predicted correct and incorrect answers as well as the number and type of mispredictions. Hopefully at the end of this classification procedure, one bug (primitive or compound) can be selected as the best explanation of the student's erroneous responses.

Diagnosis Description

Burton provides many different definitions of diagnosis and choses "determining what internalized set of instructions or rules gives results equal to the student's results" [10, p4] for his purposes. He states that for subtraction the diagnosis must predict not only whether the answer is incorrect, but exactly what digits of the incorrect answer are in error. The system will be able to predict answers to a problem based on the history of problems answered to date.

There are situations where the diagnosis may not work. There may be more than one bug that explains the error, or it is possible that no bug will explain the error. Furthermore, certain aspects of a student's behaviour, such as laziness, boredom or fatigue cannot be represented as a bug. Burton recognized that because of this potential flaw, no exact match can be expected and hence the best match must be found and then the system must decide if it is indeed a diagnosis. Hence, the assumed bug cannot be rejected with the first incidence of contrary evidence.

Burton cites four factors that complicate the diagnosis. They are identified as performance lapses (as mentioned above), errors in primitive subskills (specialized subskills may have errors), subskill variants (a subskill used in one instance does not apply in another) and compound bugs (more than one part of the skill is being misused).

BUGGY Versions

Two systems were developed: DEBUGGY which is an off-line version and IDEBUGGY which is interactive. The skill lattice is built from the given test and then another skill lattice is built from the answers given by the student, and the two are compared. DEBUGGY is really a pencil and paper system and hence is great for group administration and can be given as a commercially available standardized test. Other standardized arithmetic tests can also be used and the results can then be fed into DEBUGGY. IDEBUGGY is most appropriate for individual informal testing, which is particularly important during the latter stages of educational diagnosis. DEBUGGY has been extensively tested with about 1000 students [10].

Comparison

Comparing the skill lattice to the ideal model creates serious doubts as to whether it can be called a student model. There is no representation of facts or procedural knowledge, just the subskills which were incorrectly performed. There is flexibility in the sense that the amount of detail representing each child depends on how he/she answered the problems, but the student's subskill lattice can be equated to a subset of the expert's skills by noticing that the student can only possess bugs defined in the maximal skill lattice. There is no representation of how the child progressed through the domain and hence no way to determine a particular learning style for the student. Remedial help can be decided based upon the bugs that exist and this could be incorporated into the diagnosis if it does not already exist.

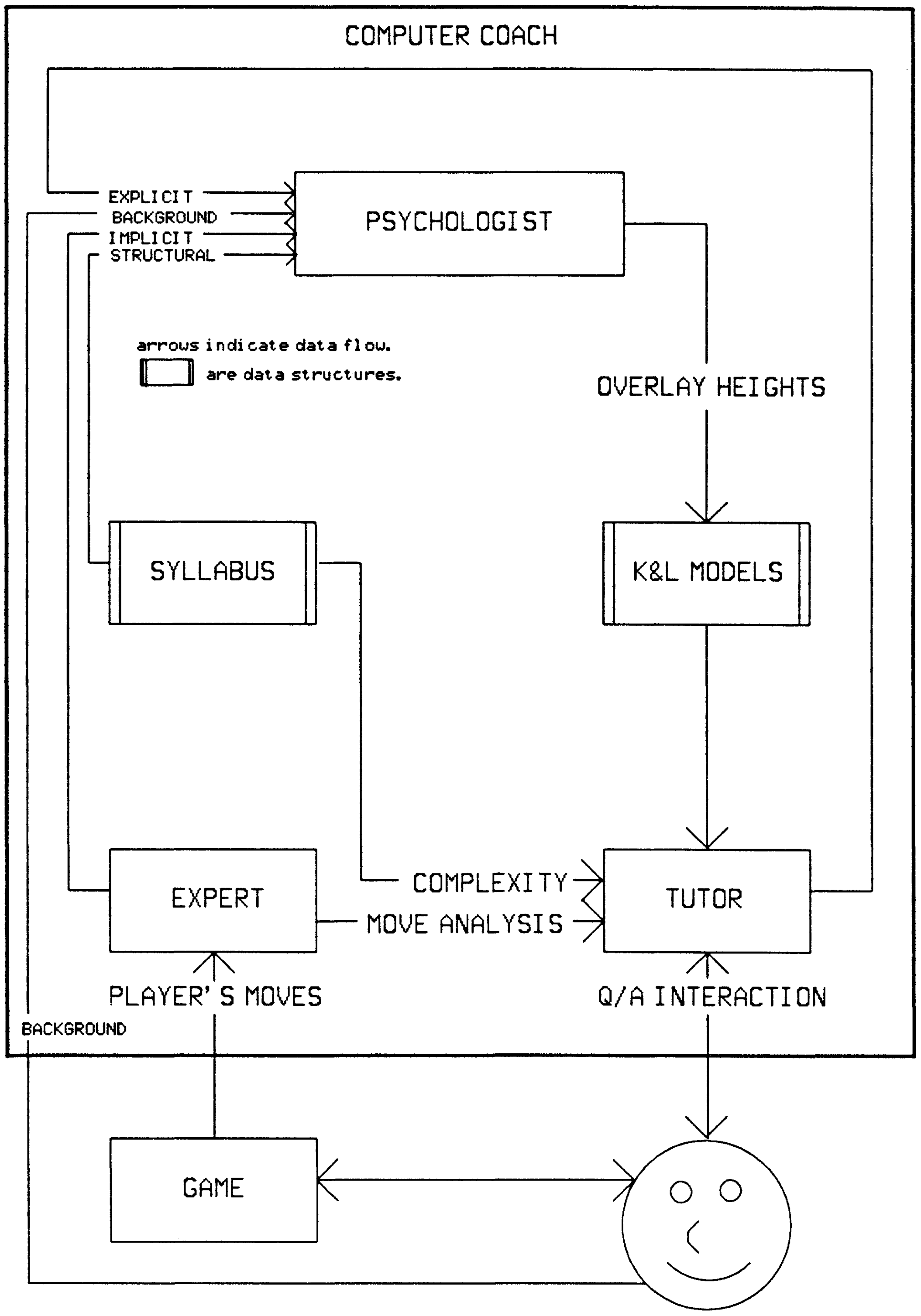
The BUGGY model is impressive in its diagnostic capabilities and it satisfied the purpose for which it was intended. However, it is insufficient by itself for use in a true ICAI system.

3.3. Overlays

Carr and Goldstein [12] realized that an effective ICAI system is extremely dependent upon a detailed model of the learner. They developed the overlay model which is a technique for describing the learner in terms of a subset of the expert's skills. The overlay model is a set of hypotheses regarding the student's familiarity with the expert's skills. In Carr and Goldstein's design, the theory of the overlay model is embedded in the Psychologist, as illustrated in Figure 3.2.

Domain Description

The domain chosen by Carr and Goldstein, was the game WUMPUS developed by Yob [36]. This mathematical game tests logic, probability, geometry and decision analysis skills of the player. The objective of the game is to slay the horrid WUMPUS with one of five arrows. The game takes place in a maze of caves, in which various obstacles can reside. These



BLOCK DIAGRAM OF A COMPUTER COACH

Figure 3.2 [12]

obstacles consist of bats, pits and the WUMPUS. Initially the player is positioned randomly in a cave and is given information about the neighbouring caves. The information provided is in the form of warnings. On encountering a bat or pit which is one cave away, the warnings are "SQUEAK! A bat is near", or "Brrr! There is a draft", respectively. The WUMPUS can be smelt either one or two caves away and is indicated by the warning "What a Stench". If the player moves into a cave with a bat, the player is randomly dropped elsewhere in the maze. Movement into a pit results in a fatal fall, and if the WUMPUS is met head on, the game is over. Victory results from shooting an arrow into the WUMPUS's lair, and defeat occurs if the player dies or wastes all five arrows.

Game Discussion

The player can improve his game by making logistic and probabilistic inferences from warnings, and using these to determine what move to make next. The more the player learns to use this information, the more skilled the player becomes.

The expert's rule-based knowledge infers the risk of visiting a new cave. Five examples of these rules are as follows [12]:

L1: *(positive evidence rule) A warning in a cave implies that a danger exists in a neighbour.*

L2: *(negative evidence rule) The absence of a warning implies that no danger exists in any neighbours.*

L3: *(elimination rule) If a cave has a warning and all but one of its neighbours are known to be safe, then the danger is in the remaining neighbour.*

P1: *(equal likelihood rule) In the absence of other knowledge, all of the neighbours of a cave with a warning are equally likely to contain a danger.*

P2: *(double evidence rule) Multiple warnings increase the likelihood that a given cave contains a danger.*

The following figure is an overlay model of a player who has mastered rules L1, L2, P1 and is in the process of learning rule L3, and has not attempted rule P2.

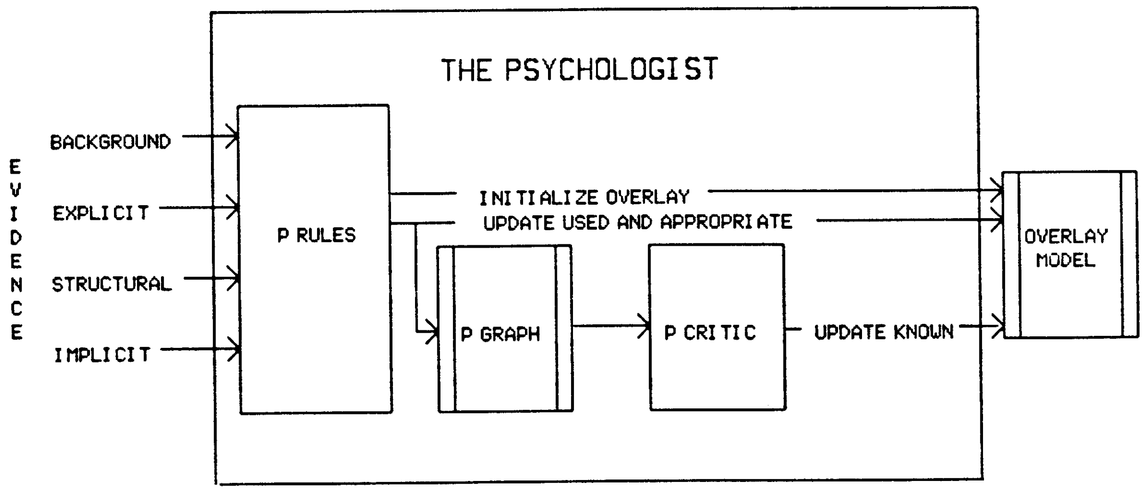
RULES	APPROPRIATE	USED	FREQUENCY	KNOWN
L1	9	9	100%	T
L2	4	3	75%	T
L3	4	2	50%	?
P1	8	8	100%	T
P2	5	1	20%	NIL

Overlay Model

In this model, FREQUENCY is the proportion of the number of times a skill is USED to the number of times the rule has been APPROPRIATE. The KNOWN variable indicates whether a player has acquired a skill (indicated by T), is learning a skill (indicated by ?), or has not learned a skill (indicated by NIL).

The Modelling

The overlay model is constructed by a set of rules which are triggered by many sources of evidence, and a critic that is responsible for discovering inconsistences and discontinuities in the learner's behaviour. Figure 3.3 shows these components and illustrates their interaction.



INTERNAL STRUCTURE OF THE PSYCHOLOGIST
Figure 3.3 [12]

P rules

The rule system, called P rules, is a set of procedures used to modify the overlay model, by gathering implicit, structural, explicit and background evidence used to determine the learner's knowledge state.

Implicit evidence, which is gleaned from the player's behaviour when playing the game, is determined by comparing the merits of the player's move with those of the available alternatives as defined by what the expert would do. Structural evidence is obtained from the syllabus, a network linking skills in terms of their complexity and dependencies, and suggests which skills are most likely to be acquired next by the player. This is determined by the belief that a student familiar with a certain region of the syllabus is more likely to learn skills on the frontier of this region. The dialogue between tutor and player, created by test cases and followup questions, provides explicit evidence. If it is not certain whether a skill has been mastered or not, the tutor can retest the player on a specific problem and gather as much explicit evidence as is required. Background evidence provides estimates of how "average" players of various backgrounds can be expected to perform. A player classifies himself/herself by answering a questionnaire provided at the beginning of the first session. This classification provides a starting state for the overlay model.

The P rules, triggered by evidence, are responsible for the initialization of the overlay model and for the maintenance of the USED and APPROPRIATE variables. These variables are plotted as the axes of the P graph, shown in Figure 3.4, and are interpreted by the P critic.

P critic

The overlay model is also effected by the P critic which decides whether inconsistencies or discontinuities have occurred in the learner. Based on these decisions, the model's KNOWN variable (acquisition state of the skill) is maintained. Inconsistencies occur when P rules fail to model the learner properly, and discontinuities are the indication of a change in the player's knowledge state. The P critic modifies the KNOWN variable after examining the P graph.

The graph in Figure 3.4 indicates the learner consistently fails to use skill S until point X, whereafter there is consistent use of the skill. Soon after point X the KNOWN variable would be set equal to T. To make the decision of when a student has acquired a skill as realistic as possible, Carr and Goldstein built in tolerance levels for when to update the KNOWN variable, as illustrated in Figure 3.5. To modify the KNOWN variable, the P critic examines the slope (where slope = USED/APPROPRIATE) of the plotted skill. Carr and Goldstein use the following measures to divide the tolerance levels of Figure 3.5. A slope of 0 to 10 degrees indicates an absence of the

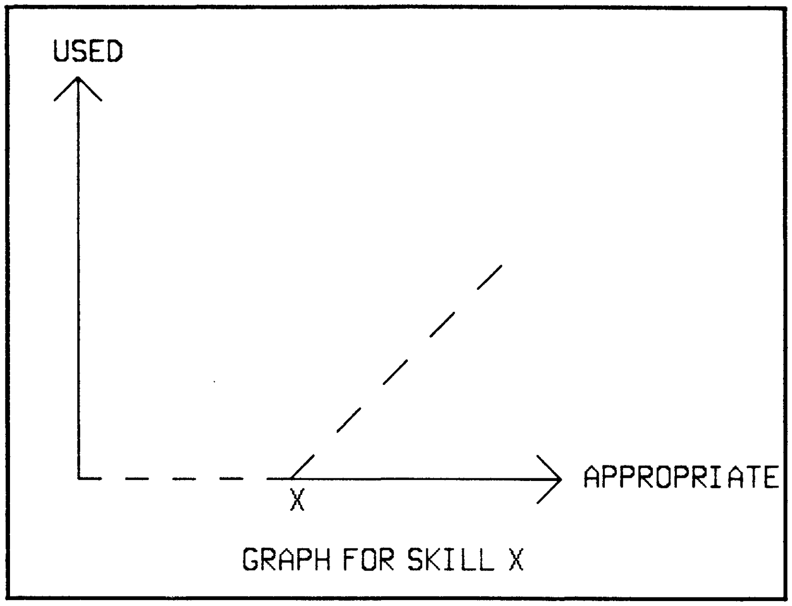


Figure 3.4 [12]

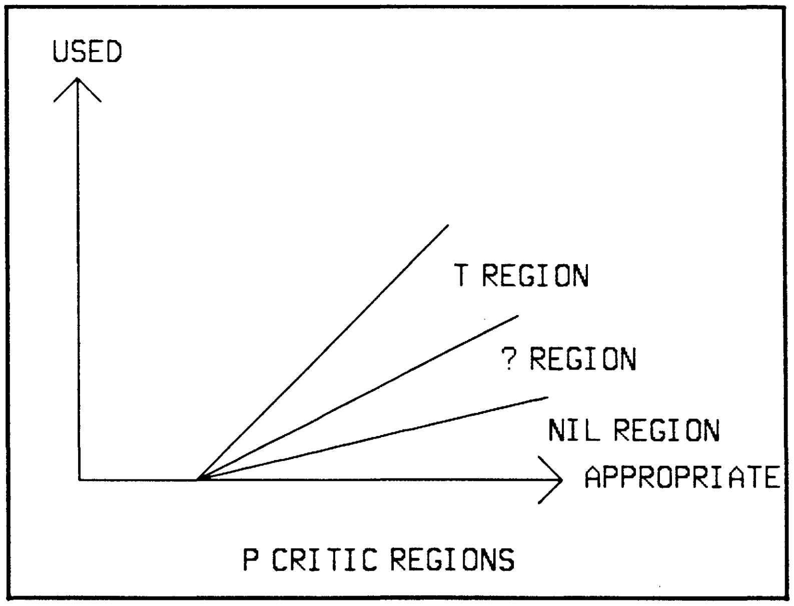


Figure 3.5 [12]

skill, hence KNOWN=NIL. A slope from 35 to 40 degrees means sufficient evidence has been gathered to support acquisition of this rule, then KNOWN=T. The interesting case arises when the slope falls within these two regions, that is KNOWN=?. This represents the uncertainty of whether or not a skill has been learned. This can occur either when a player is still acquiring a skill or when the P rules fail to model the player correctly. When this situation occurs, the tutor can prompt the player for explicit evidence. These regions of skill acquisition need to be studied further to determine if the tolerance divisions need to be adjusted.

Limitations

The most outstanding limitation of the overlay model is that the learner must possess a subset of the expert's skills. This is highly unlikely to occur during the learning phase. The learner's knowledge state will be less developed than that of the expert, hence he/she should not be expected to perform in exactly the same manner. Carr and Goldstein cite three reasons why this cannot be expected.

The first reason is that for any learning process there will be multiple paradigms. This means that there may be many different ways to acquire a particular skill. For instance, when teaching subtraction with borrowing, there are two methods, which are illustrated in Figure 3.6: decomposition and equal addition [37]. Here it is important that there be expertise to

$$\begin{array}{r} \overset{3}{\cancel{4}}\overset{10+}{7} \\ -\ 29 \\ \hline 18 \end{array} \qquad\qquad \begin{array}{r} 4\overset{10+}{7} \\ -\ {}^{3}\cancel{2}9 \\ \hline 18 \end{array}$$

Decomposition Equal Addition

Figure 3.6

support both methods. The suggested solution to this problem is to use multiple experts. A Meta-Expert is a set of experts for a particular domain, each with its own representation of the domain skills. The psychologist would then try to match a skill employed by the learner with an appropriate skill of one of the experts.

Another reason that the expert's skill choice may not coincide with that of the player is when the player purposely chooses a non-optimal move. For example, in the WUMPUS domain, the player may wish to take higher risks in order to finish the game faster. This is a difficult situation to model and perhaps an expert could be fashioned this way.

The last reason cited by Carr and Goldstein is that a player may possess an incorrect form of an expert's skill. This might occur when playing WUMPUS by applying the warning for bats and pits to the WUMPUS. The player recognizes the warning but incorrectly applies it. A partial solution was attempted by breaking rules down into "micro" skills. There is one set of rules for the bat and pit warnings (1 cave away) and another set for the WUMPUS warning (2 caves away). We recognize that it would be better to know typical bugs that arise in the learning of a skill and have the student model include these. This is a very important aspect of the student model in order to provide suggestions for a remedial program.

The overlay model will also have trouble representing extreme inconsistency on the part of the player and representing unrecognized expertise the player may possess. Two difficult situations were cited: complex verbal player explanations, that is natural language comprehension in the COACH (Figure 3.2), and the distinguishing of first order (absence of a skill) and second order (inappropriate use) bugs. Although these are interesting problems, they are not discussed further.

Comparison

In comparing the overlay model to the ideal model, it is evident that most criteria are not met. There are no means of representing misconceptions and corrections of expert skills and definitely no flexibility to the amount of detail contained in the student report. There is one set learning style, although improvements to this could be obtained by adding a Meta-Expert, and the model gives no indication of how the student progressed though the domain. As a result of these limitations, any remedial program, based upon the information in the student model, could not be tailored to the particular student in question.

3.4. BIP

BIP [35] employs a skill network to model a student's current knowledge state when learning computer programming. The skills are interrelated according to evolutionary relationships which include analogies, generalizations, specializations, prerequisite and relative difficulty relations (these relationships are discussed in Section 3.5). The model of the student includes those skills (a subset of the skill network) that the student does not possess, possibly possesses or definitely has acquired. Based on this student model, a set of skills that the student needs to learn can be determined and appropriate assignments can be developed to test when the student has learned the missing skills. Therefore, this method provides for incremental learning by the student. Instead of representing deviation links (as discussed later in Chapter 4), author-supplied exercises are attached to the appropriate skills in the network. The BIP system has some of the limitations of both the BUGGY system and the overlay approach; hence it is not discussed in detail.

3.5. The Genetic Graph

To overcome the overlay limitation of having the student possess a subset of the expert's skills, Goldstein [19] presents the genetic graph. The genetic graph is a framework for representing procedural skills (rules and facts) as nodes and evolutionary relationships as the links between the nodes. These relations can be analogies, simplifications, generalizations, deviations, corrections, refinements or specializations. Table 3.2 defines these links.

TABLE 3.2 [19]

1. R' is a *generalization* of R if R' is obtained from R by quantifying over some constant.
2. *Specialization* is the inverse.
3. R' is *analogous* to R if there exists a mapping from the constants of R' to the constants of R.
4. R' is a *refinement* of R if R' manipulates a subset of the data manipulated by R on the basis of some specialized property.
5. *Simplification* is the inverse.
6. R' is a *deviation* of R if R' has the same purpose as R but fails to fulfill that purpose in some circumstances.
7. *Correction* is the inverse.

Goldstein claims that this model supports a learner-based rather than expert-based paradigm. The genetic graph will not only provide a basis for a more accurate model of the learner, but will enhance the range of tutoring advice that the system can provide. Figure 3.7 shows the interaction of the genetic graph with the other components of the COACH. The student model is an overlay (Section 3.3) on the genetic graph instead of on the expert. This means that the student can possess skills that are built from the the expert's skills by the evolutionary links.

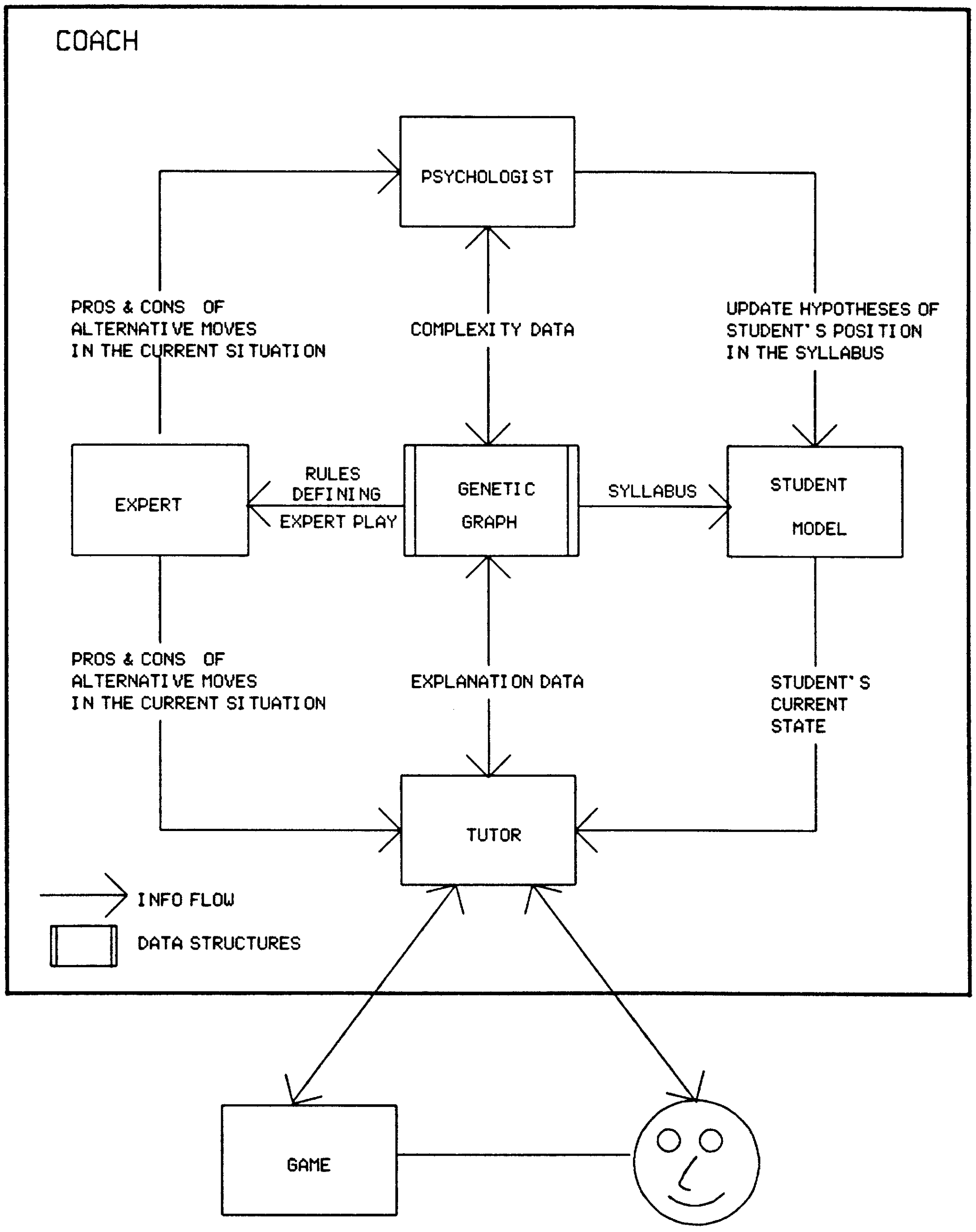
Domain Description

Goldstein uses the WUMPUS game, described in Section 3.3, because it presents an excellent domain that will enable the development of a theory of procedural skills.

COACH Development

WUSOR-I [32] was the first expert-based coach to use this domain. Rules such as the following [19] were used to analyze a player's move.

Positive Evidence:	*A warning implies that a danger is in a neighbouring cave.*


COACH
PSYCHOLOGIST
PROS & CONS OF ALTERNATIVE MOVES IN THE CURRENT SITUATION
COMPLEXITY DATA
UPDATE HYPOTHESES OF STUDENT'S POSITION IN THE SYLLABUS
EXPERT
RULES DEFINING EXPERT PLAY
GENETIC GRAPH
SYLLABUS
STUDENT MODEL
PROS & CONS OF ALTERNATIVE MOVES IN THE CURRENT SITUATION
EXPLANATION DATA
STUDENT'S CURRENT STATE
TUTOR
INFO FLOW
DATA STRUCTURES
GAME

Block Diagram of an AICAI tutor
Figure 3.7 [19]

Elimination:	***If a cave has a warning and all but one of its neighbours are known to be safe, then the danger is in the remaining neighbour.***
Multiple Evidence:	***Multiple warnings increase the likelihood that a given cave contains a danger.***

Simple rule explanations were provided to the player if its use would result in a better move than the move already chosen by the player. Goldstein states that WUSOR-I ignored the relative difficulty of the skills and hence WUSOR-II [13] resulted from a step toward an evolutionary epistemology. In WUSOR-II the rule set was divided into five increasingly more difficult skill levels [19]:

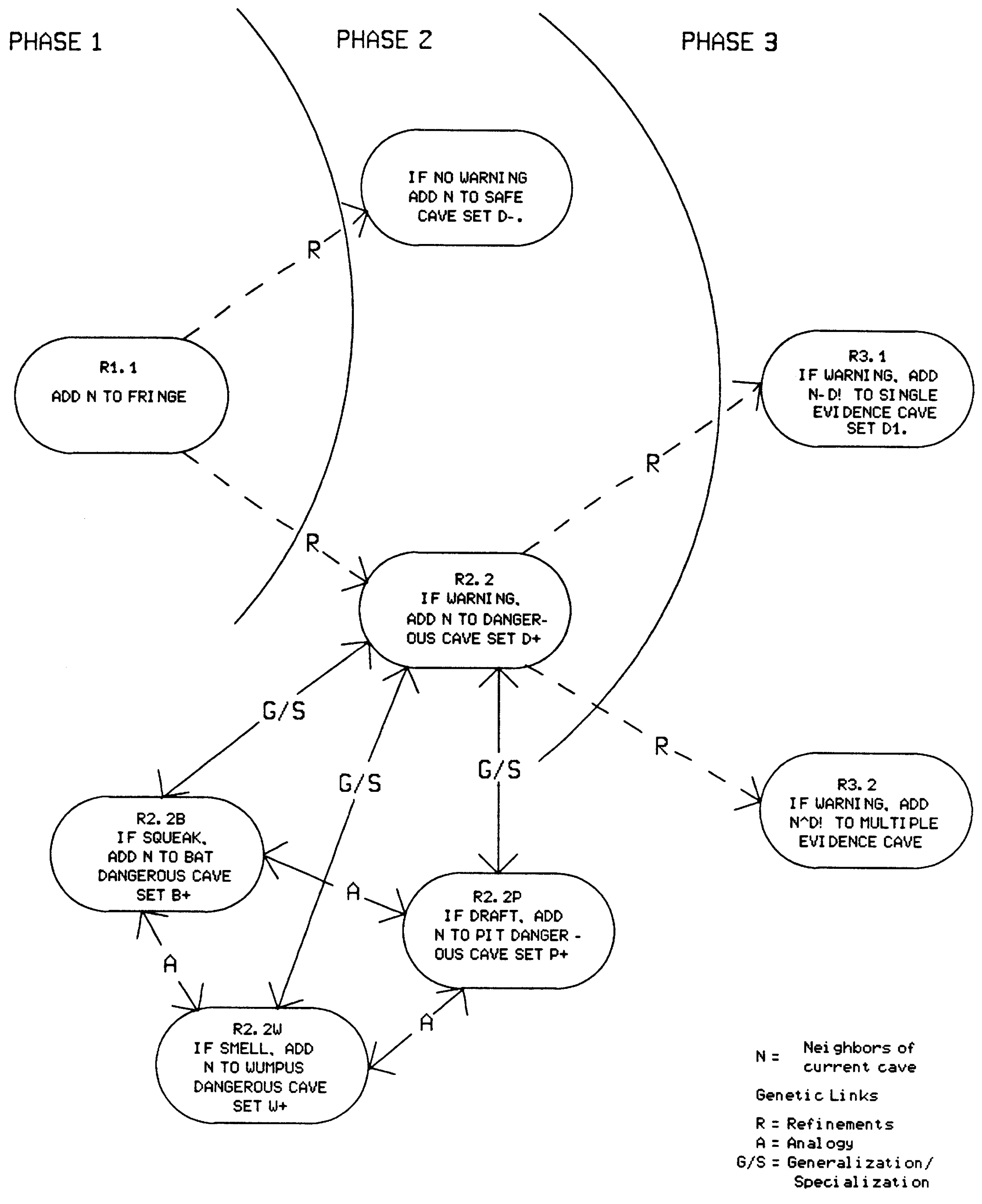
Phase 1:	***Rules for unvisited and fringe caves.***
Phase 2:	***Rules for possibly dangerous, definitely dangerous and safe caves.***
Phase 3:	***Rules for single versus multiply dangerous caves.***
Phase 4:	***Rules for "possibility sets", ie keeping track of the sources of dangers.***
Phase 5:	***Rules for numerical evidence.***

Rules of a new level of play were not described by the tutor until it was believed that all the rules of the previous levels had been acquired.

Still dissatisfied that enough information had not been provided for the student to build new knowledge from old, WUSOR-II evolved by defining a set of evolutionary relations between rules. Now the syllabus of the COACH is no longer an unordered skill set, but a genetic graph of skills linked by these relations.

The Genetic Links

The genetic graph in Figure 3.8 illustrates the use of the links defined in Table 3.2. In the WUMPUS game, refinements usually occur when advancing from one phase to the next. The advancement from recognizing a warning to distinguishing single or multiple warnings, is an example of refinement from R2.2 to R3.1 and R3.2. Generalizations/specializations can be seen in Figure 3.8 in the cluster of R2.2. Here a general warning, R2.2, is specialized into three specific cases, a squeak R2.2B, a draft, R2.2P, and a smell, R2.2W. When two or more nodes are specializations of one general node, they are usually interlinked by an analogous link. In this example, the three specific warnings are analogous by their similar dangerous nature. A deviation of a rule would occur if an incorrect rule was applied in the correct situation.


PHASE 1
PHASE 2
PHASE 3
IF NO WARNING ADD N TO SAFE CAVE SET D-.
R1.1 ADD N TO FRINGE
R3.1 IF WARNING, ADD N-D! TO SINGLE EVIDENCE CAVE SET D1.
R2.2 IF WARNING, ADD N TO DANGER-OUS CAVE SET D+
R3.2 IF WARNING, ADD N^D! TO MULTIPLE EVIDENCE CAVE
R2.2B IF SQUEAK, ADD N TO BAT DANGEROUS CAVE SET B+
R2.2P IF DRAFT, ADD N TO PIT DANGER-OUS CAVE SET P+
R2.2W IF SMELL, ADD N TO WUMPUS DANGEROUS CAVE SET W+
R
R
R
R
G/S
G/S
G/S
A
A
A
N = Neighbors of current cave
Genetic Links
R = Refinements
A = Analogy
G/S = Generalization/ Specialization

A region of the genetic graph

Figure 3.8 [19]

Testbed

The genetic graph designed for use in the WUMPUS domain contains about 100 rules and 300 links. The graph was explicitly defined, although it was planned to extend its use by making the graph expand dynamically.

ICAI systems in the past have allowed students to be defined as possessing only a subset of the expert's skills. The genetic graph overcomes this limitation by allowing one to represent a student's knowledge in terms of evolutionary relationships as well as representing a student as subset of the expert's skills. Goldstein cites that the relations he presents are incomplete and underspecified. For example, there are many different ways to obtain knowledge, such as learning by example, by being told, by induction or by deduction and the student's learning preference should be indicated within the student model. Because of these limitations, Goldstein suggests extensions to the described genetic graph.

Extended Genetic Graph

Goldstein proposes many extensions to this basic description of representing relations between rules. The first extension is to include ISLANDS of knowledge which are groups of rules with the same goal and that are generally learned together. He believes that to move from island to island requires a new conceptual base underlying a rule set.

The genetic graph also needs to include declarative knowledge and planning knowledge. Goldstein believes the same relationships linking procedural rules can be used in linking declarative statements. The genetic graph can easily include planning knowledge by adding prerequisite and postrequisite relations.

The extended genetic graph is a hierarchical structure of islands (groups of rules) with individual nodes being rules and declarative facts. The extended genetic graph also provides planning knowledge by extending the link set to include PRE and POST links. Figure 3.9 illustrates these extensions in a network of islands.

To incorporate the genetic graph into an ICAI system, the COACH (as defined in Goldstein's system) needs to be redesigned. The tutor needs to interpret the genetic graph in order to provide personalized instruction to each student by observing his/her current knowledge state and particular learning style.

Tutor Redesign

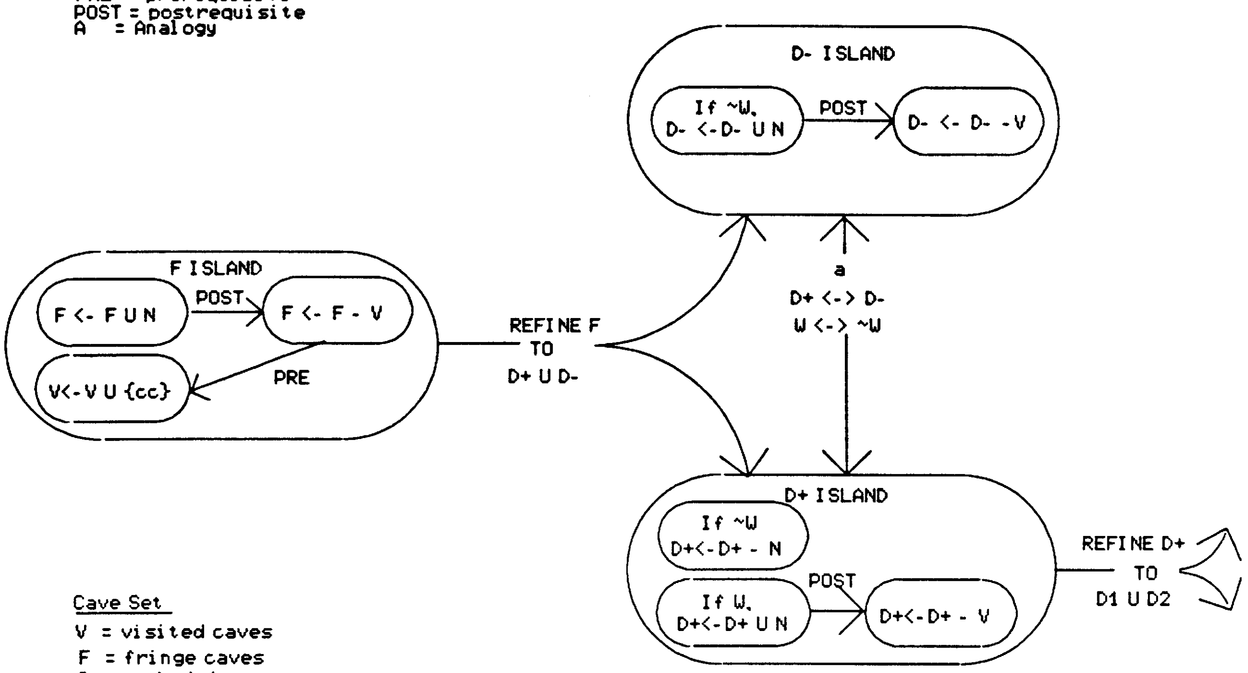
There are two types of information that the genetic graph supplies to the tutor. The skills to be taught next are those at the frontier of the student's individual genetic graph. Secondly the manner in which these new skills should be taught can be decided by observing the way in which the student

Game State

cc = the current cave
N = neighbours of the current cave
W = warning

Links

PRE = prerequisite
POST = postrequisite
A = Analogy



Cave Set

V = visited caves
F = fringe caves
D- = safe fringe caves
D+ = dangerous fringe caves
D1 = singly dangerous fringe caves
D2 = multiply dangerous fringe caves

The Extended Genetic Graph

Figure 3.9 [19]

reached his/her current position, that is the evolutionary path taken.

When deciding what to present next, script-based CAI systems have a rigid order in which to proceed through the domain. This could be a major hindrance to a student whose learning style is different from that of the expert. Expert-based CAI systems allow for domain exploration by the student. This allows for tutor intervention only when a non-optimal move has been chosen by the student. The disadvantage here is that there is no guidance, based on what the student already knows, as to when it is appropriate to introduce a new skill. The genetic graph solves both these problems; by presenting the skills with the highest priority (those on the frontier), the tutor must then decide which of these skills to teach or reject.

The tutor must make use of general teaching heuristics and specific strategies used by the student when deciding what to present next, or when to intervene. For instance, if a student appears to learn best by example, the tutor should proceed to teach a new skill by giving the student a number of examples that demonstrate that skill. Another aspect of tutoring is how a topic is explained. It is important to have multiple explanations available to correspond to the learning style of the student. The genetic graph has the role of increasing the available choices, by providing the ability to explain a new skill in terms of its genetic links, on which the teaching heuristics and student specific criteria operate. Each type of link has an explanation strategy associated with it.

The Student Model

The student model is very important when deciding what tutoring strategy is appropriate for a particular student. The genetic graph has three main features to aid the tutor. The nodes of the genetic graph provide a more refined structure for representing the student's knowledge state, the graph's organization provides a metric for determining which skills the student is most likely to obtain/acquire next, and the links of the graph provide a framework for a model of the student's learning behaviour.

Student models in script-based CAI systems only contain statistics on the correctness of the student's answers. Expert-based CAI systems provide more information in the form of hypotheses about which of the expert skills the student has acquired (recall Goldstein and Carr's overlay model). The genetic graph provides for a more sophisticated expert-based system. It provides a more detailed structure on which the student model is overlayed; the student no longer must possess a strict subset of the expert's skills.

The method of determining when a student has acquired a new skill, is the same as the method employed by Carr and Goldstein [12], described in section 3.3. The genetic graph provides improvements to this model by defining a number of "players" which correspond to the skill plateaus or levels in the graph. An example of a "player" in the WUMPUS game would be a set of skills which would be equivalent to a beginner, novice or expert player. Each "player" examines the student's move and decides which skills the student appears to be using. For each skill, the decision as to whether the skill has been obtained is based on the summation of all the hypotheses of the players. The genetic graph includes the theory that knowledge evolves along the genetic links, hence hypotheses provided by advanced players, those far from the current plateau, are assigned less weight.

A learning overlay can be obtained from the genetic links that provides a record of the effectiveness of an explanation strategy (based on the link type) on the student. This allows for a more personalized tutoring service for each student by selecting explanations that have had more effect in the past.

Limitations

Goldstein states that his initial research concerning the genetic graph was largely exploratory. The graph was implemented with approximately 100 rules and 300 links. Goldstein planned to have the COACH, WUSOR-II, converted to incorporate the genetic graph, however it appears that this has not been done. The evolutionary relations (refinements, generalizations, etc) are insufficient. As the graph exists, all nodes and links must be predetermined. Goldstein states that if procedures for generating common deviations and specializations were available, the number of rules and links could be greatly reduced. Hence the next stage in the genetic graph development is to include dynamic expansion of the graph. Also, to make the genetic graph more attractive, all extensions mentioned previously should be included.

Research

Once the initial genetic graph is implemented, Goldstein proposes many research questions that must be answered in order to enhance the original design. The first is to find out if it is possible to generate new rules using the genetic links. How are profitable analogies, generalizations and refinements determined. Another extension could be to expand the skill nodes into more primitive networks of conditions and conjunction nodes similar to those in BUGGY and BIP-II.

Other questions include the following: how effective are ISLANDS, do they really help, are they feasible, do they add relevant information to the graph? Is the belief that dense clusters are more easily learned than sparse clusters true? Research is needed into techniques for including belief measures for when a student has acquired a skill. How are incorrect applications of correct rules detected? In what specific areas could this type of genetic graph be useful? Is it too general? What methods should be used to determine and represent learning by being told, by induction from past examples, and deduction from old rules? Some of these questions are explored in later chapters.

Comparison

The genetic graph appears to meet most criteria of the ideal system. The nodes allow for representation of facts, declarative and procedural knowledge and the links for misconceptions and deviations from the expert's rules. The graph appears to be flexible to support ICAI in many different domains and this will be shown in Chapter 5.

The genetic graph provides information regarding a student's learning behaviour by presenting his/her path through the domain and indicating which explanation strategies worked best. A remedial program could be developed for the student based on information in the student model gained as an overlay of the genetic graph.

Chapter 4
The Modified Genetic Graph

After examining the existing student models in Chapter 3, the genetic graph approach [19] appears to provide a sufficiently flexible framework by which the ideal student model and ICAI system can be obtained. However, when testing the feasibility of this design, by modelling two diverse domains, several changes and additions to Goldstein's model had to be incorporated. We refine the model by extending the concept of a node and including new links to satisfy situations not covered by those defined by Goldstein. We discuss changes to the psychologist component and provide insights into the automatic generation of the genetic graph.

4.1. The Genetic Graph and the Student Model

The genetic graph is like a search space that represents all possible information that could ever be required. The student model is developed dynamically so that only that part of the genetic graph that is relevant for a particular student at a particular time exists. This is similar to the relation a proof has to the search space; the proof can be viewed as a subset of the search space.

4.2. The Nodes

The nodes of the genetic graph represent the facts, rules or procedures which describe a skill. This is straightforward when a simple skill is being described and Goldstein's [19] description is sufficient. However, a skill could also be made up of components or one may need to indicate parameters associated with a skill and hence a node. Therefore, additional specifications are necessary. We considered two options for representing components; have the components indicated within the nodes itself, or else separate the node into several nodes. After examination, it was felt that the second approach was more appropriate as it allowed for more flexibility. For example, separating each component into individual nodes allows for easy handling of deviations of each component and progress through the group of components can be marked on each separate node. This approach is illustrated in the subtraction and ballet domains in Chapter 5 and in Figure 4.1.

The problem of how to represent parameters within a node should be settled when deciding upon an appropriate knowledge representation scheme, and therefore is not addressed herein. Parameters are needed, for example, with procedural skills, demonstrated in the subtraction domain in Chapter 5 (in particular, see Figure 5.7).

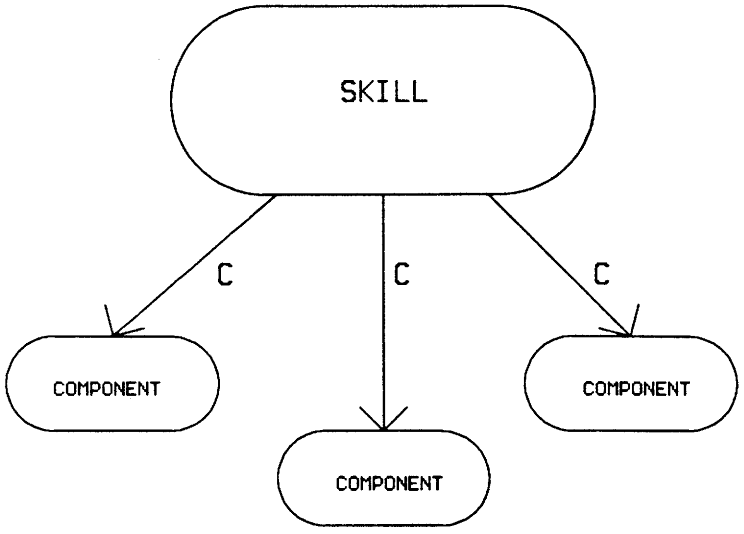
4.3. The Links

The links that are used in the modified genetic graph include those defined by Goldstein (Table 3.2) and the additional ones described in this section. The first situation which requires a new link is when the components of a skill are represented as individual nodes. Hence the *component* link, **C**, is defined as follows:

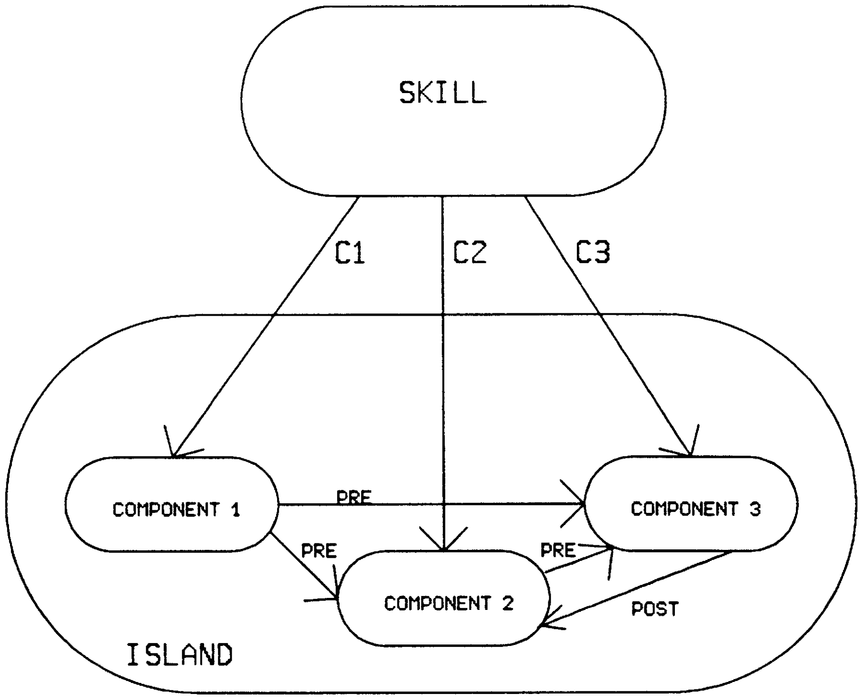
C: R' is a *component* of R if R' is a **necessary** element of R.

If a specific ordering of the components is required, the links can be numbered appropriately. This is a situation where Goldstein's concept of ISLANDS is applicable. The order of performing skills within an ISLAND is specified by the prerequisite and postrequisite links (see Section 3.5). Figure 4.1 illustrates these component situations.

Another situation which requires special attention is when a correction is involved. A correction link is used to indicate how to alleviate a deviation and is illustrated in Figure 4.2. Two variations of the correction link are needed. The correction link, **Corr1**, is used to connect a deviant skill and the corrected skill, where the corrected skill node is a different node than the correct node. This situation arises when the correction node is explicitly different than the skill node. Figure 4.2 (a) demonstrates this case. Figure 4.2 (b) illustrates the second correction link, **Corr2**. Somehow this

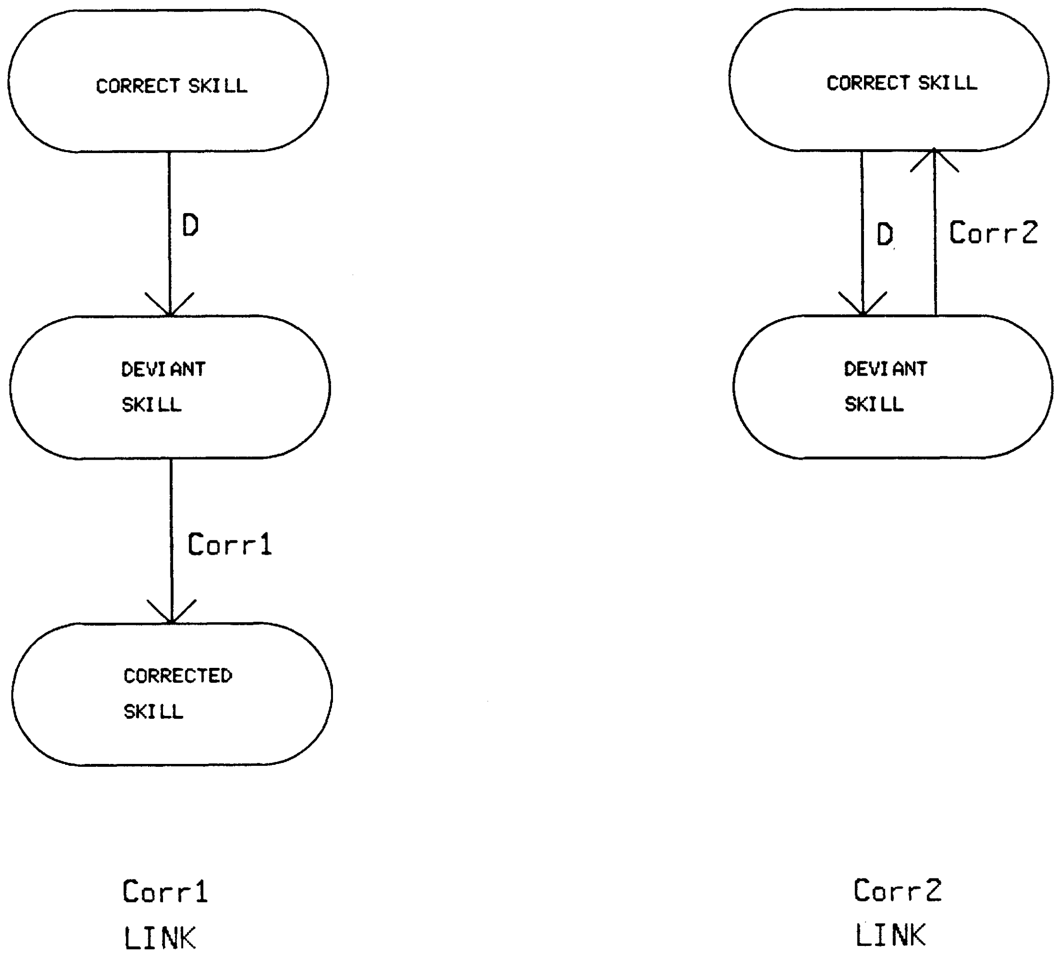


NO ORDERING



ORDERING

Figure 4.1



CORRECTION LINKS

Figure 4.2

information must be contained within the link which describes how the deviant skill must be corrected to obtain the correct skill. This situation is clearly illustrated within the ballet domain in Chapter 5, Figures 5.15, 5.16.

Another new link is a self-referencing link, **SR**, defined as follows:

SR: R' is connected to R by a *self-referencing* link if R' is a component of R and R' is the same operation or procedure as R, but with different parameters.

An example of when this link is needed is if the genetic graph is modelling a recursive algorithm, or if the parameters of a node change before the node is repeated; that is, it can be employed to represent loops or cycles. The later case is illustrated in the subtraction domain in Chapter 5, Figure 5.7.

Other links that may be necessary additions are *test* links, links that represent *learning preferences* or *learning situations* (learning by example, doing, analogy, discovery). These are not explicitly shown in Chapter 5 but can be incorporated if the need arises and should be considered when deciding upon an appropriate knowledge representation scheme.

4.4. The Psychologist

The genetic graph is a skeleton framework which is expanded and discarded dynamically by the psychologist throughout an ICAI session. The course graph and expert knowledge base are used by the psychologist to generate the initial genetic graph. A general course outline for teaching the material and information about prerequisite and postrequisite skills and knowledge are included in the course graph. The expert knowledge base contains what each node of the course graph represents. The course graph, which traditionally has been an AND/OR graph, could be included in either the expert, the psychologist or the instructional component.

Once the initial genetic graph is established, it must be maintained by the psychologist through analysis of the student's performance. The expert component provides the correct skill information and the psychologist may contain additional analysis information by means of its own expert (recall Chapter 2). The complex issues of how this is done are not addressed here, but the claim is made that with all this information, the psychologist can generate the necessary pieces of the genetic graph on which the student model will be overlayed.

4.5. Automatic Generation

The genetic graph in WUSOR-II is static. This is a major disadvantage because the entire genetic graph must be predetermined and tracing the student's progress is limited to the initial static genetic graph. For a large domain, this is an unrealistic approach. To alleviate this problem, a dynamic structure in which only necessary nodes are maintained, is required. This maintenance involves adding new nodes as new skills (and variants thereof) are acquired, updating current nodes for which new information is available, and removing old nodes which are no longer relevant to the student model. This can be done by marking those nodes which are relevant at any given time.

In order for the psychologist to maintain the genetic graph, there must be a means of automatically generating possible misconceptions, deviations, analogies, specializations, refinements or generalizations of correct knowledge or procedures. These are major issues that are currently being addressed by AI research. Chapter 5 presents an approach for generating deviations of a procedural representation for subtraction. For further discussions regarding relevant research, see [26], [34].

Chapter 5

Feasibility of the Genetic Graph

5.1. Domain Choice

Most student model structures developed to date are domain specific. This is an incredible weakness in the design. It is desirable to have a structure that is general enough to use in various domains. This chapter shows that the genetic graph approach is both feasible and appropriate in a variety of domains. In so doing, we illustrate the extensions described in Chapter 4.

When selecting domains, several requirements were considered. First of all, a domain that has been researched and implemented in another structure is ideal as it provides a basis for comparison. Hence, the domain of subtraction was selected. The genetic graph is then shown to be applicable in this domain. Another domain that was chosen and implemented, that of elementary ballet, is very different in topic and structure from subtraction. Hence, our proposed extended genetic graph model is effective to design a student model in a vastly different domain.

Furthermore, it is shown that the student model built from the genetic graph is very flexible in the amount of detail it includes. This is crucial as no two students/users progress in exactly the same way or need the same amount of detail, and few (if any) existing ICAI systems provide this feature.

5.2. The Chosen Domains

The first domain chosen was subtraction. Much research has been done in this area by Burton and Brown [6], [10] and Young and O'Shea [36]. Furthermore, there is an enormous amount of relevant educational research; for example, see [1]. Burton and Brown's BUGGY skill lattice represents the subskills of subtraction procedures which were found to indicate "bugs" or errors (Figure 3.1 Table 3.1). Young and O'Sheas's [36] production system (Section 5.5) presents an alternative, simpler approach to representing subtraction. A third interpretation (Section 5.5) for subtraction is a PROLOG design [22]. It is shown that the genetic graph can be used to represent all of these models. The fact that the genetic graph can incorporate multiple interpretations of subtraction is desirable because then no one is forced to view subtraction in one particular way.

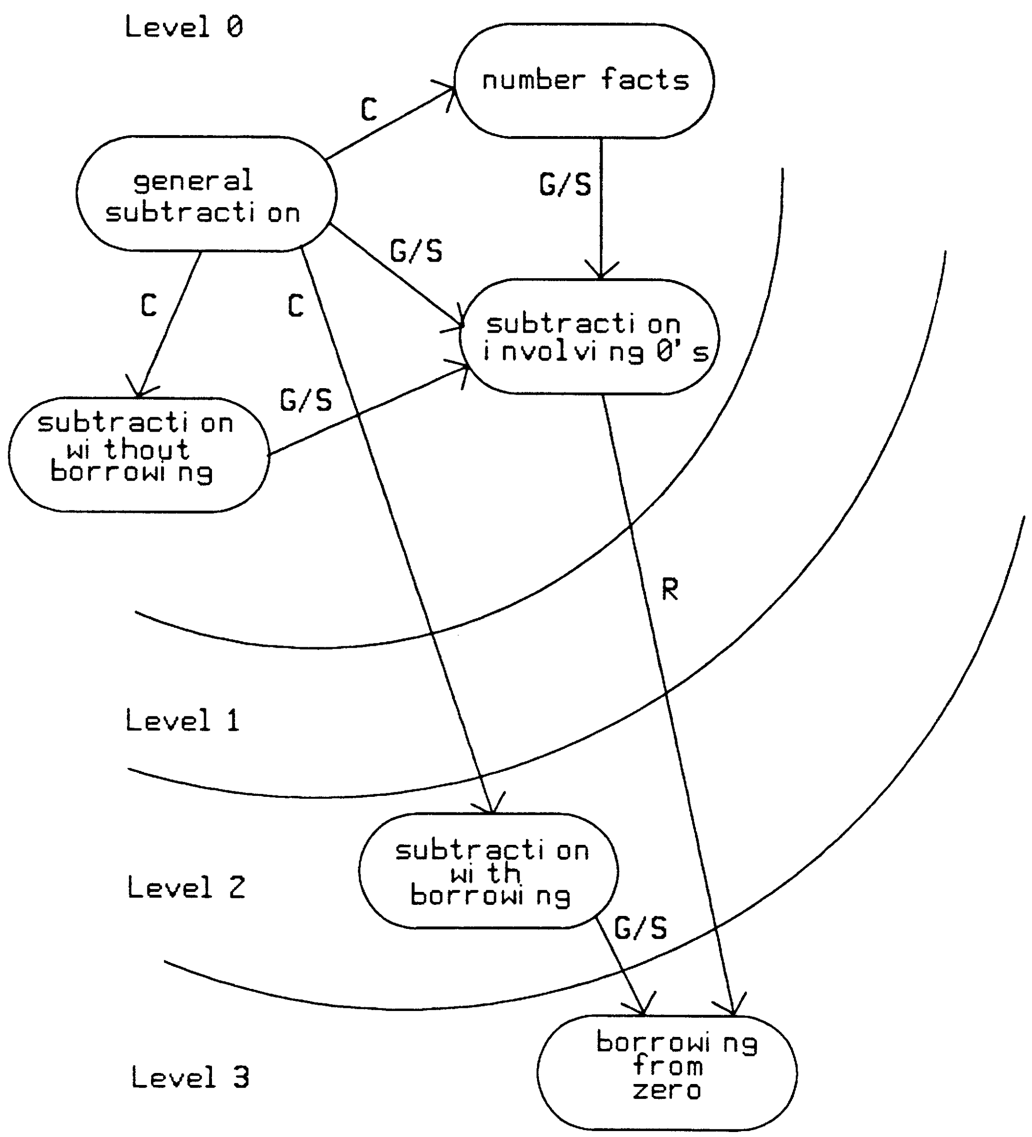
The second domain chosen is radically different from subtraction. The teaching of elementary ballet was chosen because of a personal interest and relative expertise in the area and also it fulfilled the diverse domain requirement. Because there are many teaching approaches to ballet, one well documented method was selected [24]. Relevant chapters of [24] are included in the Appendix. Other methodologies for teaching ballet could easily be incorporated.

5.3. Subtraction

The genetic graph for subtraction can be either extremely simple or relatively complex depending on the amount of desired detail. In the case of an advanced student, it might be sufficient to represent his/her skill knowledge with a very sparse graph. However, the graph should also allow sufficient detail for an in-depth educational diagnosis with enough information to develop a remedial program. The fact that there are several ways in which the information can be represented is an asset, and another illustration of the flexibility of this approach.

A Subtraction Genetic Graph

There are many different ways to interpret subtraction. Figure 5.1 presents an initial genetic graph of a superficial overview of subtraction. There are three components which constitute general subtraction; these are *number facts* (such as 7-3=4 or 12-0=12), *subtraction without borrowing* and *subtraction with borrowing*. The *borrowing from 0* node of level 3 is a refinement of level 0's *subtraction involving 0's* node and a specialization of the *subtraction with borrowing* node of level 2. This node is at a different



LEGEND
G/S = generalization/specialization
R/S = refinement/simplification
A = analogy
D = deviation
Corr1, Corr2 = correction
C = component
SR = self-referencing

A Genetic Graph for Subtraction

Figure 5.1

level since it is considered a more difficult task than borrowing from a non-zero number and many students have trouble learning to borrow from zero. For these reasons special attention is given to problems involving this skill. The information that zero is a special case is contained in the expert and it is the responsibility of the psychologist to access and expand upon this knowledge.

It is interesting to look at the link interaction with the level 0 node of *subtraction involving 0's*. This node is a specialization of the *general subtraction* node, the *number facts* node and the *subtraction without borrowing* node. All these nodes contain skills where the student encounters zeros.

This initial genetic graph can be expanded dynamically by the psychologist as the tutor and student progress through the course material. The remainder of this section extends the genetic graph of Figure 5.1 to include each subtraction subskill of the BUGGY skill lattice (Figure 3.1, Table 3.1).

Level 0

To incorporate those BUGGY subskills which entail general subtraction skills into the genetic graph, level 0 is needed. The details of this level are shown in the genetic graph of Figure 5.2. The BUGGY subskills 2,3,4 and 60 are all considered basic number facts. For example, subskill 60 states:

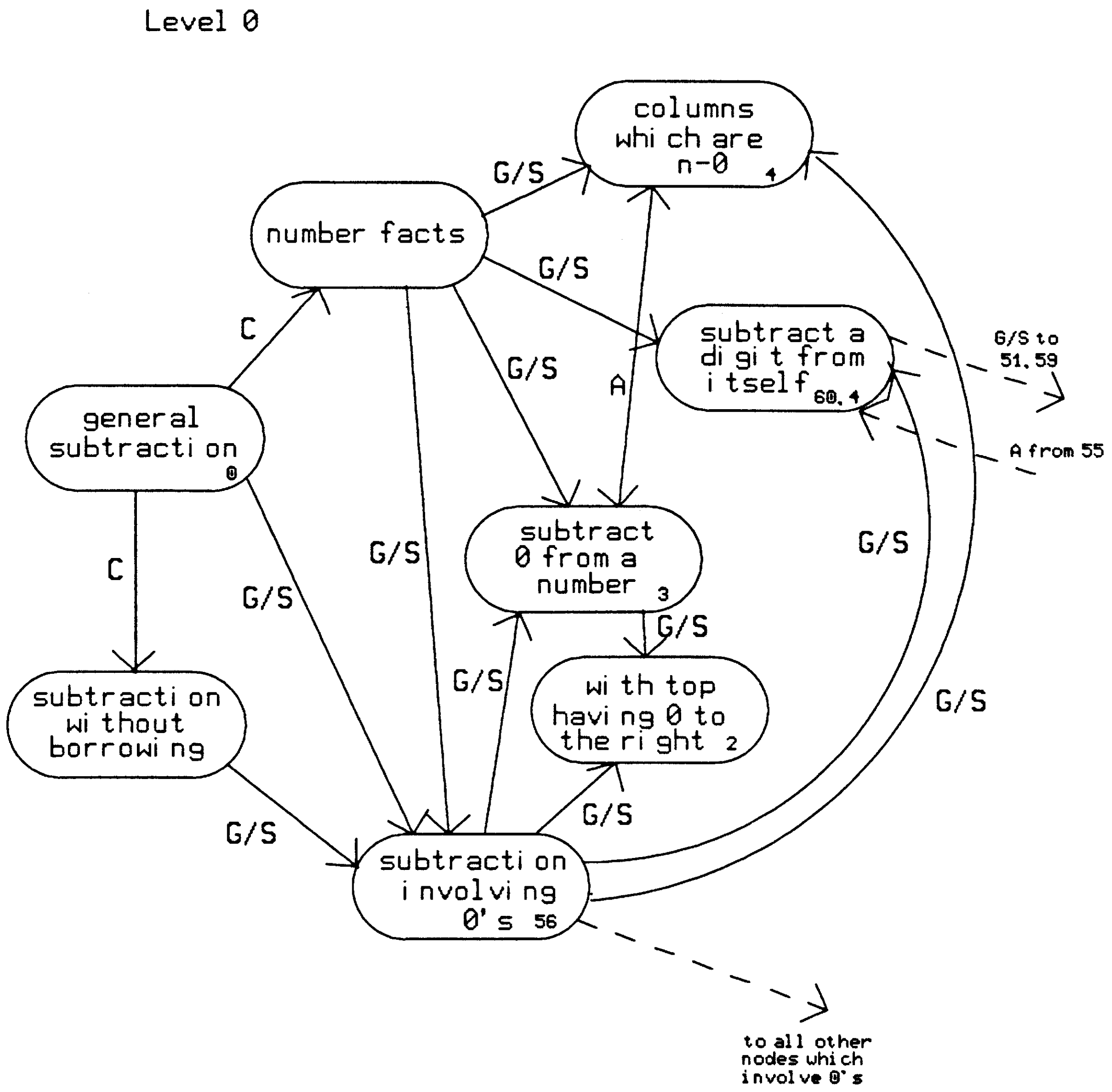

Level 0
columns which are n-0 4
number facts
subtract a digit from itself 60, 4
general subtraction 0
subtract 0 from a number 3
with top having 0 to the right 2
subtraction without borrowing
subtraction involving 0's 56
G/S
C
A
G/S to 51, 59
A from 55
to all other nodes which involve 0's

Figure 5.2

subtract columns that have top and bottom digits the same. This is the same as the subtraction skill of knowing **n-n=0** and hence, is a number fact. Any skill which involves 0's is a special case of the ***subtraction involving 0's*** node and is connected to it by a generalization/specialization link. For example, BUGGY subskills 2,3,4 and 60 all somehow involve 0's, hence are special cases of this node.

The node representing subskill 56 (**subtract numbers with zeros in them**) could be expanded into three nodes as illustrated in Figure 5.3. For some children all three nodes will be needed and in other cases where all three specialization are mastered, only node 56 is needed. In the later situation, mark node 56 as mastered and then this portion of the genetic graph will not have to be expanded. All other nodes would be linked to either node 56 or the three specialized nodes depending on which version is included in the graph.

Two analogies surface in this level: the subtraction of digits **n-0** is analogous to the subtraction of numbers **N-0** and similarly, **n-n** is analogous to the level 1 node which contains **N-N**.

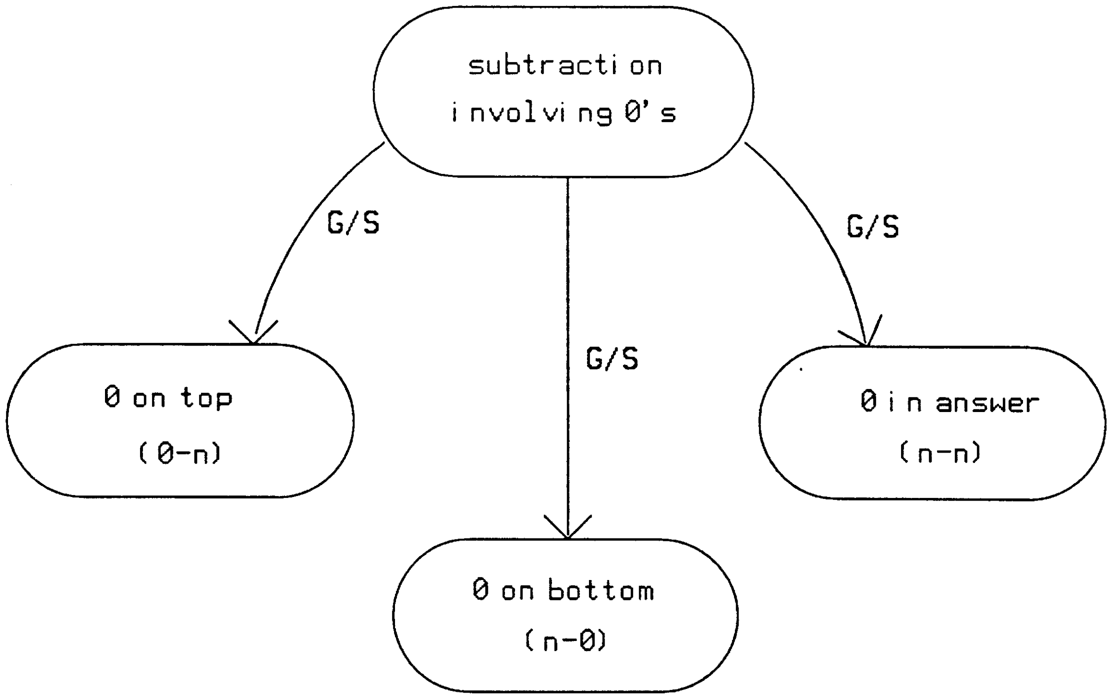

subtraction involving 0's
G/S
G/S
G/S
0 on top (0-n)
0 on bottom (n-0)
0 in answer (n-n)

Figure 5.3

Level 1

For many domain representations, the amount of specific information shown is Figure 5.4 may not be necessary. However, if the diagnosing of arithmetic difficulties is taking place, immense amounts of detailed information are required. Hence, level 1 may not be necessary for the "average" student, but it is essential that the genetic graph is expandable into this amount of information when it is required. For these reasons, level 1 is a more specific look at the ***subtraction without borrowing*** skill.

One might ask whether a central node is necessary in level 1. This version of the genetic graph does not contain a central node because it is felt that the inclusion of such a node is redundant. However, if the inclusion of a central node would lead to a clearer understanding or seem more logical, such a node can be incorporated.

Level 1 includes the BUGGY subskills numbered 42,46,47,50-55, and 57-59, all of which represent specific situations which arise when no borrowing is necessary. For example, there is a special situation of **n-n** when subtraction is taking place in the leftmost column. In this case a blank is written in the answer instead of a 0. Another interesting occurrence in level 1 is the ***0 in the top which is not the result of having been borrowed from*** node. This is also a case where a 0 may or may not be borrowed from in future steps, hence there is a generalization/specialization link to it from the

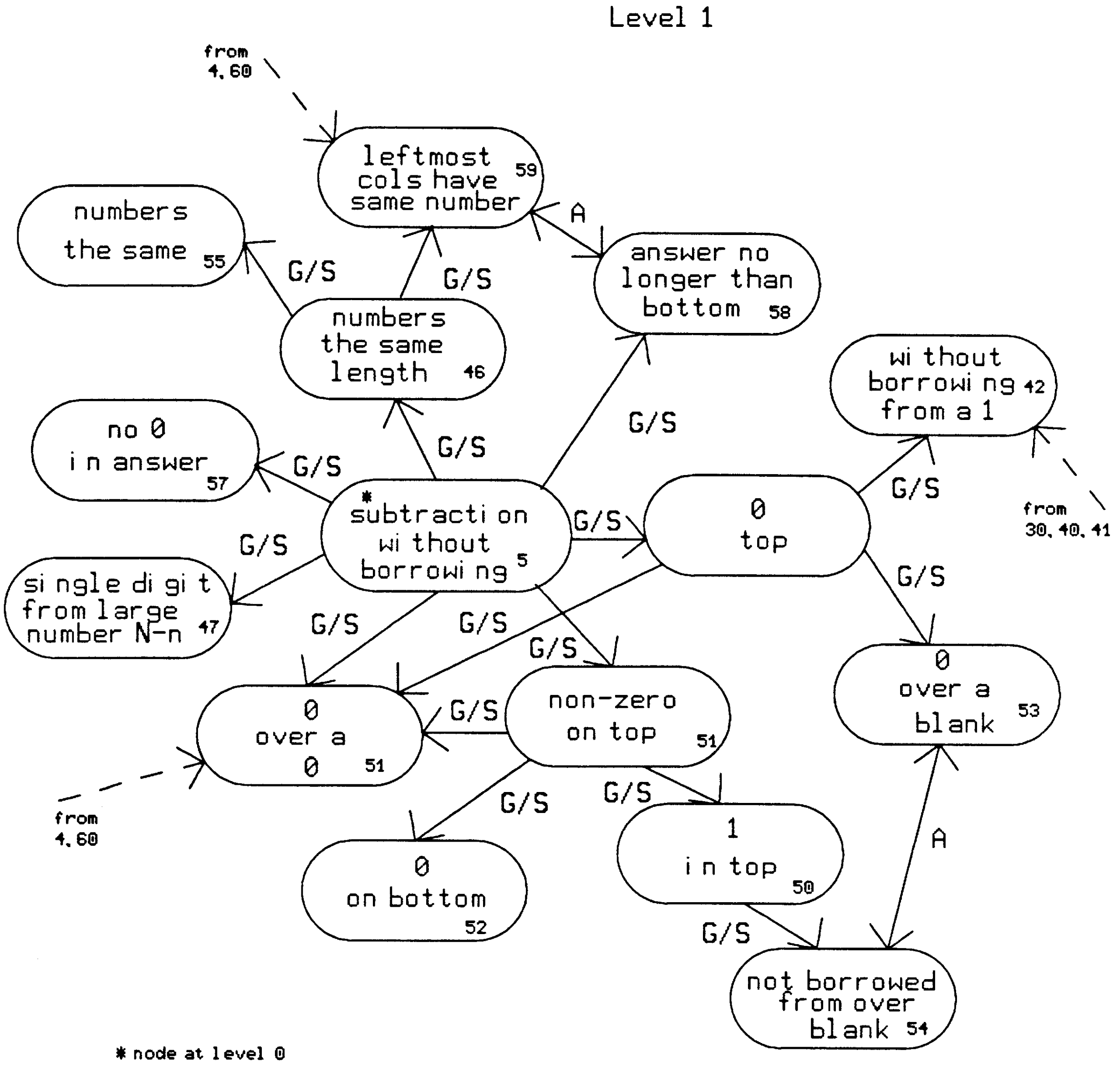

Level 1
from 4, 60
leftmost cols have same number 59
numbers the same 55
answer no longer than bottom 58
numbers the same length 46
without borrowing from a 1 42
from 30, 40, 41
no 0 in answer 57
* subtraction without borrowing 5
0 top
single digit from large number N-n 47
0 over a 0 51
non-zero on top 51
0 over a blank 53
from 4, 60
1 in top 50
0 on bottom 52
not borrowed from over blank 54
G/S
A
* node at level 0

Figure 5.4

borrowing involving 0 node of level 3.

Level 2

The fact that many students have trouble learning to borrow is evident from the complex genetic graph of Figure 5.5. For this reason, there are many BUGGY subskills concerning subtraction which involves borrowing. When analyzing the genetic graph for this level, remember it is not the only possible representation.

There are two components of the *subtraction with borrowing* node: *decrement top to left by 1* and *add 10 to top of column working on*. This is the situation discussed in Chapter 4 of how to represent components of a skill. These components are additions to the BUGGY subskills, but are included to demonstrate the interaction of debugging subskills and the components of the skills being taught.

There are many ways in which the nodes of this level can be interlinked. For example, look at the cluster of (27,49), (15,29) and (9,28,29). The generalization/specialization link from (9,28,29) to (15,29) may not be necessary, but because the subskill 29 includes both, the link is there.

The *borrow from leftmost column* node has three special cases. These nodes are all analogous because of the common characteristic of being special cases of left column subtraction.

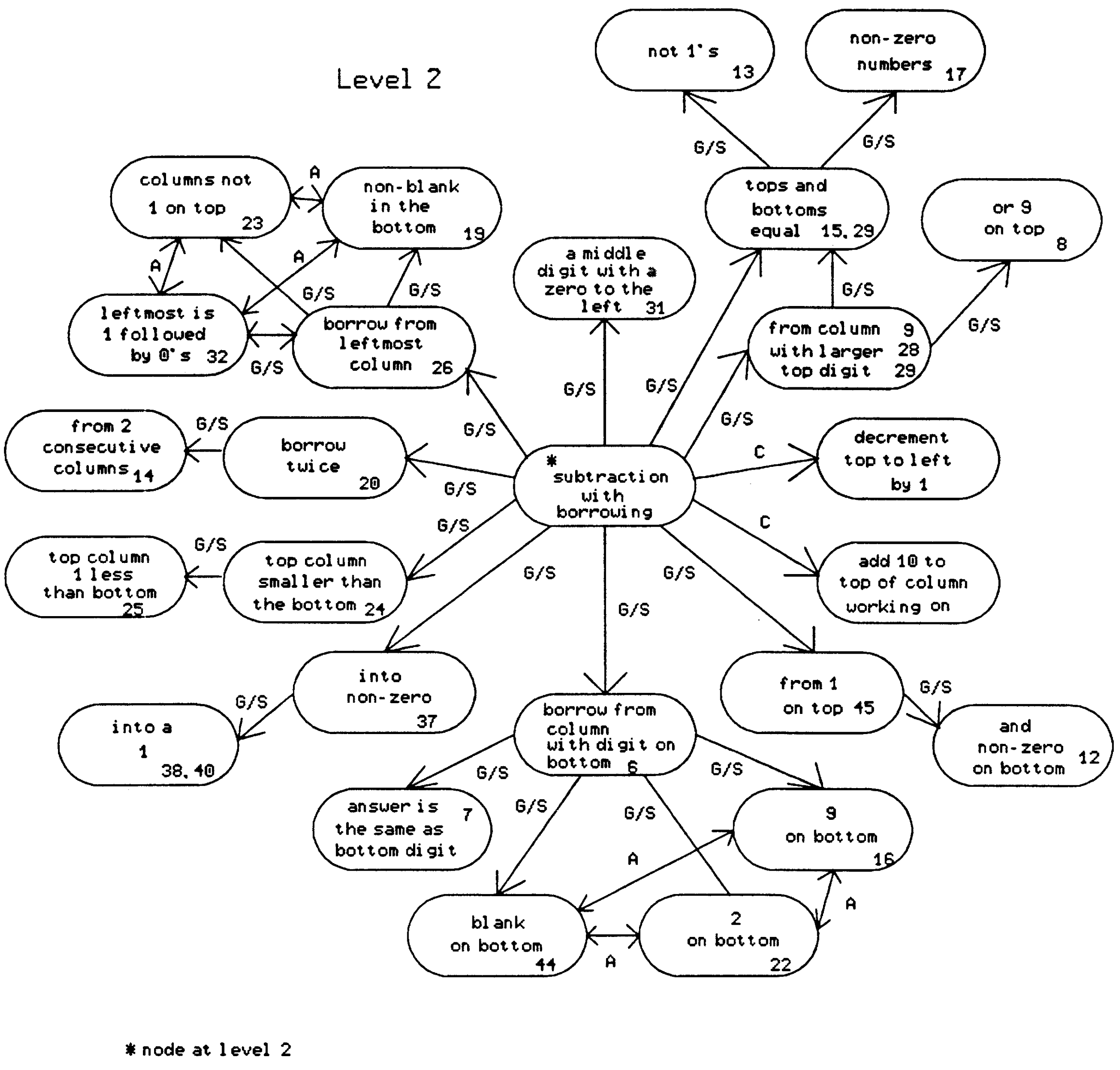

Level 2
not 1's 13
non-zero numbers 17
G/S
G/S
columns not 1 on top 23
A
non-blank in the bottom 19
tops and bottoms equal 15, 29
or 9 on top 8
A
A
a middle digit with a zero to the left 31
G/S
G/S
G/S
leftmost is 1 followed by 0's 32
borrow from leftmost column 26
G/S
from column with larger top digit 9 28 29
G/S
G/S
G/S
G/S
G/S
from 2 consecutive columns 14
G/S
borrow twice 20
G/S
* subtraction with borrowing
C
decrement top to left by 1
G/S
top column 1 less than bottom 25
G/S
top column smaller than the bottom 24
G/S
C
add 10 to top of column working on
G/S
G/S
G/S
into non-zero 37
into a 1 38, 40
G/S
from 1 on top 45
G/S
and non-zero on bottom 12
borrow from column with digit on bottom 6
G/S
G/S
answer is the same as bottom digit 7
G/S
G/S
9 on bottom 16
A
A
blank on bottom 44
2 on bottom 22
A
* node at level 2

Figure 5.5

Level 3

The genetic graph of level 3, illustrated in Figure 5.6, expands the *borrowing including 0* node. All nodes in this level are special cases or components of the central node. The case of when there is a *0 on top and a 0 to the left of it* is a very special situation. This node is in turn refined into *borrowing from* or *borrowing into* nodes (BUGGY subskills 33,34,35 and 43).

Two components, *0 becomes 9* and *to next column to borrow*, are the two steps necessary when *borrowing from 0*. These are required skills but are not subskills of the BUGGY skill lattice.

Conclusions

The BUGGY subskills are correct skills that have been recognized as portions of skills that might be used incorrectly. Nowhere in the BUGGY system is the operation of subtraction defined. The genetic graph of Figure 5.1 is one interpretation of subtraction.

Assuming the expert contains knowledge about the correct procedure for subtraction and the psychologist includes the diagnostic components of the BUGGY system, the initial genetic graph has been shown to be expandable to include almost all BUGGY subskills. There are a few subskills (18,21,39) that were not included because their interpretation is unclear. The

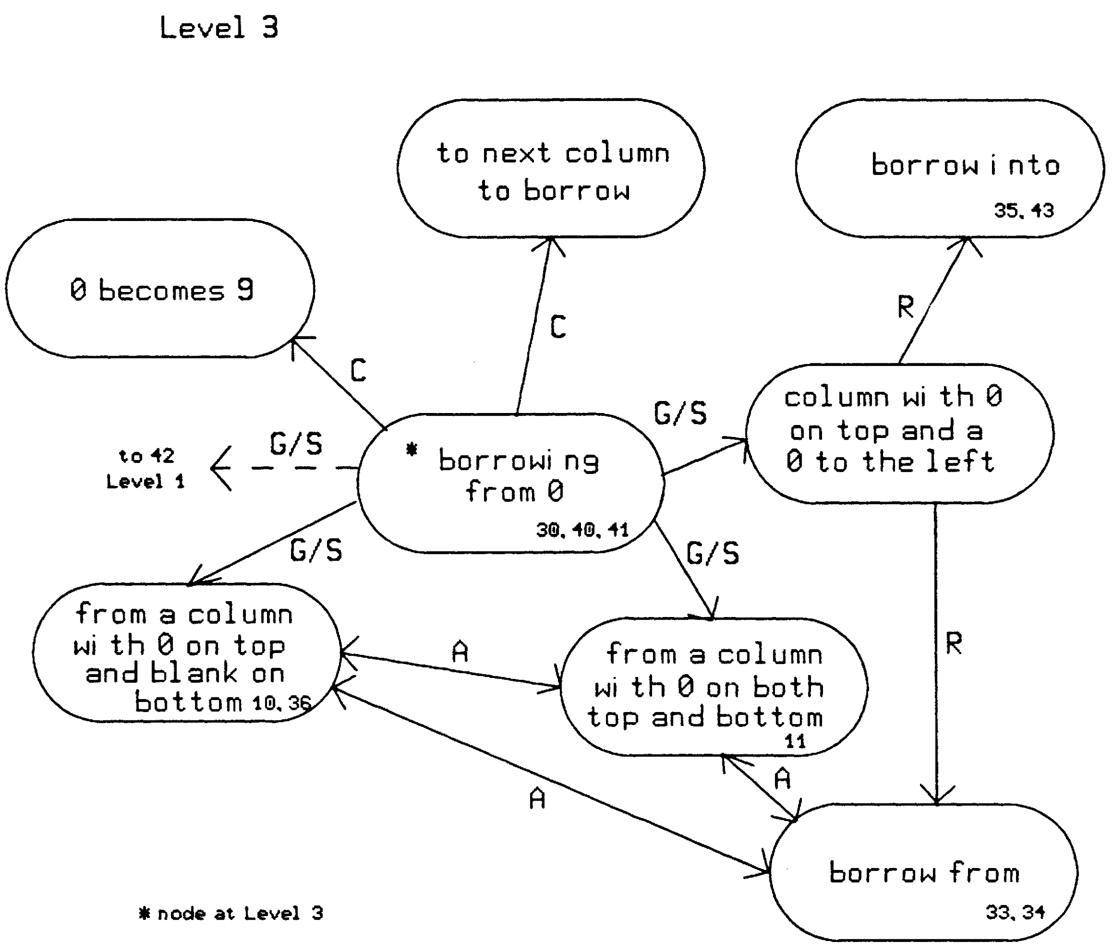

Level 3
to next column to borrow
borrow into 35, 43
0 becomes 9
C
C
R
G/S
to 42 Level 1
G/S
* borrowing from 0 30, 40, 41
column with 0 on top and a 0 to the left
G/S
G/S
from a column with 0 on top and blank on bottom 10, 36
A
from a column with 0 on both top and bottom 11
R
A
A
borrow from 33, 34
* node at Level 3

Figure 5. 6

interaction between the correct subtraction interpretation and the BUGGY subskills is shown when all levels (Figures 5.1-5.6) are combined.

5.4. Alternative Approaches to Subtraction

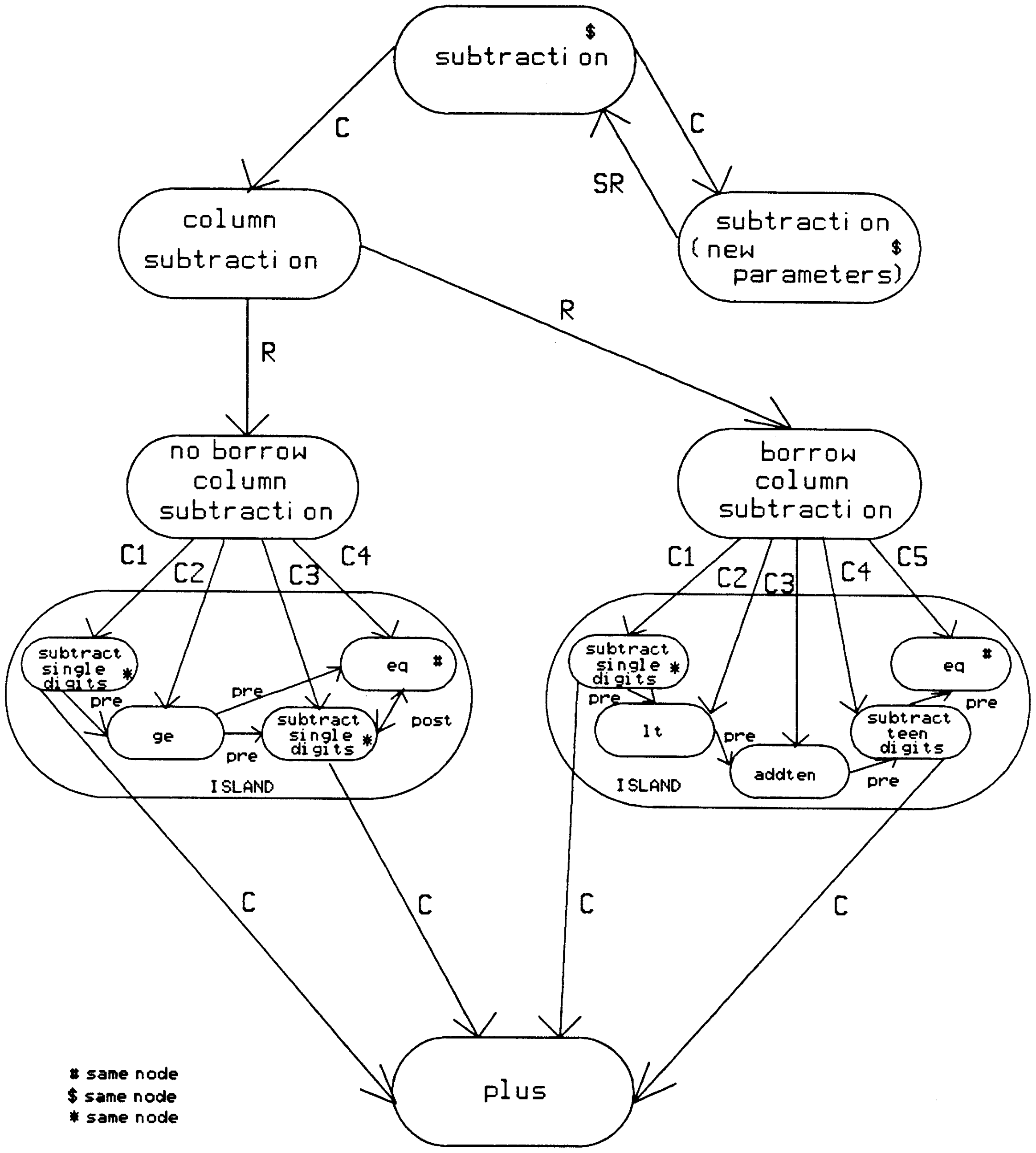
This section illustrates the flexibility of the genetic graph by representing two different subtraction interpretations. The two alternative approaches to subtraction are a PROLOG description from [22] and the production rules employed by Young and O'Shea [37]. The following PROLOG description is represented as a genetic graph in Figure 5.7.

```
subtraction ([right-minuend | rest-of-minuend] [right-subtrahend | rest-of-subtrahend]
    [right-answer | rest-of-answer] prev-borrow) <-
    column-subtraction (prev-borrow right-minuend right-subtrahend right-answer
    borrow-required)
    subtraction (rest-of-minuend rest-of-subtrahend rest-of-answer borrow-required);

column-subtraction (prev-borrow minuend subtrahend answer borrow-required) <-
    subtract-single-digits (minuend prev-borrow minuend)
    ge (minuend subtrahend)
    subtract-single-digits (minuend subtrahend answer)
    eq (borrow-required 0);

column-subtraction (prev-borrow minuend subtrahend answer borrow-required) <-
    subtract-single-digits (minuend prev-borrow minuend)
    lt (minuend subtrahend)
    addten (minuend minuend)
    subtract-teen-digits (minuend subtrahend answer)
    eq (borrow-required 1);

subtract-single-digits (minuend-digit subtrahend-digit answer-digit) <-
    plus (answer-digit subtrahend-digit minuend-digit);
```



PROLOG Gentic Graph
Figure 5.7

```
subtract-teen-digits (minuend-teen subtrahend-digit answer-digit) <-
    plus (answer-digit subtrahend-digit minuend-teen);

addten ( x x+10) <- plus (x 10 x+10);
```

The genetic graph in Figure 5.7 presents some new arrangements. First of all it is necessary, in this application, to indicate parameters of a node. How they are to be represented is a decision which will be made when deciding what method of knowledge representation is to be used, and is beyond the scope of this paper. However, it is an important decision since this approach includes an instance of self-referencing or recursive links which need to indicate changes in parameters of the linked nodes. The *subtraction* node is a component of itself with the parameters changes indicated to take place in the *subtraction** node. This recursion is shown by the self-referencing link between the two *subtraction* nodes.

Secondly, the components of the refined *column-subtraction* nodes require a specific ordering. This is handled by numbering the components. The numbered component links form a single concept which designate an ISLAND [19]. The required order of performance within ISLANDS is indicated by prerequisite and postrequisite links. Components C3 and C4 can be performed interchangeably and this is indicated by the links. This ISLAND information is required by the tutor so that these concepts can be taught as a group.

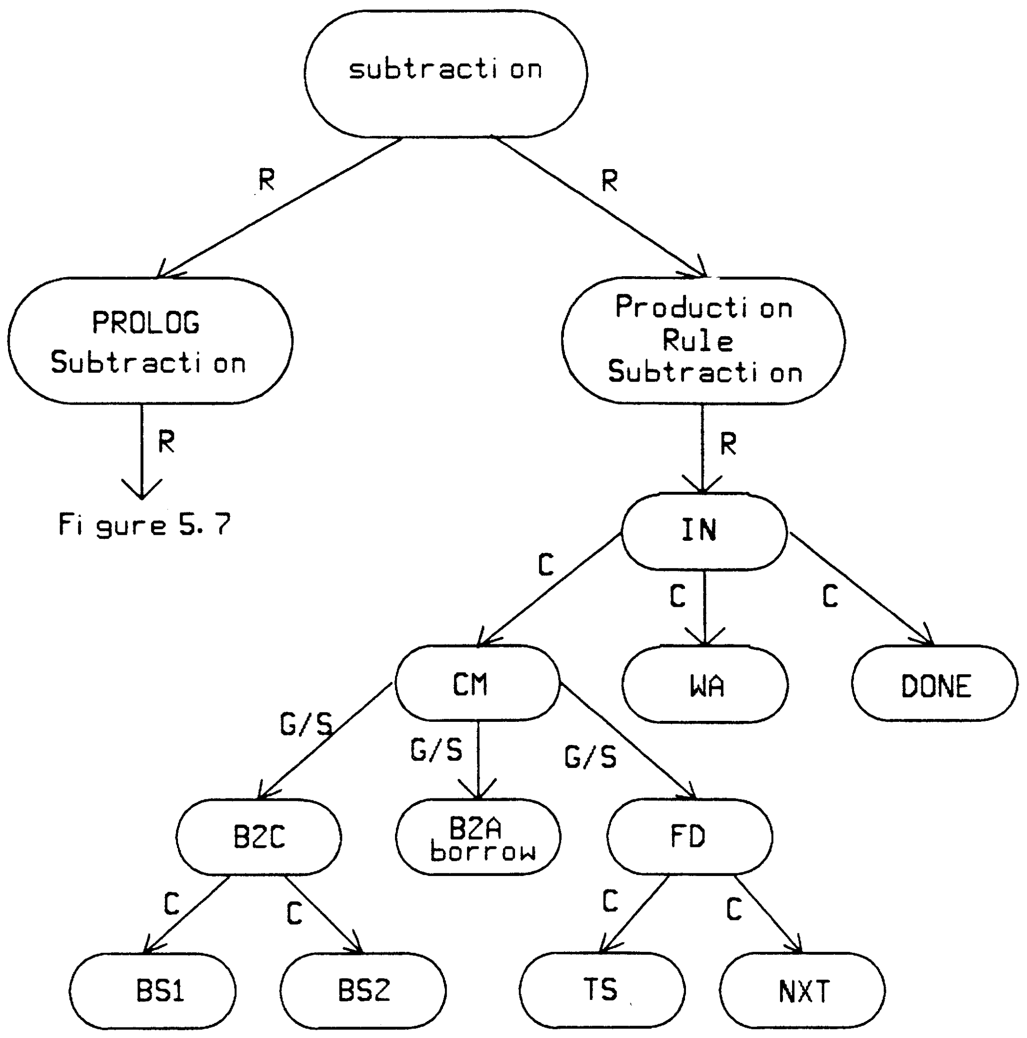
Two Style Representations

The genetic graph can also be used to incorporate two (or more) different approaches to subtraction. To illustrate this, Figure 5.8 shows subtraction with the PROLOG design as one methodology, and the following production system [37] as a second methodology.

FD:	M =m,S =s	==>		FindDiff, NextColumn
B2A:	S > m	==>		Borrow
BS1:	Borrow	==>	*	AddTenToM
BS2	Borrow	==>	*	Decrement
CM:	M =m,S =s	==>	*	Compare
IN:	ProcessColumn	==>	*	ReadMandS
TS:	FindDiff	==>	*	TakeAbsDiff
NXT:	NextColumn	==>	*	ShiftLeft, ProcessColumn
WA:	Result =x	==>	*	Write =x
DONE:	NoMore	==>	*	HALT
B2C:	S = M	==>		Result 0, NextColumn
AC:	Result 1 =x	==>	*	Carry, Result =x

Production system for subtraction by decomposition
* - actions

Each rule of the production system represents a component of a student's total subtraction ability. These rules can be independently present or absent in a pupil's knowledge state at any given time. For more details on this method, consult [37].



Two Subtraction Styles
Figure 5.8

5.5. Automatic Generation of the Genetic Graph

Another important aspect of the genetic graph that must be discussed is the feasibility of having procedures to automatically generate such a structure. Recall Section 5.4 which discussed the generation of the initial genetic graph and expressed the need to dynamically develop permutations or bugs of skills. This section illustrates how permutations to a procedural system could be automatically generated.

Consider a procedural description of subtraction such as, the PROLOG description given in Section 5.5 (which would reside in the expert), where the following perturbations are possible:

-omit, add or permute the ordering of the given steps

-test conditions incorrectly

-omit, add or permute the ordering of the given arguments for a given step

-substitute one argument for another.

The implementation of some of these perturbations such as "omitting a step", "permuting arguments" or "substituting one argument for another" appears not to be difficult. The perturbations "add an argument" and "add a step" are restricted to adding only steps or arguments known within the algorithm. With this restriction added to the perturbations, we claim that such perturbations can be generated automatically is made. Young and O'Shea's work [37] provides much support for these claims.

Young and O'Shea [37] claim that their production system provides an alternative, simpler interpretation of the subtraction errors analysed by Burton and Brown [6]. They add 10 additional rules to handle the zero-pattern errors flagged by Burton and Brown (for details see [6]). Close examination of these additional rules reveals that they can, in fact, be thought of as steps from other parts of the PROLOG description being applied in the incorrect place (such as borrowing when not required), or that the bug can be accounted for by another perturbation such as "substituting on argument for another" [22].

To illustrate this point, consider the following bugs explained by the above perturbations to the PROLOG algorithm, which Young and O'Shea explain via additional rules:

(i) 72-59=27; this error involves incorrectly testing the condition **ge(minuend subtrahend)**, followed by a permutation of the arguments to ***subtract-single-digits*** within the non-borrowing case of column-subtraction.

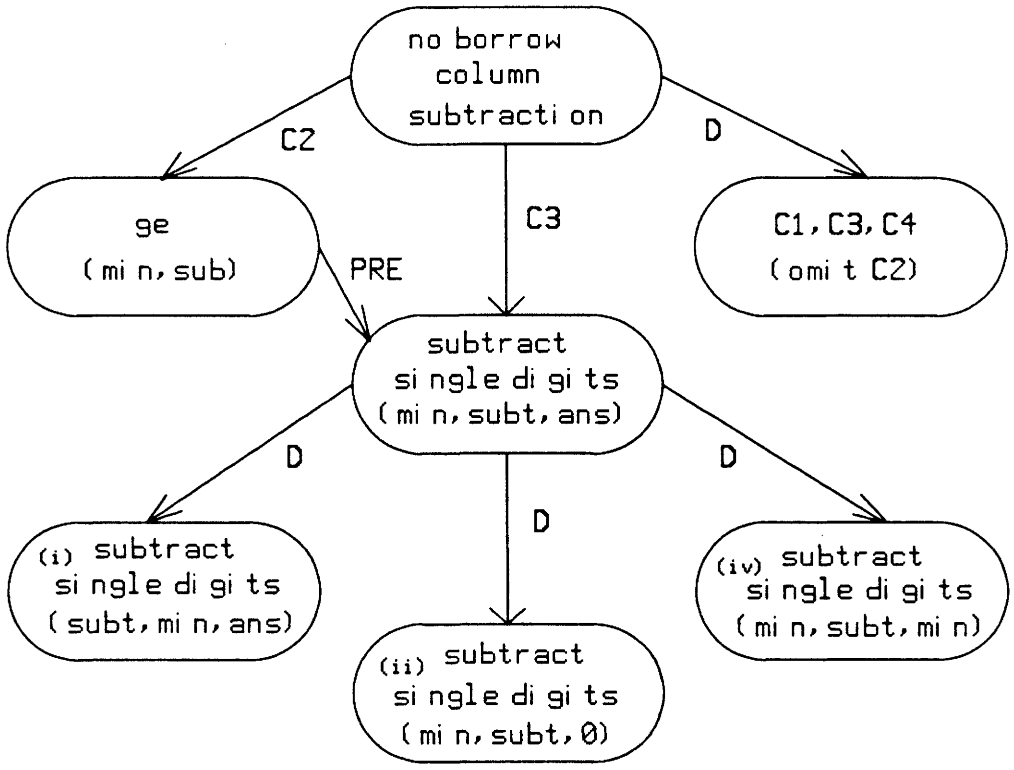
(ii) 96-42=44; this error involves incorrectly testing the condition **lt(minuend subtrahend)** in the borrowing case of column-subtraction.

(iii) 72-59=20; this error results from substituting the minuend for the subtrahend within ***subtract-single-digits***.

(iv) 70-26=56; this error results from substituting the subtrahend for the answer within ***subtract-single-digits***.

To illustrate this pictorially, consider the graph in Figure 5.9. The above permutations in example (i), (iii) and (iv) are shown.

There are many aspects of generating and maintaining the genetic graph which we have not addressed here such as recognizing generalizations, detecting analogies etc. These are difficult tasks which arise in other areas of AI and obviously warrant further investigation. Such research will enhance the effectiveness of the student model.


no borrow column subtraction
C2
D
C3
ge (min, sub)
PRE
C1, C3, C4 (omit C2)
subtract single digits (min, subt, ans)
D
D
D
(i) subtract single digits (subt, min, ans)
(ii) subtract single digits (min, subt, 0)
(iv) subtract single digits (min, subt, min)

(i) n-m = m-n when m>n
(ii) n-m = 0 whem m>n
(iv) 0-n = 0

Permutations
Figure 5.9

5.6. Ballet

The chosen ballet technique studies common faults which arise when teaching classical ballet to both males and females. The attitude of the expert, Lawson, is that ballet teachers are not always aware of faults which arise due to a child's growth pattern. The faults, cited by Lawson, and included in the genetic graph are as follows[24]:

INCORRECT STANCE which can result in an over-arched back, over-turned feet and loss of flexibility throughout the torso, head and arms because of tension and lack of stability in the supporting leg.

FAILURE TO UNDERSTAND THE CORRECT TILT OF THE PELVIS, i.e. the hingelike movement of the torso and hip-joint, and the need to raise the weight firmly upwards away from the waist line, leading to a loss of the full extension of the legs and torso.

INCORRECT RISE to *demi-point*, leading to sickling of the foot and loss of the straight line of the legs.

INCORRECT TRANSFER OF WEIGHT and proper use of the supporting leg, together with the failure of the foot to use the floor as a spring board and firm base for the dance.

BAD PLACING of the hand on the barre, poor use of the head and the focusing of the eyes as the child grows.

IGNORANCE OF THE ESSENTIAL DIFFERENCES to be made between the teaching of girls and boys.

FAILURE consistently to use the conventions of classical dance during the first years of training.

The approach taken in diagnosing these faults is to indicate the proper technique for various classical ballet movements and flagging dangers for which to watch. Hence the genetic graphs developed could represent a child with perfect technique or a child with various technical flaws. Using some of these common errors and the extensive teaching explanations, the following genetic graphs were developed (Figures 5.10-5.17).

Overall Genetic Graph for Ballet

Figure 5.10 illustrates one version of the entire genetic graph across all levels without including node expansion. When beginning ballet, a child is first taught how to hold his/her body correctly. When the child's body begins to mature, certain adjustments to this beginning stance must be made to account for the physical differences. This can be seen in the graph where it starts with the beginning stance node which is then refined one level to the nodes for female and male mature stances. It is interesting to note that a component in level 1 (arms and hands node of Figure 5.10) can also be a component of a node in another level (level 2 in this case). Allowing for this interlevel linking reduces the redundancy that could occur if it was necessary

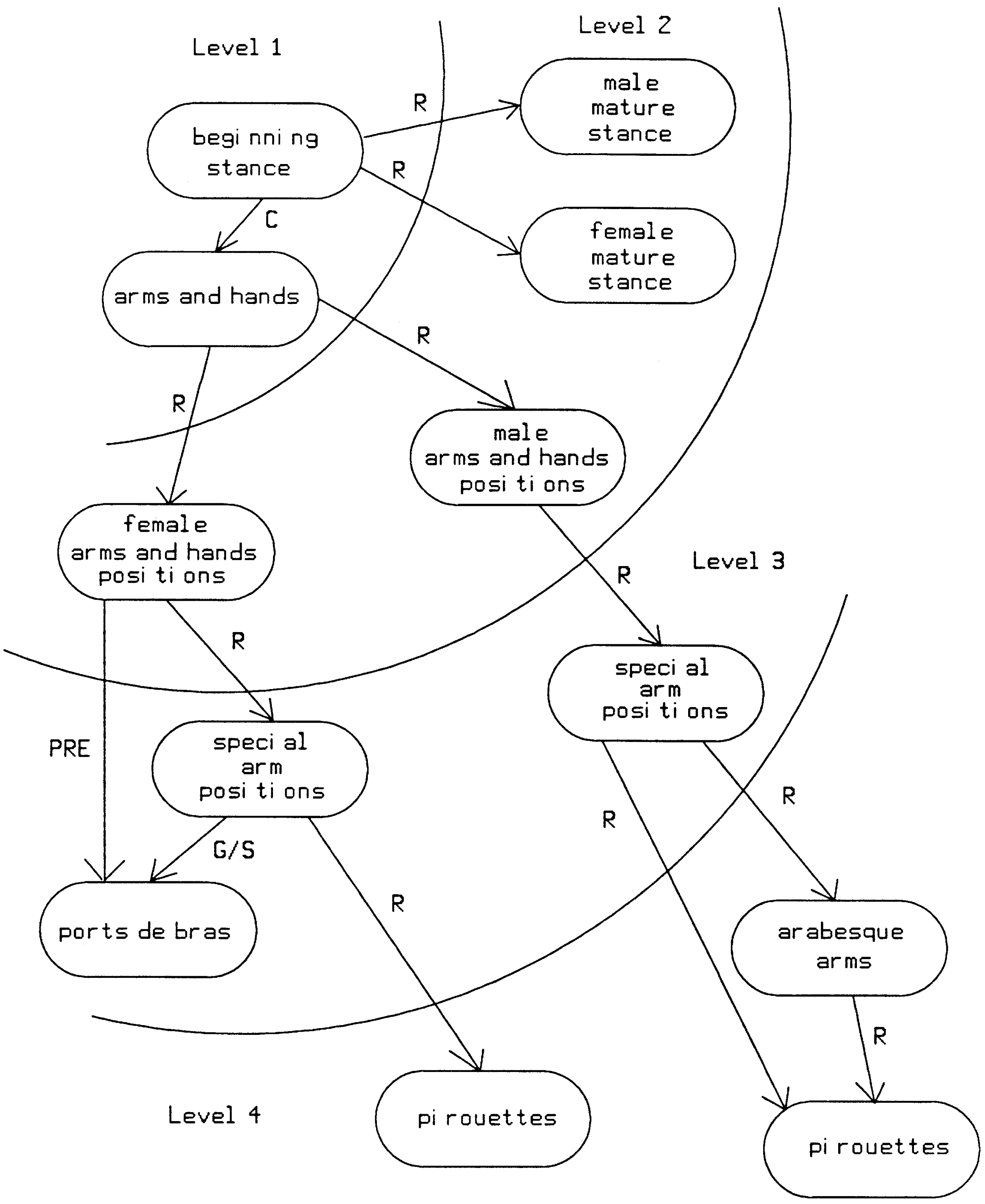

Level 1
Level 2
Level 3
Level 4
beginning stance
male mature stance
female mature stance
arms and hands
male arms and hands positions
female arms and hands positions
special arm positions
special arm positions
ports de bras
arabesque arms
pirouettes
pirouettes
R
C
PRE
G/S

Figure 5.10

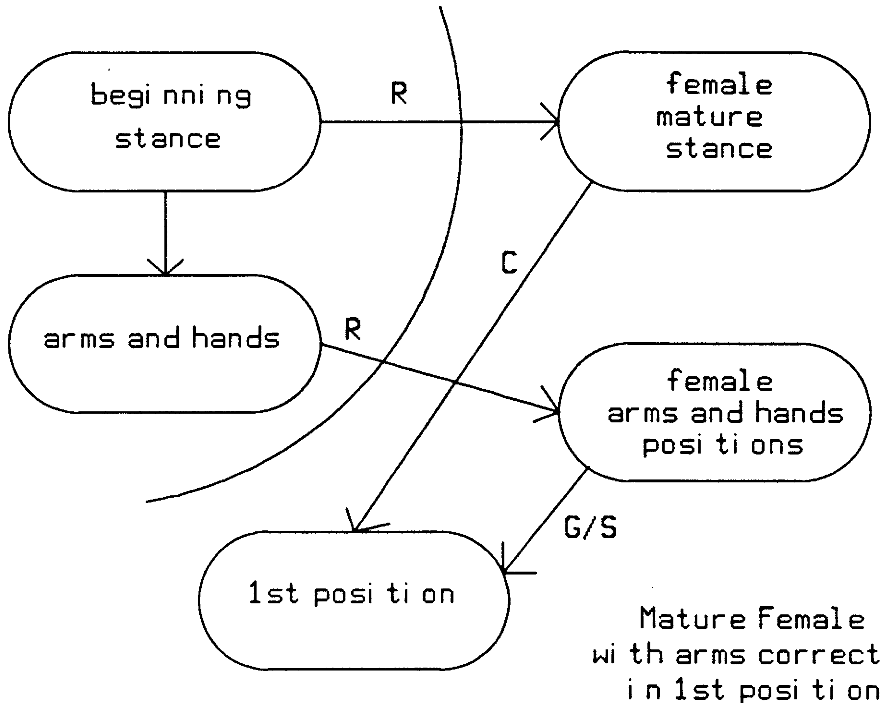
to repeat an identical node.

The next item of importance in Figure 5.10 is that the arms and hands component of the beginning stance node becomes refined over several levels independently of the stance refinements. Hence, the genetic graph can be regarded multidimensionally. The level 2 of either mature stance can have as its arms component, a node from any of the levels of arms. This leads to a large number of possible combinations.

To illustrate this flexibility, Figure 5.11 demonstrates a genetic graph for a mature female with correct stance and arms held correctly in 1st position, and a mature male with correct stance and his arms incorrectly in 5th position.

Beginning Stance Details

Figure 5.12 shows level 1 of the beginning stance in detail. To describe the proper position for both young boys and girls, information regarding placement of the head, spine, arms, knees, feet, eyes and weight is given. Each of these components is included as a node in the genetic graph; the details of their position, or directions on how to position are contained in the expert. The expert contains the following information about each of the seven components:


beginning stance
R
female mature stance
C
arms and hands
R
female arms and hands positions
G/S
1st position
Mature Female with arms correctly in 1st position

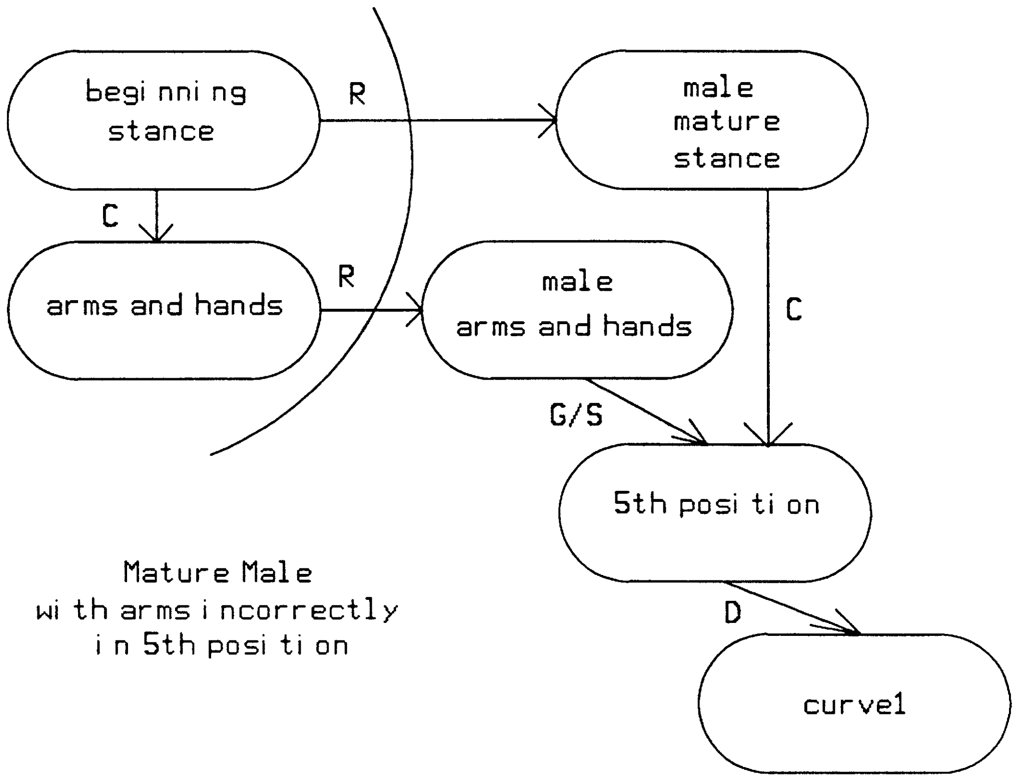
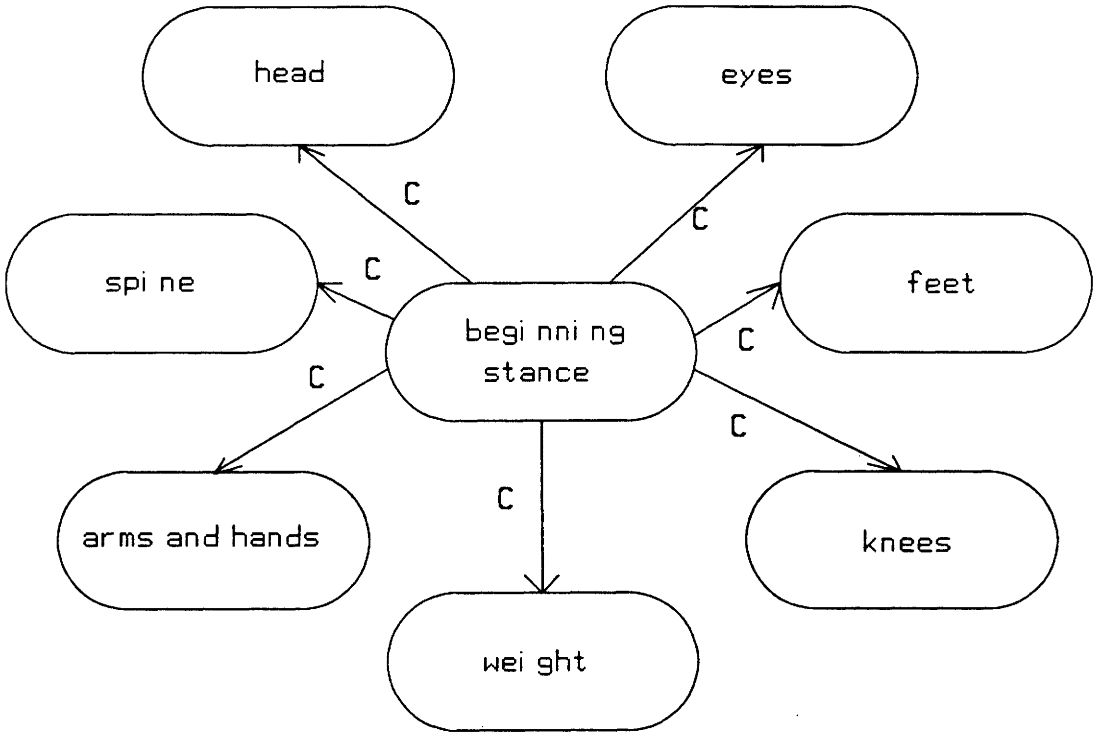

beginning stance
R
male mature stance
C
arms and hands
R
male arms and hands
C
G/S
5th position
D
curvel
Mature Male with arms incorrectly in 5th position

Figure 5.11


head
eyes
spi ne
begi nni ng stance
feet
arms and hands
knees
wei ght
C
C
C
C
C
C
C
C

Fi gure 5. 12

head: crown is directly over instep

spine: pulled up with no lift of chest

arms: hang softly at sides

weight: balanced over three points of foot, which are the big and little toe joints and the heel

knees: directly above the toes

feet: close but not pressed together, pointing straight ahead

eyes: directly forward and focusing on something opposite their level

Deviations may arise if the test condition for a component is not met exactly. This would be shown in the genetic graph with a deviation link and a correction link if it is known how to correct the mistake.

Male Mature Stance

As a boy begins to mature, the difference between his stance and a girl's must be explicitly defined. A mature male body is held erect more easily than a female's because he has less changes in his physique. However, since boys lose their "baby fat" later than girls, they must work harder to maintain the correct stance from the beginning. Figure 5.13 shows the details of the male mature stance. The following information resides in the expert:

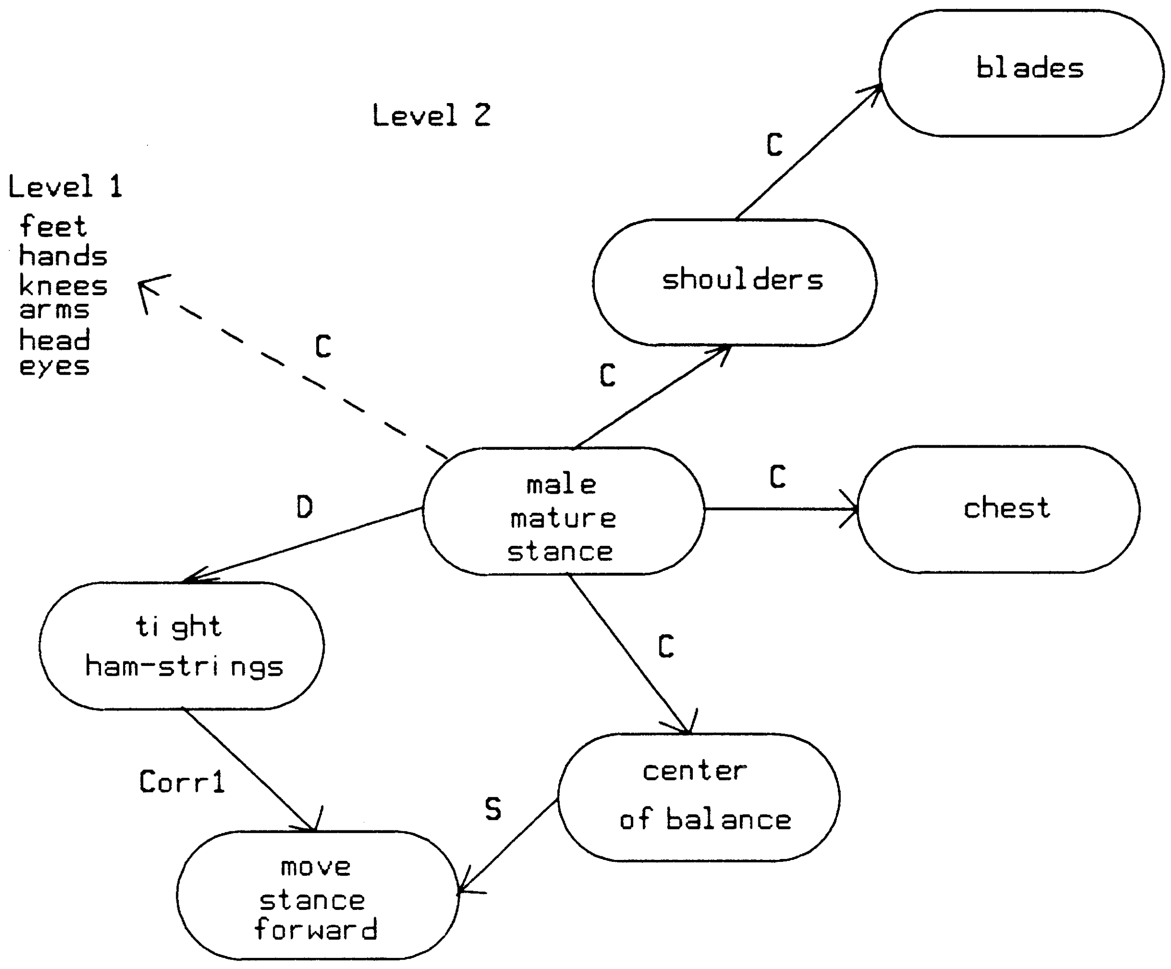

blades
Level 2
C
Level 1
feet
hands
knees
arms
head
eyes
shoulders
C
C
male
mature
stance
C
chest
D
tight
ham-strings
C
center
of balance
Corr1
S
move
stance
forward

Figure 5.13

chest: expanded

shoulders: held over, but just in front of the hips

blades: pressed downwards along the spine

weight: forward over three points of balance

centre of balance: from back of crown through spine to where he balances over centre of his feet

tight ham-string: correct with a more forward stance

The eyes, head, arms, knees and feet components are the same components from level 1. The interesting case of a deviation is explicitly shown in this portion of the genetic graph. The deviation, tight ham-strings, is corrected by having a more forward stance. This more forward stance is very important as it prevents the knees from being pulled too far back when a child's muscles are tightened. The correction, in this case, is a node itself because it represents a different physical position than that of the male mature position. If the correction did not physically alter the body position, the link from the deviant node to the male mature stance node would contain the correction and it would not be a node itself. The correction, in this example, is a special case of where the centre of balance is located, hence it is also a specialization of the centre of balance node.

Female Mature Stance

The maturing female has more to cope with than the maturing male. She has to deal with the changes caused by a developing bust and buttocks. Four slight changes need to be made to the beginning stance. First of all, she must maintain correct breathing so the chest expands outwards and sideways to carry the extra weight. The head needs to be lifted upwards so her centre line of balance is adjusted and her shoulders are now directly over her hips. Lastly, she must learn to stretch her spine fully upwards from the waist and her legs fully outwards away from the hip-socket. This allows for freer movement between the upper body and lower body. The genetic graph in Figure 5.14 shows these changes. Once again, the following information is included in the expert:

centre of balance: from centre of crown through the spine where she balances over the centre of her feet

head: slight lift upwards

shoulders: directly over the hips

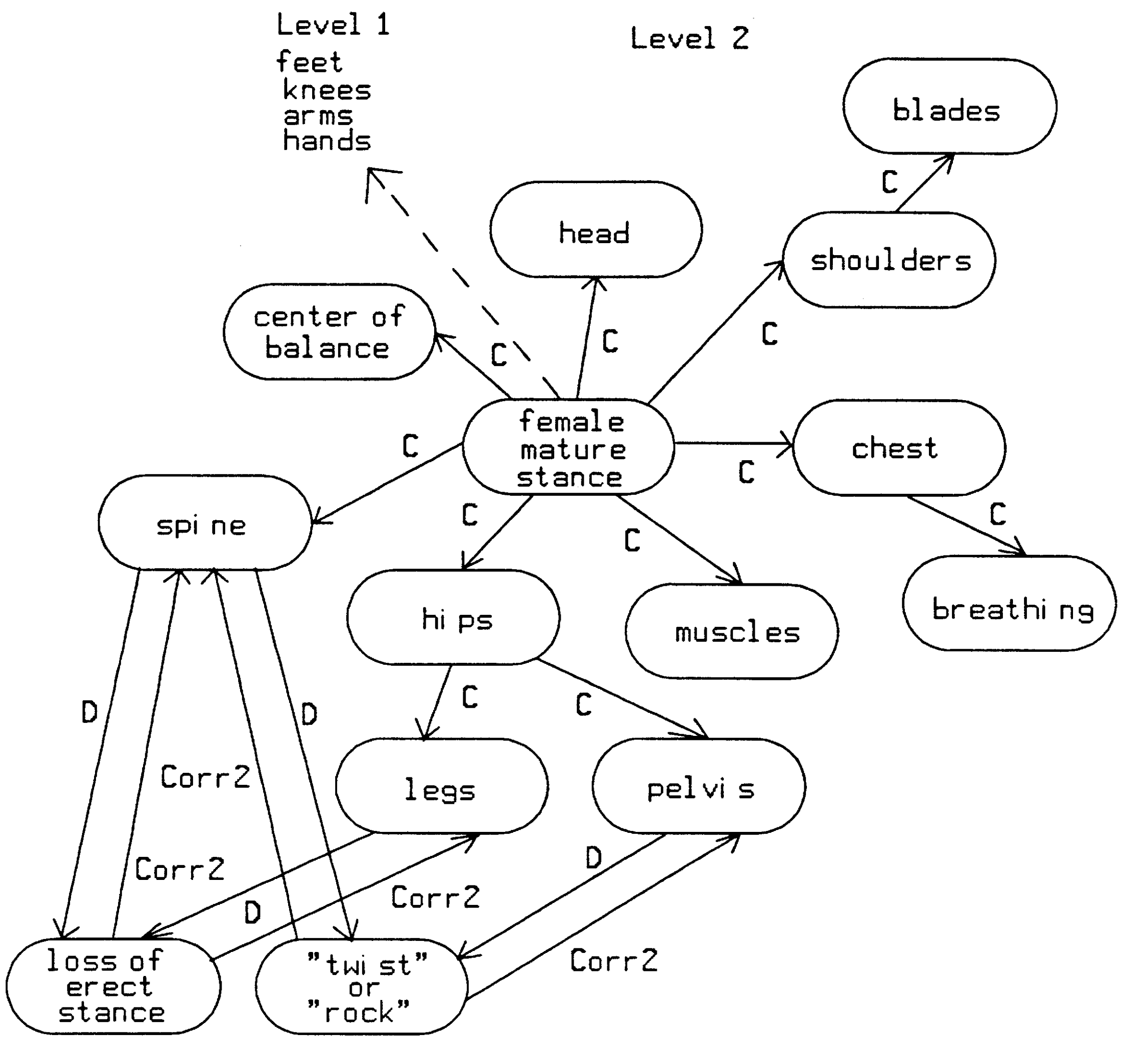
blades: slide downwards along the spine

chest: held firmly and freely, expanding upwards and sideways

breathing: deeply so chest expands outwards and sideways

muscles: in trunk and thigh sustains the correct stance

spine: stretched fully upwards from the waist



Level 1
feet
knees
arms
hands
Level 2
blades
C
head
shoulders
center of balance
C
C
C
female mature stance
C
chest
C
spine
C
C
C
hips
muscles
breathing
D
C
C
D
Corr2
legs
pelvis
Corr2
D
D
Corr2
loss of erect stance
"twist" or "rock"
Corr2

Figure 5.14

hip: in correct position

legs: stretched fully outwards and away from the hip-socket

pelvis: tilted forward from the hip-joint

"twist","rock": watch for backwards tilt of pelvis when curve of spine is needed, and correct by having pelvis tilted properly and the spine in correct position

loss of erect stance: wrong placement of legs or spine, correct by maintaining proper positioning

One of the interesting aspects of this genetic graph is the interaction of the pelvis, legs and spine. This interaction is necessary to allow for the free movement between the upper and lower body. The pelvis and legs are analogous components because their correct position is indicated by their relationships to the hip. A proper line of the spine will occur only when the pelvis is tilted forwards, from the hip-joint, the spine is pulled out of the waist and the leg away from the hip-socket. A "twist" or "rock" occurs to the pelvis and the spine looses it's erect stance if these components are not used correctly.

Male Arms and Hands

In ballet, the arms and hands of the dancer contribute greatly to the overall presentation. There are technical positions for their placement but at the same time they must flow and be relaxed and never look tense or uncontrolled. They must be an enhancement to the dancer but never a distraction. The genetic graph in Figure 5.15 provides the details of the male arm and hand positions node, and how this node is refined over several levels. The nodes defined in the expert are:

1st position: arms in first position

curve 1: follow the line of shoulder, slightly downwards

fingers 1: fingertips level with the diaphragm (breastbone)

fingers 2: breadth of the forehead apart

hand not 2nd: downwards according to line required

5th position: arms in fifth position

curve 2: above the ears and by lifting eyes he can see the insides of his hands

fingers 3: fingertips over and just in front of crown of head

fingers 4: width of his forehead apart

2nd position: arms in second position

hand 2nd: facing directly downwards

shoulders: pulled outwards and pressed downwards

chest: fully expanded with easy breathing

handshake: natural position

too stiff: correct by softening

In this genetic graph, the male arm and hand positions node has four specializations associated with it. They are the handshake and arms in 1st, 2nd and 5th positions. The arm positions illustrate Goldstein's definition of analogous nodes (recall Chapter 3). The finger components of the curve nodes are included because they are test conditions for proper curve of the arm. There are many further analogies which could be included in the graph; they are not illustrated here for the sake of brevity and simplicity. For example, there should be an analogy link between all the finger nodes of curve 1 and a curve 2 as well as between all the finger nodes in the female arm and hand position genetic graph. Similarly all male arm positions are analogous to all female arm positions. To demonstrate the flexibility of the graph, suppose the ballet teacher separated his/her teaching of the arm positions and hand positions. Figure 5.16 illustrates a possible variation in the genetic graph representation.

The refinements from the male arm and hand positions node through to the pirouette node is based upon the assumption that this is the order in which the teacher taught the student. Essentially the refinement from one node to the next corresponds to an increase in difficulty of the move or positioning. For example, in order to use the arms properly while performing a pirouette, the student must already know how to place the arms in both 2nd and a shortened 1st position. The refinement from the special

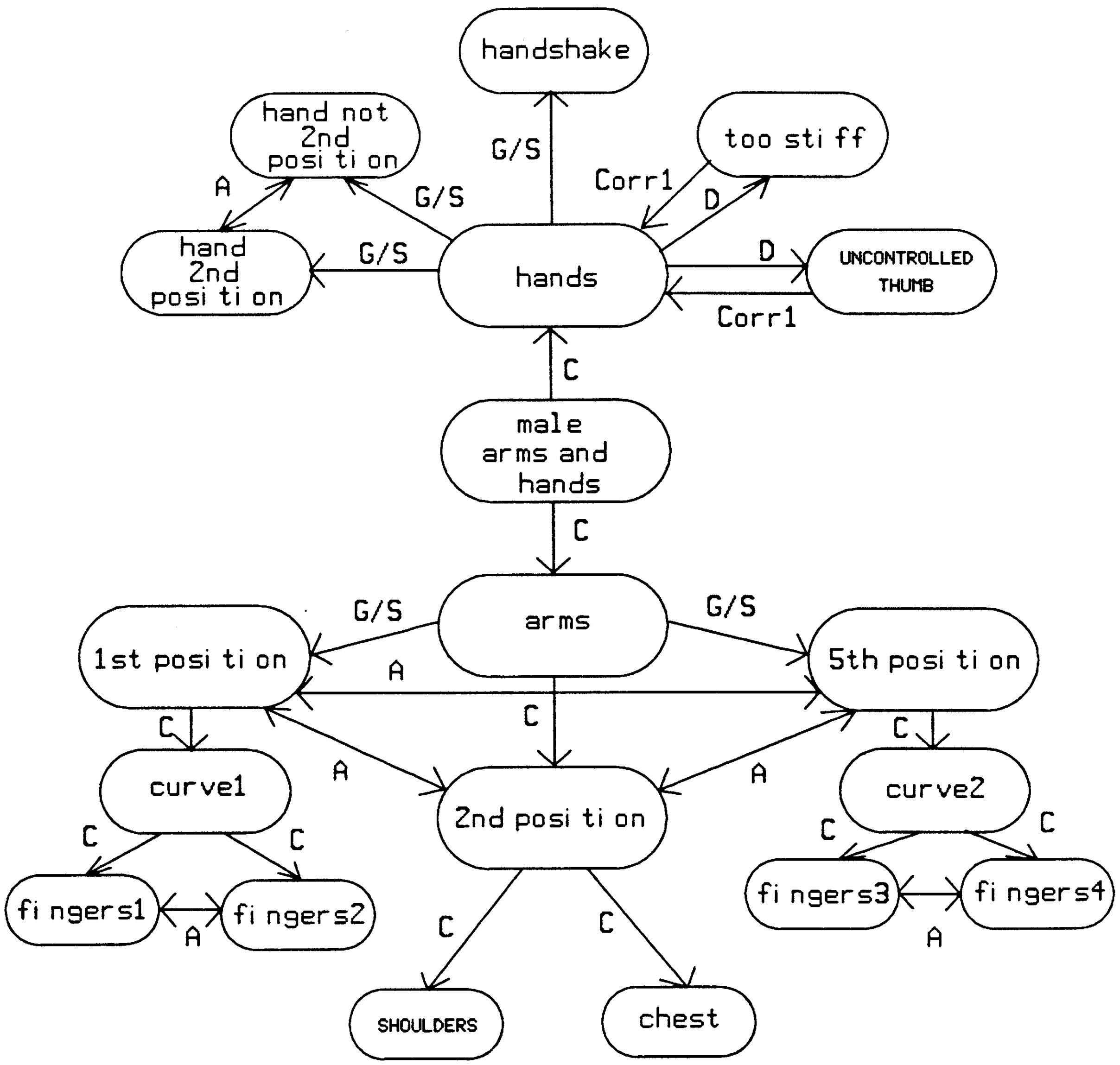

handshake
hand not 2nd position
too stiff
G/S
Corr1
A
G/S
D
hand 2nd position
G/S
hands
D
UNCONTROLLED THUMB
Corr1
C
male arms and hands
C
G/S
arms
G/S
1st position
A
5th position
C
C
C
curve1
A
2nd position
A
curve2
C
C
C
C
fingers1
fingers2
fingers3
fingers4
A
C
C
A
SHOULDERS
chest

Figure 5.16

arm position node through level 4 to the pirouette node indicates that it is not necessary for the student to know the arabesque position in order to perform a pirouette. There are many genetic graph representations for the developing arm which would depend on the method of teaching used by the instructor.

Female Arms and Hands

The arms and hands of a girl must always be rounder than a boy's and must move more smoothly. The arms are always held in a position such that the elbows are just in front of the shoulders. The girl should try to make her arms appear to be an extension of her shoulders and not a separate entity. Figure 5.17 shows the details of level 2 of the girl's arms and hands node and the following information is included in the expert:

female arm and hand positions: softer and rounder than males with elbows always just in front of the shoulders

bras bas: slightly rounded elbows with arms forward and round the body, the eyes should be able to see inside the palms without the head being lowered

fingers 6: middle fingers are 4 inches apart

fingers 7: fingers are 6 inches from the body

1st position: raise arms from ***bras bas*** without moving shoulders

fingers 5: fingers opposite diaphragm

5th position: raise arms above head

curve 3: arms curved above but in front of the head, if eyes are raised they look into the palms

hands 1: hands are 9 inches above the head

hands 2: hands are 6 inches from the perpendicular

2nd position: from 1st position open arms to side

shape: rounded so if head is turned sideways eyes can glance down the back of the arm from shoulder to elbow

square shoulders: if a girl's shoulders are very square correct with a slight lowering of the arms

tense: correct by softening

straight: never straight for basic arm positions therefore correct by rounding

incorrect line: correct by following the line of the shoulders

Once a child has learned the four arm positions (*bras bas*, 1st, 2nd and 5th) the *ports de bras* can be performed. To indicate that these must be learned first, the prerequisite link is used. Incidentally, the *ports de bras* is a beautiful series of arm movements (from one base position to another) set to music and combined with different tilts of the head.

Conclusions

The genetic graphs contained in this section on ballet are only a few of many possibilities. One reason for chosing the ballet domain was to illustrate the domain independence of the genetic graph approach. The domain fulfilled this goal, as well as demonstrated how simple and uncomplicated the subtraction domain was in comparison to that of ballet. This section just touched on the possibilities for representing a small subset of elementary ballet. Associated with almost every node could be possible deviations and alternative styles of teaching. These were not included as they would require additional genetic graphs to illustrate them.

We have demonstrated that the genetic graph can represent mistakes that arise when learning classical ballet. Having this information in the form of the genetic graph is beneficial to the instructor when performing a diagnosis or analyzing a student's progress. Our goal here was not to represent all aspects of ballet, but rather to demonstrate the feasibility, including both flexibility and appropriateness, of the genetic graph approach to student modelling. The graphs contained herein are sufficient evidence of both.

Chapter 6
Conclusions

The student model is one of the major components of an ICAI system. Without an accurate and detailed model of the learner the ICAI system will be ineffective. The genetic graph approach to student modelling fulfills the criteria for an ideal student model. What we have done is formalized the notion of a genetic graph and demonstrated that the same representation scheme, the genetic graph, can be used in several different ways within one domain, and for several possible ICAI domains. Furthermore, the design is accommodating in the amount of detail it contains. These are desirable features since most previous models are domain specific, and static in what they represent.

The genetic graph overcomes the problem of modelling the student as a subset of the expert's skills by extending the type of knowledge that can be represented. Goldstein's [19] evolutionary relationships encompass most situations. However, if a situation arises that is not covered, the extended genetic graph relationships described in Chapter 4 demonstrate that new relationships can easily be incorporated.

Goldstein states "the GG [genetic graph] does not solve the modelling problem", however, he also points out "it provides a more powerful foundation for modelling than either a script of correct answers or a set of expert skills" [19, p69]. We support these claims entirely. The major problems that still exist with modelling are not in the representation of the student, but deal with the interaction of all components of an ICAI system.

One major issue is how to automatically generate the genetic graph from the expert and the course graph. Secondly, a problem arises with maintaining the genetic graph. This is a two-fold problem: how to automatically generate new sections of the graph and how to discard old sections of the graph. These problems require the enhancement of the psychologist to include algorithms that recognize analogies, specializations, and deviations that may arise. This should not be viewed as an insurmountable situation since some current AI research addresses these problems.

The representation scheme employed during implementation of the expert and student model is another major concern. This is also being addressed by AI research in the area of knowledge representation and a good summary can be found in [34].

References

[1] Asklock, R.B., **Error Patterns in Computation**, Bell and Howell, 1976.

[2] Barr, A. and E.A. Feigenbaum (eds.), **The Handbook of Artificial Intelligence, Vol 2**, William Kaufmann, 1982.

[3] Bregar, W.S. and A.M. Farley, "Artificial Intelligence Approaches to Computer-Based Instruction," *Journal of Computer-Based Instruction*, *Vol.* 6 , *No.* 4, pp. 106-114, May 1980.

[4] Brown, J.S., "Uses of Artificial Intelligence and Advanced Computer Technology in Education," **Computers and Communications: Implications for Education**, ed. R.J. Seidel and M. Rubin, pp. 253-270, Academic Press, 1977.

[5] Brown, J.S., R.R. Burton, and K. Larkin, "Representing and Using Procedural Bugs for Educational Purposes," *Proceedings of 1977 Annual Conference, Association for Computing Machinery*, pp. 247-255, Seattle, October 1977.

[6] Brown, J.S. and R.R. Burton, "Diagnostic Models for Procedural Bugs in Basic Mathematical Skills," *Cognitive Science*, *Vol.* 2, pp. 155-192, 1978.

[7] Brown, J.S. and R.R. Burton and J. de Kleer, "Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II and III," **Intelligent Tutoring Systems**, ed. Burton and Brown, pp. 227-280, Academic Press, 1982.

[8] Brown, J.S. and R.R. Burton (eds.), **Intelligent Tutoring Systems**, Academic Press, 1982.

[9] Burton, R.R. and J.S. Brown, "Semantic Grammar: A Technique for Constructing Natural Language Interface to Instructional Systems," BBN Report No. 3587, ICAI Report No. 5, May 1977.

[10] Burton, R.R., "Diagnosing Bugs in a Simple Procedural Skill," **Intelligent Tutoring Systems**, ed. D. Sleeman and J.S. Brown, pp. 157-183, Academic Press, Toronto, 1982.

[11] Carbonell, J.G., R.S. Michalski, and T. Mitchell, "An Overview of Machine Learning," **Machine Learning**, pp. 3-24, Tioga Publishing Company, 1983.

[12] Carr, B.P. and I.P. Goldstein, "Overlays: A Theory of Modelling for Computer Aided Instruction," MIT Tech Report #210, February 1977.

[13] Carr, B. P., "WUSOR II: A Computer Aided Instruction Program with Student Model," MIT Tech Report #255, June 1977.

[14] Chambers, J.A. and J.W. Sprecher, "Computer-Assisted Instruction: Current Trends and Critical Issues," *Communications of the ACM, Vol.* 23, 6, pp. 332-342, June 1980.

[15] Colbourn, M.J. and J. McLeod, "Computer-Guided Educational Diagnosis: A Prototype Expert System," *Journal of Special Education Technology VI, Vol.* 1, pp. 30-39, 1984.

[16] Colbourn, M.J., "Applications of Artificial Intelligence within Education," *International Journal of Computers and Math, special issue: Practical AI Systems*, to appear 1985.

[17] Collins, A., E. Warnock, and J. Passafiume, "Analysis and Synthesis of Tutorial Dialogues," **The Psychology of Learning and Motivation**, ed. G. Bower, *Vol.* 9, Academic Press, 1975.

[18] Dietterich, T.G. and R.S. Michalski, "Inductive Learning of Structural Description: Evaluation Criteria and Comparative Review of Selected Methods," *Artificial Intelligence, Vol.* 16, pp. 257-294, 1981.

[19] Goldstein, I.P., "The Genetic Graph: A Representation for the Evolution of Procedural Knowledge," **Intelligent Tutoring Systems**, ed. D. Sleeman and J.S. Brown, pp. 51-77, Academic Press, Toronto, 1982.

[20] Hayes-Roth, F., D.A. Waterman, and D.B. Lenat (eds.), **Building Expert Systems**, Addison-Weley Publishing Company, Inc., 1983.

[21] Howe, J.A.M., "Artificial Intelligence and Computer-Assisted Learning: Ten Years On," **Selected Readings in C-Based Learning**, ed. Nick Rushby, London/Nichols Publishing Company, New York.

[22] Jones, M.L., D. Poole, and B.J. Wasson, "Student Models: The Genetic Graph Approach," preprint.

[23] Lantz, B.S., W.S. Bregar, and A.M. Farley, "An Intelligent CAI Sytem for Teaching Equation Solving," *Journal of Computer-Based Instruction, Vol.* 10 , *No.* 1 & 2, pp. 35-42, Summer 1983.

[24] Lawson, J., **The Teaching of Classical Ballet: Common Faults in Young Dancers and Their Training**, A & C Black Limited, London, 1973.

[25] McCalla, G. and D. Peachy and B. Ward, "An Architecture for the Design of Large Scale Intelligent Teaching Systems," 9Proc. of the Fourth National Conference of the Canadian Society for the Computational Studies of Intelligence, Saskatoon, Saskatchewan, pp 85-91, May 1982.

[26] Michalski, R.S., J.G. Carbonell, and T. Mitchell (eds.), **Machine Learning**, Tioga Publishing Company, 1983.

[27] O'Shea, T., "A Self-Improving Quadratic Tutor," **Intelligent Tutoring Systems**, ed. D. Sleeman and J.S. Brown, pp. 309-336, Academic Press, 1982.

[28] Pagliaro, L.A., "The History and Development of CAI: 1926-1981, An Overview," *The Alberta Journal of Educational Research, Vol.* XXIX , *No.* 1, pp. 75-84, March 1983.

[29] Peachey, D.R., "An Architecture for Plan-Based Computer Assisted Instruction," M.Science Thesis, Department of Computational Science, University of Saskatchewan, 1983.

[30] Sacerdoti, E., "A Structure for Plans and Behaviour," *The Artificial Intelligence Series*, Elsevier North-Holland, 1977.

[31] Self, J.A., "Student Models in Computer-Assisted Instruction," *Journal of Man-Machine, Vol.* 6, pp. 261-276, 1974.

[32] Stansfield, J., B. Carr, and I. Goldstein, "Wumpus Advisor I: A First Implementation of a Program that Tutors Logical and Probabilistic Reasoning Skills," MIT AI Memo No. 381, September 1976.

[33] Tennant, H., "Evaluation of Natural Language Processors," PhD. Thesis, University of Illiniois, 1981.

[34] Wellsch, K.C., "A Computational Model for Reasoning by Analogy," M.Math Thesis, Department of Computer Science, University of Waterloo, in progress.

[35] Westcourt, K., M. Beard, and L. Gould, "Knowledge-based Adaptive Curriculum Sequencing for CAI: Application of a Network Representation," *Proceeding of 1977 Annual Conference, Association for Computing Machinery*, pp. 234-240, October 1977.

[36] Yob, G., "Hunt the Wumpus," *Creative Computing*, pp. 51-54, September/October 1975.

[37] Young, R.M. and T. O'Shea, "Errors in Children's Subtraction," *Cognitive Science, Vol.* 11, pp. 153-177, 1982.

APPENDIX [FROM 24]

Due to poor print quality, this appendix has not been duplicated. It is available upon request.