CANADIAN THESES

THÈSES CANADIENNES

The Effects of the Formal Features of Language Learning Activities on the Selection of Authoring Systems

Roger Kenner

A Thesis

for

The Centre

for

Teaching English as a Second Language

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Arts at
Concordia University
Montréal, Québec, Canada

March 1986

ISBN 0-315-30611-4

ABSTRACT

The Effects of the Formal Features of Language Learning Activities on the Selection of Authoring Systems

Roger Kenner

A model of those formal features of language learning activities which affect their realisation as computer-based activities, through the use of an authoring system, is inductively constructed. Three phases of an activity are considered: The presentation of the activity, the analysis of the learner's response, and the selection of subsequent items. The possible relationship between the formal features of language learning activities and their pedagogical effectiveness is also examined and no correlation is found. The sophistication of a given authoring system should not, therefore, have any appreciable negative effect on its ability to be used in the creation of quality courseware. The criteria established by the model can be used in the selection of effective authoring systems for computer-assisted language learning (CALL) and can serve as a guide to those types of language learning activity which can best be presented via the computer.

Table of Contents

Chapter | Page

Table of Contents

Appendices and Tables

Chapter One: Introduction

1.1 The general problem

The importance of computers to our society and to education in particular has become the subject of widespread speculation since the advent of powerful and inexpensive microcomputers a few years ago. The impact of what has often been called the "computer revolution" has been keenly felt in the field of language teaching (Wright, 1980, p.4; Davies, 1982, p.4; Gillespie, 1982). The acute interest with which language teachers follow developments in this area can be seen from the increasing number of presentations on computers given at conventions of language teachers.

The role and future of the computer in language teaching remains far from certain. Computer-assisted language learning (CALL) is one of the possible applications. CALL can be briefly described as structured and interactive language practice where the performance of the learner is mediated by the computer. (This definition is more fully developed in section 2.1). In section 2.3, the current state of CALL and those factors prejudicial to its widespread use by language teachers are examined, and the production of an adequate supply of quality teaching materials for the medium is suggested as one of the major problems to be solved. Authoring systems are software packages that allow teachers to create CALL materials without having to delve into computer programming or seek the services of a computer programmer (The nature of authoring systems is more fully explored in section 2.2). The use of authoring systems to allow teachers to create their own materials is proposed as a likely solution to the problem mentioned above.

1.2 The specific problem

Before the use of authoring systems can begin to have a significant impact on the shortage of computer-based language teaching materials, language teachers must be provided with a means of selecting those authoring systems appropriate to their particular needs (Ashmore, 1983, p.5). For such a selection to be possible, authoring systems must be evaluated with respect to those features which are important to the development of effective language learning activities (Jensen, 1982, p.50). No such measure exists at present (Kearsley, 1982, p.436).

Current classifications of authoring systems employ criteria which, while important in themselves to the global selection process, are only peripheral to the requirements of language learning materials (Raschias and Lange, 1984). Indeed, Dowsey (1974, p.402) indicates that specific reference to particular subject matters should be avoided when doing a general survey of authoring systems because of the diverse requirements of the different domains. A common ground for comparison is established by concentrating on those features which affect the nature of the interaction between the author and the computer (Dowsey, 1974, p.402). Classifications of authoring systems have thus been neutral with respect to language learning (Boyd, 1970; Bagley, 1974; Dowsey, 1974; Barker & Singh, 1983).

More recent evaluations of authoring systems have continued this trend. Grabinger (1985) evaluates three authoring systems according to diverse criteria such as cost, copy protection, ease of learning, ease of use, graphics and sound capabilities, peripheral interfacing capability, support documentation, screen design, instructional design features, and management and testing features. Locatis and Carr (1985) list as their criteria: hardware requirements, cost and contract terms, graphics, sound, external interfaces, management, documentation, and the time required to author a package. As a practicioner in this area, the writer feels that Pattison's (1985) criteria: answer-matching, feedback, flexible branching, record keeping, screen display, and timing, might be closer to what is required.

O'Neal and Fairweather (1984) set out to establish the important factors by which to measure authoring systems as to their applicability to computer-assisted language learning. They selected three dimensions of evaluation: Power, ease of use, and productivity. In their scheme, "Ease of use" was roughly equivalent to "How long it takes to learn the system" and "productivity" to "How long it takes to create an hour of instruction". The "power" dimension was largely ignored. Their analysis, therefore, cannot be considered as specific to the subject at all, but rather a general overview of the type described above.

Recently, authoring systems have been developed with the goal of directly reflecting the needs of language teachers. One such example is Dasher (Pusack, 1982). The underlying assumption in Dasher is that enhanced answer processing is the key to effective computer-based language activities. Another example is Prompt (Paramskas and Mydlarski, 1985), where the underlying assumptions appear to be that the ability to gloss vocabulary and provide a range of error-specific feedback messages is significant.

Chapter One: Introduction

Discussions of the application of computers to language learning tend to focus primarily on the powers of the computer and then work backwards in applying these powers to actual language-learning activities. Nowhere in the literature is to be found an analysis which isolates features inherent to language activities in general and then discusses the impact of these features on the possible computerisation of the activities.

1.3 Specific questions addressed in this thesis

In this thesis, I will attempt to answer the following questions relevant to the problem of isolating criteria for the selection of authoring systems for computer-assisted language learning activities:

1. Can a model of existing language learning activities be constructed which will reveal those inherent features of the activities which must be considered when planning their computerisation? (Such a formal ~~feature~~ model is proposed in Chapter Three of this paper and a series of features and their requirements for realisation using an authoring system are listed.)

2. What is the range of computer-based language learning

activities that can theoretically be created through the use of an authoring system? The application of the model elaborated in Chapter Three delimits this range.

3. What special features inherent to meaningful and communicative language learning activities affect their realisation through the use of an authoring system? This writer's hypothesis is that the meaningful/communicative dimension of such activities is primarily a function of their content and therefore will not be reflected in any specific way in the formal feature model.

1.4 The purpose of the study

The purpose of this study is to establish those variable features of authoring systems which are important in determining the range and type of language learning activities that can be realised. Teachers may then use these features as criteria in selecting a particular authoring system.

Given a particular authoring system, the formal feature model permits teachers to determine what types of activities they can produce. The model also make teachers aware of the practical limits of what can be accomplished on the computer when using an authoring system and helps them to alter particular features of an activity so as to make its production easier.

Finally, the model is offered to as a guide to the developers of authoring systems for computer-assisted language learning. It points to those features which should be made more flexible if the authoring system is to be applicable to the creation of a wide range of language-learning activities. It also illustrates those features which have no direct effect on the applicability of the authoring system to language practice.

1.5 The limitations of the study

The scope of the present study has certain important limitations. The wide range of possible applications of the computer to language learning has been arbitrarily restricted by the rigid definition of computer-assisted language learning. Only the more traditional applications of the computer to language learning are considered. The almost limitless range of what the computer may ultmately make possible has been narrowed down to consideration of what authoring systems can accomplish, given current, generally affordable technologies.

The question of what particular types of activities should or should not be presented via the computer, nor, indeed, the larger question of whether particular activities should or should not be done at all, is not considered. Rather, the scope of the survey has been limited to the more neutral consideration of more or less traditional language learning activities, without reference to their pedagogical effectiveness.

Finally, no actual authoring systems will be analyzed. This paper discusses authoring systems in the abstract and, where necessary, assumes a theoretical, general authoring system, the characteristics of which are elaborated in section 2.2.

Chapter Two: Background and Definitions

The purpose of this chapter is to define the terms computer-assisted language learning (CALL) and "authoring system" and to discuss the importance of the latter to the future development of CALL.

Computer-Assisted Learning (CAL) is discussed and CALL is defined as a special subset of CAL which presents unique difficulties which the those designing software for the support of authors have not sufficiently addressed.

The various approaches that have been used for the production of language learning materials, or courseware, are briefly described. The term "authoring system" is defined and contrasted with other mechanisms for courseware production, such as authoring languages, The nature of authoring systems, as well as some of the features that serve to differentiate different types of authoring systems are also briefly examined.

Finally, the current state of CALL as a teaching tool is discussed and the major remaining impediments to its widespread reviewed. These include the dearth of acceptable courseware and the lack of teacher involvement in the creation of CALL materials. Key factors governing the quantity of software produced, the quality of that software, and the relationship between the two are examined. Reasons are advanced for arguing that use of authoring system by teachers provides the best means of satisfying the demand for both quality and quantity.

2.1 Computer-assisted language learning (CALL)

To arrive at an understanding of computer-assisted language learning (CALL), the more global term, computer assisted learning (CAL) is examined first. The specific type of CAL dealt with in this study is defined, the confusion over terminology existing in the field is discussed, and examples of the types of activities that are most often called CAL are given. CALL is then defined as a subset of CAL and the special factors which set it apart are considered.

2.1.1 Computer-assisted learning (CAL)

For the purposes of this study a specific type of computer-assisted learning (CAL) which might be refered to as "practice-oriented CAL" is defined. Practice-oriented CAL may be thought of as learners' practice involving interactions with computers in a structured environment.

Chapter Two: Background and Definitions

This structured environment is provided by computer programs called "courseware", the content of which has been established prior to the learners' interaction. The course of the interaction is mediated, or controlled, by the courseware. (Henceforth, references made to actions taken by the "computer" should be interpreted to mean actions foreseen in the computer programs that make up the courseware.)

"Practice", in the context of learning, is defined as the act of doing something repeatedly in order to acquire a skill. It can also be interpreted as instruction received through the repetition of exercises or lessons. "Rehearse" is offered as a synonym (Funk & Wagnall's Dictionary, 1976, p.1059).

The general application of the term CAL involves more than simply the "practice" of a given concept or skill. Learning via the computer can, after all, certainly take place in many situations in which there is no engagement in practice activities, per se. These applications are excluded from the current, working definition of CAL. The use of the computer as a tool of inquiry when making a data-base search, for example, is not practice, unless the subject is data-base retrieval. Learning will often, nevertheless, result from this activity. Similarly, the use of the computer as a word processor is not strictly practice unless one is practicing typing and redaction skills. Many experts maintain, however, that use of the computer for word-processing can facilitate learning higher level composition skills (Hopper, 1984; Bloch, 1986, p.4)

There is general agreement that CAL is interactive in the sense that the learners participate in an immediate, back-and-forth interaction with the computer during the learning experience. This activity can be contrasted with situations where the user deals with the computer in a more remote, detached manner, as when submitting a batch program for processing and returning hours later to receive the results.

Interactive practice with the computer may take place within the structured environment provided by the framework of a pre-established computer program. Alternatively, learners may interact with the computer directly, through some high-level computer language. Boyd (1982, p.307) sees the need to consider programmers as learners, and Bagley (1974, p.5) even considers such non-structured practice (where students must define the problem themselves and write a program to solve it) as a second category of CAL.

Chapter Two: Background and Definitions

The more general trend, however, is to adopt the narrower interpretation of CAL as "learners' interactions with computers within the structured framework of CAL courseware."

Higgins (1982, p. 28) illustrates the wide range of forms that interaction within a structured environment can take, from the restrictiveness of courseware built on the model of programmed instruction, to the variability exhibited by modern 'adventure' games. The common element is that all variation follows paths pre-determined by the courseware author. Higgins (1982, p.28) contrasts this with 'synthetic' programs where the computer is able to explore variations not previously foreseen by the author. He cites, as possible examples of synthetic programs, Weizenbaum's _Eliza_, Winograd's _Shrdlu_, and his own _Grammarland_, none of which could be considered as CAL according to the above definition.

The final component of the CAL definition is mediation, or evaluation of students' progress through the material, on the basis of some set of pre-established criteria. The courseware can act as a surrogate moderator on the basis of its built-in 'knowledge' of correct answers or paths. Mediation might involve matching the student's response against a stored 'correct' response and responding accordingly, as in a simple drill and practice exchange. It could also, as in a sophisticated adventure game, involve making learners accountable for their actions, some of which are more appropriate than others to the situation at hand. The Eliza program, where there are no "correct" or "incorrect" responses, would represent an example of non-mediated interaction and thus be excluded from consideration as an example of CAL.

It is difficult to find any definition of CAL which is acceptable across-the-board, hence the qualification of the current definition as "practice-oriented" CAL. Several historical factors make it difficult to achieve a definition which would receive universal acceptance.

Initially there existed a confusion between CAL and Programmed Instruction (PI) because CAL was simply seen as an outgrowth of print-based PI materials (Orstein, 1970, p.216). Scanlan (1971, p.85) distinguished the two terms by pointing out that PI did not allow for student-controlled learning or "unprogrammed instruction". Kenning and Kenning clearly differentiate PI and CAL by showing that any attempt to match the flexibility of the computer by the use of traditional print materials would be technically infeasible, requiring "huge 'scrambled books' with pages and pages of mostly unnecessary explanations, together with an extremely complicated system of cross-references." (Kenning and Kenning, 1983, p.2)

The difficulty in defining CAL is increased by the large number of acronyms and abbreviations that have come into use. The greatest recent confusion has been between CAL and Computer Assisted Instruction (CAI). CAL has historically been seen as the more generic term, encompassing all manifestations of computer-aided learning (Bagley, 1974, p.4). CAI has been seen as a subset of CAL, including only those applications of the computer which most resembled traditional programmed instruction (Hooper, 1975, p.13).

Recently, however, several writers have been more conservative in their application of the term CAL. In an article on the subject of CAI/CAL acronyms, Goldes (1984, pp.353-357) views CAL as only slightly wider in scope than CAI, while in his book on the subject, Davies (1982, p.3) sees the two terms as more or less synonymous.

At the 1983 annual convention of TESOL (Teachers of English to Speakers of Other Languages) in Toronto it was found that a major source of difference in the application of the two terms was geographical. British writers tended to use CAL while American writers used CAI. While a perceived difference in philosophy between the two terms was recognized, "instruction" being seen as "teacher-centred" and "learning" as "student-centred", a general consensus developed that CAL should be standardized as the acronym to use in the field of language learning. (Sanders and Kenner, 1983-b).

Besides CAL and CAI, there is an abundance of other acronyms and abbreviations, some of which fall within the scope of the definition of CAL given above and some of which do not. Usually roughly synonymous with CAL/CAI are computer-based education (CBE) (Bagley, 1974, p.4; Avner, 1978, p.24), computer-based instruction (CBI) and computer-based learning (CBL) (Davies, 1982, p.3), and computer-assisted teaching (CAT) (Kenning & Kenning, 1983, p. x).

Examples of non-CAL abbreviations are computer-generated materials (CGM), computer-managed instruction (CMI), and computer-based testing (CBT). CGM can be seen as a teacher's aid, and not as CAL, because it is the teacher who interacts with the computer, not the student. The student works with the materials that are produced. CMI, by itself, need not include interactive practice on the computer. It can be seen, as well, as a teacher's aid, used for keeping track of a student's progress. CMI "does not instruct but rather guides instruction." (Goldes, 1984, p.355). CBT does not meet the requirement that CAL involve "practice".

Lists of typical CAL activities generally include the following basic classes:

drill and Practice programs, which drill learners on their knowledge of a particular subject,

simulation and modelling programs, where the student engages in a true-to-life situation,

gaming, where there is a competitive interaction between participants (one of whom might be the computer) to achieve a goal, and

problem solving, which requires the synthesis of higher order rules and concepts. (Bagley, 1974, p. 5; Crawford, 1981, pp. 23-37; Cohen, 1983, pp. 10-11).

Drill and Practice is often further broken down into Drill and Practice and Tutorial CAL, the distinction being that Drill and Practice involves only practice while Tutorial CAL involves the presentation of the concept, its practice, and subsequent evaluation of the students' understanding. (Fitzgibbon & Grate, 1970, p. 916; Crawford, 1981, p. 30; Paul, 1982, pp. 6-7; Cohen, 1983, p. 11)

2.1.2 CALL as a special subset of CAL

Before proceeding with the examination of what makes CALL a unique subset of CAL, it is useful to simplify the question by excluding from the discussion the many applications of the computer to language study which do not fit within the definition of CAL offered above. Davies (1982, pp. 1-3) lists a series of language-related uses of the computer which are not CAL; such as machine translation, automatic dictionaries, literary & linguistic processing of texts, word processing, and speech synthesis and analysis. To this list, Higgins (1982, p. 25) adds style editor software such as The Writer's Workbench.

CAL has been applied to the learning of language since the earliest days of its introduction into the domain of teaching. As early examples, Suppes and Mackey (1978 p. 9) cite a first-year Russian program at Stanford in 1967; Scanlan cites (1971, p. 84) the use of the PLATO CAL system for Latin since 1968; and Suppes and Mackey (1978, p. 10) cite the use of the TICCIT system for English at Brigham Young University in 1972.

Applications of CAL to language learning have consistently presented course designers and programmers with "severe technical problems in relation to strategies for coping with the complexity of natural language within the confines of a computing context" (Last, 1983, p. 84). Orstein (1970, p. 215) notes that the ease with which material can be adapted to CAL is in direct proportion to the concreteness and specificity of the material. However, linguists and educators have so far been unable to provide a concrete, specific and workable description of how language works, let alone how people learn it.

Many of the programming approaches popular in CAL are difficult to apply to CALL. For example, "Generative CAL", where the courseware, based on an established algorithm, generates new, unique problems for each student, is a popular tool in scientific disciplines. It is often used "to clarify a particularly difficult concept by means of models and simulations, the performance of calculations, the plotting of graphs, etc." (Davies, 1982, p. 10). Application of this technique to CALL, however, is made difficult, except in carefully constructed contexts, because generation of original language by the computer assumes the computer's ability to analyze the learner's original, possibly mal-formed, open-ended responses.

The parsing of open-ended natural language is considered briefly in section 3.10.4 but lies largely outside the scope of this thesis as its realization is beyond the capabilities of the current generation of special-purpose computer languages and systems under discussion. Hart (1981, p. 9), for example, believes that the application of the very popular TUTOR authoring language to CALL is hindered by the language's poor string (word) manipulation and limited recursive capabilities which make the creation of parsing and language-generation routines difficult. Contrary to the situation in other forms of CAL where developmental work on generative CAL has proceeded much more rapidly, the production of CALL courseware will continue to require the provision, in advance, of data for each discrete interaction.

Another complication in CALL programming is the frequent necessity to provide for situations that often yield multiple correct answers, varying degrees of acceptability of an answer, and the placement of emphasis on either the content or the form of the answer.

The uniqueness of CALL involves more than the simple enhancement of computer programming techniques. It extends to the very nature of the learning experience, which differs radically from that of many of the disciplines to which CAL has traditionally been applied so successfully. Paramskas (1983, p. 6) explains that language learning cannot be measured in terms of the quantity of information acquired, in terms of black and white or right and wrong. The fact that students can sucessfully master a set of exercises does not, in any way, indicate that they have improved their ability to use the language. Crawford (1981, p. 10) proposes that CAL materials must lead students beyond the simple "knowledge" of material towards its integration and the synthesis of new systems of understanding. Nowhere does this seem more important than in CALL, where students can only acquire language skills through the ongoing synthesis of new language systems based on the integration of new, and often conflicting, information. The traditional programmed instruction approach, which still provides the basis of much CAL, where students cumulatively add to their repertoire of "mastered" skills, will not lead inexorably towarda the desired result, the mastery of all aspects of language use, in the same way as it might in, say, biochemistry or physics.

The problem posed by evaluation of learners' performance offers a good example of the CALL design difficulties which stem from the very nature of the language learning experience. In many disciplines, improvements in performance can be measured through a simple comparison of pre- and post-test results and can be linked almost directly to students' competence in the field. Students can proceed step-by-step, mastering one unit before moving on to the next, and, at the end, the computer can assure them that they have "learned" the content of the activity. Language is a much more holistic enterprise, where performance in any one activity is coloured by a host of factors and where mastery of a single "unit" in a language course can in no way be linked directly to improved linguistic competence.

The analysis to be developed in this thesis focuses on the unique programming problems posed by CALL and on the programming implications of the particular nature of language learning.

2.2 Authoring systems

"Authoring" is defined and the historical approaches to authoring CALL courseware are briefly examined. The historical evolution of the term "authoring system" is discussed and a working definition of authoring system is established. Finally, the nature of authoring systems and some of the major factors that differentiate one from another are considered.

2.2.1 "Authoring": A working definition

Authoring, in the context of CALL, is defined here as the process of producing courseware. It involves not only the design and writing of the educational content, but also its subsequent realization in a form deliverable by the computer. The need to produce both a lesson and the computer program that embodies it means that, ideally, potential writers of lessons have to have an ability, not only to communicate their subject matter well, but also to work well in a computer programming environment (Dowsey, 1974, p. 401). It has long been recognized that some type of support is required to bridge the gap between the existing level of computer programming expertise exhibited by most potential "authors" (usually teachers with ideas for learning activities) and the high level of such expertise required to produce the desired final product in the form of computer programs (McCambridge, 1982).

This problem was addressed early in the development of CAL, through the creation of a family of "special-purpose" computer-programming languages designed to simplify the authoring process and referred to as "authoring languages". They first made their appearance around 1960. IBM introduced T.I.P. (Translator for Interactive Programs), the forerunner of the Coursewriter system, and the University of Illinois was working on C.A.T.O. (Compiler for Automatic Teaching Operations), the forerunner of the TUTOR authoring language used in today's PLATO teaching system (Kearsley, 1982, p. 430).

2.2.2 Approaches for authoring CALL

The many options available for the production of computer programs range from the "low-level" or "machine" end of the spectrum to the "high-level" or "human" end. As prospective authors move towards the high-level end of this continuum, the increased complexity and sophistication of the software they are using, the "instrument" with which they actually interact with the computer, allows them to describe in ever more abstract, human terms what they want the computer to do (Boyd, 1970). This increased sophistication is purchased at the cost of pre-defining and limiting ever more the available options. The avenues explored in the production of CALL courseware range, then, from the use of low-level "machine" language, through the use of "general-purpose high-level" languages such as BASIC, through the high-level authoring languages noted above, to the most sophisticated instruments for producing CALL courseware which are called "authoring systems".

2.2.2.1 Machine language

Programming in low-level computer languages, "machine" language or "assembly" language, remains the true domain of computer professionals. Here alone is the programmer actually controlling the computer directly, providing in the instructions a code for each discrete computer operation. "Low-level" programmers must have an intimate knowledge of the structure of the computer. There are, however, no pre-established formats that prevent them from taking full advantage of its capabilities. Little use has been made of this approach in CALL beyond the production of over-the-counter devices such as "Speak 'n' Spell" or pocket translators (Jensen, 1982, p. 50).

2.2.2.2 Original programming using general-Purpose, high-level programming languages

Original programming of educational material using general-purpose, high-level, programming languages (GPL) allows programmers to describe the operations to be performed in terms which begin to reflect human rather than computer realities. One can, for example, can issue the instruction to "print" something, rather than the instruction to "move contents of register C into memory location HB8000". The object of a GPL remains, however, the discrete, step-by-step control of the computer. The syntax and organization of a GPL mirrors the structure of the computer, and not the needs of any particular application or group of computer users, such as educators (Shuyler, 1979, p. 29).

GPL programming represents by far the most widespread approach used for CALL. A 1976 survey showed that six of the ten most commonly used programming languages for CAI were GPLs, the most popular being BASIC, APL, and FORTRAN (Kearsley, 1976). Olsen (1980, p. 345), citing another survey, notes that BASIC is the leading language for CALL, being used 51 times as compared to 21 for TUTOR and 10 for Coursewriter, the latter being "authoring" languages (Olsen, 1980).

2.2.2.3 Shortcuts to the original programming of courseware

In order to reduce the time required for programming and to improve the quality of the resultant computer programs, various "shortcuts" to the completely original programming of courseware have been adopted by programmers.

Chapter Two Background and Definitions

"Structured programming" is a method of programming in which programmers develop independent, modular sub-procedures to accomplish various fundamental operations common to all of a set of programs. Holmes (1983, p. 29) describes the use of this method to streamline programming operations during the development of the CLEF series of French lessons. A series of programming routines specific to language teaching requirements, such as analysis of input, spelling error checks, feedback strategies, and help procedures were modularized and standardized so as to make those aspects of producing the computer programs for the lessons a degree easier and less time-consuming.

"Programming Aids" or "Authoring Utilities" are sets of such procedures, specially designed for CALL, which are available in the marketplace. They are usually programmed in low-level computer languages so as to be very efficient and rapid during execution, and they often exhibit a level of sophistication beyond that of many would-be CALL programmers. Jensen (1982, p. 52) points out that "Utilities reduce the drudgery of some kinds of programming by turning over the tedious, error-prone aspects of coding to the computer". He cites the example of Apple PILOT utilities for graphics, sound effects, and foreign character sets. EnBASIC (Enhanced BASIC) is another example, being a set of machine language routines that the BASIC program can call upon for sophisticated and otherwise time consuming operations such as the mechanical analysis of a student's answer and the noting, via proofreader's symbols, of the discrepancies from the correct answer (Chapelle & Jamieson, 1983).

"Template" programming is a time-saving technique that involves the separation of the "logic" of the program (the type of presentation and the logical structures, such as answer-handling routines) from the "data" (the course-specific content) (Boyd, Keller & Kenner, 1982, p.110). Authors can create new lessons by fitting new data into old programs, so long as the new content can be accommodated to the logic of the original program (Dowsey, 1974, p. 403). Teachers can thus insert their own pedagogical materials into pre-determined lesson formats such as multiple-choice, fill-in-the-blank, transformation, cloze tests, etc. (Holmes, 1982, p. 12). A template program is, in effect, an "editable CALL lesson". The programmer simply replaces the data and content statements of one lesson with new content, yielding a different but algorithmically identical clone. To be effective, the original template program must have been designed so as to offer some degree of flexibility and so as to facilitate the editing process (Jensen, 1982, p. 51).

"Driver" programs are still more sophisticated. These programs are neutral, and do not have to be edited in order to produce a new lesson. All lesson-specific content is stored on an external data-base and is presented to the student in a pre-established format (Shuyler, 1979, p. 33). This approach is not new. As long ago as 1970, Fitzgibbon and Grate (1970, p. 921) described what was in essence a cloze-generator, a program to delete every nth word from a text externally in data then allow the student to fill it in. Many driver-based CALL systems provide an option which allows for the creation of the external data bases by teachers and other non-programmers.

2.2.2.4 Authoring languages

"Authoring languages" (AL) can be described as "special-purpose" programming languages because they are designed specifically for one application: the coding of CAL programs (Ashmore, 1983, p. 4). Authoring languages are sometimes called "very-high level" languages because several GPL instructions may be required to accomplish what can be covered by a single AL instruction (Shuyler, 1979, p. 30). "They represent an attempt by their designers to predict the types of commands and capabilities educators will need" (Wyatt, 1983a, p. 36). Essentially, the ready-made, CAL-specific routines described above under "authoring aids" have been incorporated into the language and can be invoked with a single program instruction. Programmers are thus offered features which they would have to design from scratch if they were using a GPL (Wyatt, 1983-b, p. 7). The convenience and simplicity of ALs, however, is achieved at great cost in versatility (Jensen, 1982, p. 51; Higgins & Johns, 1984, p. 102; Wyatt, 1983-b, p. 7).

Wyatt (1983-a, p. 37) offers the following example of the convenience vs. versatility problem: Apple PILOT contains a powerful feature which allows for the easy creation of special characters, such as accented letters. The drawback is that each new letter must be allocated its own key on the keyboard, and the result is a confusing cross-reference of individual keys. Designing the routine from scratch in a GPL, while more difficult, would allow for the possibility of combining keystrokes so that a simple rule like "press letter-key followed by accent-key" could be taught to the student.

Merrill (1982, p. 70) offers the framework for summarizing the key features of ALs. 1) They reduce the domain of possible commands and structures. (In other words, they make the language do less.) 2) They provide commands and structures which meet the specific needs of instructional applications. (That is, they second-guess the needs of educators.). Finally, 3) they provide commands or routines which perform high level tasks. (Thus, each command does more.)

2.2.3 The definition of "authoring system"

An "authoring system" (AS), in the context of CALL, is a computer software package through which teachers can create language-learning activities without having to do any actual computer programming (Pogue, 1980, p. 58; Merrill, 1982, p. 77; Kearsley, 1982, p. 429; Wyatt, 1983a, p. 37; Higgins & Johns, 1984, p. 102). This removal of the need to do any actual programming of the computer is central to the definition and is the primary factor which distinguishes ASs from other related software (Higgins & Johns, 1984, p. 10).

There remains, nevertheless, a great deal of confusion in CALL literature over the nature of ASs (Wyatt, 1983a, p. 37). There is confusion over the family of software to which ASs should be ascribed. For observers ASs are an extension, somewhat more sophisticated but of the same species, as authoring <u>languages</u> (Villeneuve, 1984), while, for others, they are an extension of template- and driver- programs (Jensen, 1982, p. 51) Paramskas (1983, p. 5) states that the only consideration setting templates apart from true authoring systems is that the generation of new lessons from existing templates still requires the intervention of programmers. Indeed, Holmes (1983-b, p 22) points out that some of the simplest ASs are little more than template programs in which the entry of new lesson material has been made interactive. A rigid application of the no-need-to-program criterion can serve to clearly separate the two types of software.

The distinction between ASs and other avenues of courseware production is often further blurred by the practice of many AS designers to resort to exterior, independently- designed procedures programmed in GPL for the handling of sophisticated, specialized applications (Pogue, 1980, p. 65; Holmes, 1983). The ability to reference external, GPL procedures renders much more difficult the determination of the range of procedures that a particular AS is inherently capable of. The AS becomes capable of anything that is programmable. The no-need-to-program criterion of the above defintion of AS removes the need to consider what might be possible using external GPL procedures.

A lingering confusion over terminology stems from changes in the application of the term "Authoring system" over the years. When first used, it referred to an entire educational process which started with an analysis of the needs of the target population, proceeded through the design and programming stages, and ended with the evaluation of the material against stated goals. By the early 1970's, the term was being used mainly to refer only to the process of translating lesson-content into computer-program. 1974 finally saw it used as a label for the computer software that facilitated the process. (Pogue, 1980, pp. 57-58; Zinn, 1974, pp. 381-384) By that time, there were at least 64 different examples of this type of software in use (Braun, 1973, p. 1).

2.2.4 The nature of authoring systems

The prime criteria of ASs are that they should be easy to learn, simple to use, and should demand of the user no more than a fundamental level of computer literacy (Pogue, 1980, p. 6; Wyatt, 1983-a, p. 37). Pogue (1980, p. 58) lists, in addition to this, four other desirable characteristics of an authoring system:

1. a wide variety of instructional strategies should be available to authors. Authors should be able to create new strategies and modify existing ones;
2. an AS should be easy to use with all instructional

strategies, whether simple or complex, and by all authors, whether novices or highly experienced;

3. an AS should allow for the use of the full range of possibilities offered by the computer, from sophisticated answer-analysis and complex branching to the use of audio-visual media;

4. authors should be able to test and modify their material easily.

A typical AS could be described as follows. Teachers work in an interactive mode to create CALL material. They sit before the computer and respond to the software's prompts as these appear on the screen. Teachers indicate the number of items in the exercise, their sequencing, the explanations to be offered, the number of tries the pupil is to be allowed, etc. They then provide the material, stimulus, correct answers, predictable wrong answers, and so forth, for each of the items. When finished, the AS software makes up the exercise in accordance with the material and instructions that were originally entered. (Barker &Singh, 1982, pp. 167-196; Edwards & Tillman, 1982, p. 19; Jensen, 1982, p. 51; Kearsley, 1982, p. 431; Kenning & Kenning, 1983, p. 11.)

2.2.5 Factors differentiating authoring systems

Authoring systems differ from one another in three ways: the manner in which the user interacts with the software during the entry of course material, the manner in which the software formats the material for later delivery by the computer, and, most importantly, the degree to which the user can alter the pre-established pedagogical structures inherent in the software.

2.2.5.1 The entry of data

Data entry can be "on-line" (entered directly by the teacher seated before a computer) or "off-line" (written on paper by the teacher and entered into the computer by a computer professional). During the data-entry phase, the AS software can proceed in an "interrogative mode", prompting the teacher to provide each piece of infomation, or it can allow the user to simply type in the required data, without offering prompts.

Today, most authoring systems are interactive computer programs, but this has not always been the case. Many early ASs used elaborate coding sheets which required authors to prepare their material "off-line" in a format which facilitated data entry. The data which represented lesson material was then entered by computer operators. (Dowsey, 1974, pp. 404,410). It was only as access to computer terminals became more widespread that the convenience of "on-line", interactive entry of course material began to be appreciated (Zinn, 1974, p. 388).

Zinn (1974, p. 391) finds that, while the interrogative mode seems well suited to the beginning user, it often frustrates the more experienced author. What is easy for the beginner may be constraining and time-consuming to the professional. He submits that, ideally, an AS should provide for both interrogative entry of material by beginners, as well as off-line preparation by more experienced users. Paloian (1974, p. 442), as well, opts for two levels of interaction, interrogation for the inexperienced author and straight entry of data for the experienced author.

The manner in which lesson data is entered is part of the larger question of "user-friendliness". Pogue (1980, p. 62) believes that the level of assistance should be determined by the author. When interacting with beginners, the AS should guide authors through the entire entry process, prompting them for information, checking for structure or format mistakes, and allowing them to ask the system for additional "help" if they do not understand the next step.

2.5.2 The formatting of lesson material

Once authors have provided the course content, ASs can realize the final product by one of two mechanisms, as "drivers" or as "code generators". With the "driver" system, a part of the AS software system elicits the data from the author and places it onto a data file. Another part of the software, the driver itself, "reads" the file and presents the material to the students. Code generators, on the other hand, actually "write" new, separate computer programs based on the content data elicited. Once written, the lessons are independent of the AS that created them. (Pogue, 1980, p. 65).

The literature does not clearly indicate a preference for either approach, insofar as the pedagogical effectiveness of the product is concerned. Rather, the mechanism chosen affects factors such as transportability of courseware across different types of computer, and memory storage requirements, which only impinge upon pedagogical effectiveness in extreme cases.

2.2.5.3 Design options offered to the author

Most important to the purpose of this thesis is the range of options an AS offers to authors during design of the course material. Again, there are two fundamental approaches, the "paradigm" approach and the "toolbox" approach (Hart, 1981, p. 7).

An AS which uses the paradigm approach calls upon the author to fit lesson content into pre-established instructional formats, such as matching or multiple choice. This approach is resembles the interactive templates described earlier. Jensen (1982, p. 51) finds that the convenience and ease of use offered by this approach is achieved at great cost to flexibility, because of the fact that the underlying format and strategy remain fixed. Merrill (1982, p. 77), however, considers that the lack of flexibility of any one template can be offset if the AS provides a wide range of paradigms for different types of learning.

Kearsley (1982, p. 434) believes that AS should allow authors to specify "instructional logic and strategy" as well as content. The "toolbox" approach offers this more flexible possibility. In this system, the author begins by describing to the AS the structure of the lesson and the pedagogical strategy to be used. Such description is made through the use of high-level "verbs". These verbs call into play a range of previously programmed basic strategies called "macros", which the author can further modify through the specification of "arguments". Once the logic of the lesson has been established, the AS elicits the actual content, based on the requirements of the chosen strategy (Paloian, 1974; Dowsey, 1974, p. 405; Dean, 1978, pp. 21-23).

As already incicated (Pogue, 1980, p.58), it is desirable that an AS be easy to use by beginners while providing for the realization of the full range of computer possibilities. These two vectors often move in opposing directions. Using an AS can sometimes be even more complex than using programming languages. Indeed, as the user is forced to provide "verbs" in order to design the lesson, there is a danger of blurring the distinction between programming language and authoring system . In other cases, the AS is so unsophisticated that it forces authors to answer an endless series of questions, without being able to produce anything worthwhile (Villenueve 1984, p. 5).

Jensen (1982, p. 50) and Dean (1978, p. 23) echo the warning that sophisticated authoring systems, although very versatile, may become too cumbersome to use and the degree of flexibility may be more than some users can handle. Dean opts for a system more closely following the paradigm approach in which "pedagogical structure, instructional logic, and the nature of the machine-student interfaces (are) defined separately from the course content (and are) provided by professionals" (Dean, 1978, p.23).

Background and Definitions

Merrill (1982, p. 77) opts for an AS which allows authors to begin with a simple subset of instructions and capabilities. As authors become more adept, they can learn additional commands which will give them additional capabilities and allow them to develop more sophisticated courseware.

Kearsley (1982, pp. 432-434) proposes, as an optimum situation, four levels of interaction with the AS software. The lowest level would involve authors in a paradigm-authoring environment, where they would be required only to provide content for pre-established, fixed templates. In the next level, authors would be able to alter the parameters of the established template in relation to 1) how information is to be displayed on the screen, 2) how answers are to be analyzed, and 3) how instructional segments are to be linked together (branching). The third level would allow authors to create new templates through the use of high-level verbs. A final level would allow authors to control the nature of the AS interaction itself, by opting, for example, to turn off the interrogative input format.

2.3 The Importance of authoring systems

The current state of CALL is examined and the serious problem posed by the continued lack of a sufficient quantity of quality material is explored. The terms "quality", "quantity", and "flexibility" are defined and the various approaches to producing CALL courseware outlined in section 2.2 are then re-examined with regard to these three factors. Finally, authoring systems are shown to provide a mechanism for overcoming the shortage while offering the potential for quality materials.

2.3.1 The current state of CALL

CALL has developed slowly since the earliest research in the mid-sixties. The literature is unanimous in outlining the three major factors which have acted against the widespread use of CALL, or CAL, or any use of high technology in education: the high cost and contrary nature of the equipment, negative teacher attitudes, and the shortage of teaching material.

The early epoch of CALL research has been called by Paramskas (1984, p. 14) the period of "monster machines". CALL experimentation was limited to those few large institutions which could absorb the high cost of computer access. The equipment of the day, consisting, for the most part, of noisy teletype terminals connected to remote, seemingly mysterious, and rather undependable main-frame computers, was considered to be "unfriendly" (Paramskas, 1984, p. 14). Language teachers were found to have a generally negative attitude towards the use of computers in language teaching (Fitzgibbon & Grate, 1970, p. 921). Those teachers who decide to investigate the possibilities of CALL, and whose parent institutions could afford to offer the computing resources, were faced with a drastic shortage of useful and usable materials (Orstein, 1970, p. 216).

The complicated procedure of passwords and account numbers of the earlier main-frame epoch has given way to the much more democratic and non-threatening procedure of simply switching on the inexpensive microcomputer and seeing it instantly respond, "READY" (Higgins & Johns, 1984, p. 10).

With the adverse factors of cost and the unfriendly nature of the equipment eliminated, the current euphoria among educators would seem to indicate that teachers have developed a positive attitude, which should lead to a rapid increase in the use of CALL. Such a conclusion may be premature and illusory, however, as the course and direction of CALL, indeed, its very future in language teaching, remains far from settled. The desire to move from the experimentation phase of CALL into the implementation phase, now that many of the other factors which inhibited the use of computers have been resolved, faces a major remaining obstacle.

Experts are almost unanimous in warning that widespread implementation cannot proceed until the serious shortage of quality teaching materials, or courseware, for the medium is resolved. The supply of sound, substantial CALL courseware is not much greater than existed before the "computer revolution" and this lack of material represents the major barrier to widespread use and acceptance of computers in language teaching. (Crawford, 1981, p. 3; Davies, 1982, p. vii; Barker & Singh, 1983, p. 174; Holmes, 1983, p. 21; Stevens, 1983-a, p 293).

Centres wishing to implement CALL have essentially only two options: to make use of existing commercial materials or to create new material. The first option is inadequate, however. Commercial sources offer only a very limited number of computer programs for language learning (Kenning and Kenning, 1983, p. 10). What is available is often of mediocre quality and even what "good" material there is frequently requires adaptation (Wright, 1980,

p. 11). The list of available material is even further reduced by the inability of most programs to run on any computer except the one on which they were designed and programmed on, a situation likened to the "Tower of Babel" (Wyatt, 1983-b, p. 7). Wyatt thus concludes, "With microcomputer-based CAI, it is very likely that an institution will have to spend at least some time developing its own courseware" (Wyatt, 1983-b, p. 7)

The local creation of courseware by teachers is also seen as a necessary factor in maintaining a positive attitude towards CALL among language teachers. Many language teachers have felt left out by the computer revolution, a situation that will be put right only when teachers are given the tools with which to exploit computers on their own (Higgins & Johns, 1984, p. 10).

Often cited among CALL experts as a "worst-case" scenario is what can be referred to as the "Language Lab Analogy" (Davies, 1982, p. vii; Alatis, 1983, pp. 11-12; Kenning & Kenning, 1983, p. 1; Holmes, 1983-b; Sanders & Kenner, 1983-a, p. 33; Higgins & Johns, 1984, pp. 11-12). It is feared that, as with the language lab of the 1950's and 1960's, the current wave of optimism about the possibilities of CALL will result in educators attempting to make heavy use of computers in the language teaching curriculum. Results will be disappointing, not because of any basic shortcoming in CALL as a medium, but because of the lack of quality courseware to use with the computers and because of teachers' inability to participate actively in the creation of "home-grown" material. Weible (1983, p. 64) points out that language labs began to experience a rebirth in popularity once cassette recorders became familiar everyday conveniences and teachers could make their own recordings. Unless a similar development occurs in CALL, disillusionment may set in and CALL will be discredited as a force in language teaching.

2.3.2 Factors affecting the quantity and quality of CALL courseware

There is a measure of agreement that the resolution of the coureware shortage problem involves two factors, the quantity of courseware and the quality of that courseware. These terms, "quantity" and "quality", are used very loosely in the literature and need to be more closely examined, and their role in the present discussion more rigorously defined.

The quantity of production of CALL courseware is a function of the cost, in terms of time and resources, of designing and programming CALL material (Barker & Singh, 1983, p. 174). Figures of anywhere from 50 to 500 hours of development-time per hour of finished product using a GPL approach are cited. (Braun, 1973, p. 2; Kearsley, 1982, p. 430; Ashmore, 1983, p. 6). It would seem that any method of speeding up the development process would be laudable, providing quality can be maintained.

Speeding-up the production of material, however, often affects quality adversely. Gillespie (1983, p. 6) sees this relationship as a direct one, arguing that the more time and and money available for production, the better is the quality of the product. Orstein (1970, p. 215) counters this view by citing the millions of dollars that government and private foundations poured into massive CAI projects in the late sixties, not always yielding a positive effect on quality.

The relationship between quantity and quality must be more complex than Gillespie indicates above. The question of speeding-up or facilitating the development process can relate to either speeding-up the design process or the coding process or both. Each expedient will inevitably affect the final quality of the product.

It was pointed out earlier that CALL courseware is really a combination of two separate factors: the logic of the program, which can be called the "vehicle", and the content of the lesson. The two are often confused. Dean (1978, p. 20) points out that, despite the fact that over 90% of a CAI author's time and effort is expended in activities which are unrelated to computer coding procedures, the question "How do I write a CAI lesson ?" is often interpreted to mean "How do I code a lesson into the computer ?"

Discussions of quality must take into account both the quality of the pedagogical content and the quality and sophistication of the vehicle. (Holmes, 1982, p.12). The interrelationship of these factors is addressed below.

Two important questions arise in any discussion of the quality of pedagogical content: 1) Who should design the material, regular teachers or those specially trained in CALL techniques? and 2) To what degree can CALL can be adapted to current language teaching philosophies.

One school of thought holds that effective materials cannot be created by teachers alone, that people trained in CALL techniques must be involved in the design process (Edwards & Tillman, 1982, p. 20; Stevens, 1983-a, p. 294).

The skills required to generate effective courseware are said to be different from those required of an effective teacher (Joiner, 1982, p. 13) and failure to provide the necessary expertise results in the sort of unsound CALL material widely found today (Dimas, 1978, p. 27).

Teachers' difficulties stem from their inability to fully appreciate the possibilities of the new medium. Computers represent a level of complexity far greater than that found in devices such as tape recorders, for which teachers can often prepare a good lesson in single day (Marti, 1981, p. 26). Many teachers, as authors, still adhere to the approaches of these other, more familiar media, producing lessons that tend to look like lessons prepared for presentation in textbooks or on blackboards (Stevens, 1983-b, p. 28). Teachers fail to fully appreciate or exploit the power of the computer: they are unable to see beyond a linear sequencing of material and tend to want students to complete all of the material provided (Hart, 1981, p. 14). Stevens (1983-b, p. 28) finds the situation similar to the phenomenon exhibited in early educational films, where teachers were shown standing in front of blackboards.

Adherents of the opposing viewpoint dispute the notion that professionally produced commercial materials are necessarily any better than teacher-produced materials, even going so far as to claim that the latter are often superior in quality (Crawford, 1981, p. 16). According to this thesis, only experienced language teachers can be relied upon not to make pedagogical errors as well as errors in actual language usage (Allen, 1972, p. 349). It is therefore probably easier to teach language teachers about CALL than it is to teach CALL "experts" about language teaching (Wilson, 1982).

The training of teachers to understand and take advantage of the full range of possibilities offered by CALL does not relate directly to the present discussion. Nor does the ability of an authoring system to guide teachers through the authoring process and to provide assistance as to the various CALL options available at each point. Both of these factors are important, however. The former is clearly one of the most important variables in any discussion of the preparation of quality courseware, while the latter may be an important criterion in the selection of an authoring system for teachers not already familiar with CALL.

Kenning and Kenning (1983, p. 5) do not consider that there is any incompatibilty between CALL, in general, and the present emphasis on meaningful, communicative practice which forms the basis of current ideas of quality courseware. If the ability of teachers, in general, to fill the role of qualified, knowledgeable authors is assumed, then the remaining variable is the degree to which the quality and sophistication of the software vehicle used to create CALL lesson material, the authoring mechanism, affects and colours the nature and quality of that material.

The term most often used in describing the quality and sophistication of a authoring mechanisms is "flexibility", which can be defined as "the number of choices available to the instructor and the learner" (Paul, 1982, p. 11).

Again, there are alternate schools of thought as to the effect of flexibility. Merrill (1982, p. 77) and Kearsley (1982, p. 435) believe that the flexibility of the vehicle determines the quality of the material it presents, while Holmes (1982, p. 49) and Wyatt (1983-b, p. 10) hold that flexibility need only have a minor effect on quality. The latter hold that valid, effective courseware can be produced using very simple, unsophisticated mechanisms;

that it is not necessary to exploit the full range of features available on the computer.

Both arguments have some degree of merit, but there are subtle, confounding factors which must be taken into account. Authors with a deal of imagination and skill will be able to create excellent activities with even the simplest, least flexible vehicle while other, less gifted authors may allow a poorly designed vehicle to impose or encourage poor lesson design (Joiner, 1982, p. 52). A highly flexible authoring mechanism allows for higher quality material, without being able to guarantee it (Joiner, 1982, p. 52; Paul, 1982, p. 11). The highly sophisticated system may provide options which many authors may not know how to take advantage of (Joiner, 1982, p. 52).

2.3.3 The quality and quantity of courseware production as a function of the approach to authoring

Of the three major approaches to the programming of CALL courseware, 1) using general purpose programming languages (GPLs), 2) using authoring languages (ALs), and 3) using authoring systems (ASs), only the latter shows promise in providing a mechanism for the production of a sufficient quantity of courseware of sufficiently high quality to be accepted in the field. The two former approaches, which have been in use since the beginning, have failed on one or both counts.

2.3.3.1 Production using general purpose languages (GPLs)

While original programming in a GPL is the most time-consuming approach to the coding of educational content, the flexibility offered by this approach makes it the most popular (Stevens, 1983-a, p. 294; Wyatt, 1983-a, p. 38) and, as has been shown, it accounts for most of the CALL material produced to date.

Productions using this approach fall into any one of four main categories: 1) Courseware conceived by those trained in programming, but without a language teaching background; 2) Courseware conceived and programmed by language teachers newly trained in the rudiments of computer programming techniques; 3) Courseware conceived by language teaching professionals and programmed by experienced programmers (the "team" approach); and 4) Courseware conceived and programmed by a language teaching professional who was, at the same time, an experienced programmer (the "great man" approach).

There is little debate among CALL professionals about the need to exclude materials designers of the first category, that is, those who have a knowledge of computer programming alone and expertise in neither language teaching nor CALL methodology (Davies, 1982, p. vii). Productions realized in this category, unfortunately, account for a good deal of what is commercially available and contributes to the shortage of "quality" courseware by crowding out better material (Stevens, 1983-a, p. 294; Gillespie, 1983, p. 7). On the basis of the discussion of the unique nature of CALL (in section 2.1.2 above) and the discussion of who should design CALL courseware (in section 2.3.2 above), those prospective courseware designers who are experts in the field of CAL alone, but who are not experienced in language teaching must be included in the present category.

The recent exaggerated claims for the "computer revolution" in language teaching has awakened among teachers the desire to become involved in CALL and has given great impetus to the second approach, the programming of courseware by language teachers newly trained in the rudiments of computer programming. Proponents of this approach argue convincingly at teaching workshops that teachers can learn in a few hours all the computer-programming skills they need in order to produce simple language activities. Many of those who venture into programming language activities find, however, that it is much more difficult to produce acceptable results than they had been led to believe.

There is no doubt that GPL programming offers, in theory at least, the greatest flexiblity in designing CALL materials (Wyatt, 1983-a, p. 35). The only constraints limiting the possible design of activities are those imposed by the computer, the computer language, and the programming abilities of the designers (Holmes, 1983, p. 27). While the constraints imposed by the computer are most often blamed by amateur authors for the ultimate shortcomings of their material, their own lack of programming expertise is usually the prime limiting factor (Hazen, 1982, p. 22).

Davies (1982, p. 44) notes that programming is essentially a creative art in which programmers can give free rein to their imagination. The key word is "art". Knowing how to paint or how to play an instrument is not the same as being a gifted painter or musician. Many experts agree that GPL programming is only as flexible as the creativity and skill possessed by the practitioner (Kearsley, 1982, p. 429; Otto & Pusack, 1983, p. 26). Language teaching is sufficiently demanding that it is impractical for teachers, en masse, to seek to become accomplished computer programmers as well (Pogue, 1980, p. 64; Merrill, 1982. p, 70).

The second approach to GPL programming has thus failed to produce either the quantity or the quality of CALL courseware required. Not enough teachers have become involved in authoring materials, because of the immense learning burden of GPL programming. Of those who did succeed in learning to program, their lack of sophistication in programming has had a negative effect on the quality of their materials.

The approach to GPL programming favoured by most experts is the "team" approach, where there is a collaboration among specialists, each providing input in their respective area of expertise (Crawford, 1981, p. 15; Marty, 1981, pp 43-44, Mydlarski, 1981, p. 119; Davies, 1982, p. 44). The team approach to courseware development can involve as little as a faculty member designing content material and having it programmed by a student programmer (McCambridge, 1982). The process may include a third party, the educational technologist or CALL expert (McGee, 1982).

Chapter Two Background and Definitions

Among the courseware accepted as being of higher quality is much that has been produced using a team approach. The _Clef_ series, a set of beginner's French grammar lessons, was produced through a collaboration between teachers and professional programmers (Holmes, 1981), as was the _Grammar and Vocabulary Mastery_ series produced by the American Language Academy (Wyatt, 1982). Similarly, the production of courseware at Concordia University's Learning Laboratories has been made possible through a collaboration of interested teachers, who develop the content, and the Learning Laboratory staff, who design, code, and test the computer program (Kenner & Richards, 1980).

The team approach, while responsible for the production of some quality courseware, has failed to satisfy the demand for the delivery of quality courseware in massive quantities. The most severe disadvantage is the high cost of hiring professional programmers (Pogue, 1980, p. 64; Kenning & Kenning, 1983, p. 145). In situations where programming resources are minimal, faculty may have to concentrate on less sophisticated CALL (Dimas, 1978, p.27). In the case of Concordia University, instructors may have to resign themselves to a very slow rate of production and long delays between the design of the content and its subsequent realization as a computer program.

Complications in the "team" approach arise out of the need for the various parties to communicate when each may know little or nothing of each other's field of competence (Kenning and Kenning, 1983, p. 145). Elliot (1982, pp. 6-8) does not believe such a situation can produce effective results, unless each party is willing and able to learn about the discipline of the other. For teacher - programmer collaboration to succeed, teachers must know enough about programming to understand what level of performance is to be expected from programmers and to communicate to programmers their requirements on content and format. Without the added expertise of a CALL specialist to act as intermediary, the team approach begins to suffer from some of the same constraints as those noted above for teachers learning to program on their own.

A few institutions are extremely fortunate in that they have individuals who are, at the same time, language teaching experts, experts in CALL methodology, and accomplished programmers. Dimas (1978, p. 27) refers to this as the "great man" approach and it represents the fourth option to GPL programming cited at the beginning of this section. Barker and Singh (1982, p. 169) note that this enviable combination of cross-disciplinary skills is extremely rare. Again, there is no guarantee that what is produced will be of the quality desired. The courseware will reflect the particular author's personal preferences and it may very well be found that other members of the teaching faculty are reluctant to use it (Dimas, 1978, p. 27). Often, when the "great man" moves on, or loses interest or motivation, an institution's CALL programme collapses and the material passes out of circulation.

GPL programming, then, despite the fact that it accounts for a majority of what has been produced, has failed to satisfy the double demand for quantity and quality in courseware. The first and the second approaches have led to the production of a large amount of material, but often of inferior quality. The third and fourth approaches have had limited impact in terms of quantity because of the high cost and/or lack of available experts, still with no guarantee of quality.

2.3.3.2 Production using authoring languages (ALs)

Authoring Languages (ALs) have been advanced as the final solution to the courseware shortage situation. It was suggested that teachers who knew nothing about programming and little about computers could sit down at the computer and dash off computer-based exercises quickly and with great ease. It was supposedly no more difficult to learn to type than it was to learn to use ALs (Allen, 1972, p. 348), yet they were to offer a sophistication equal to that of material produced by accomplished programmers (Merrill, 1982, p. 76). ALs were designed with non-computing, liberal arts people (such as language teachers) in mind (Otto & Pusack, 1983). ALs were to be languages oriented towards the specific needs of instructors, and were to create an environment where individual instructors were freed from technical, programming concerns. (Hazen, 1982, p. 21).

The question inevitably arises as to why ALs have had so little impact on the production of CALL materials, given these apparent advantages, the disadvantages of GPL programming, and the fact that ALs have been around almost since the inception of CAL.

There are several key criticisms of ALs which must be examined. Universal to all ALs is the observation that they do not live up to their rationale of being easy to learn and use. Possibly a function of the specific AL under discussion is the criticism that ALs place limits on flexibility, with little apparent gain in other areas;

that they tend to steer authors in the direction of less effective pedagogical strategies; and, finally, that there are technical aspects of their operation which limit their effectiveness.

Most non-programming teachers who have endeavored to use ALs would not agree with the claim, often made by proponents of ALs, that only an hour or two of initial training is needed before courseware can be produced (Pogue, 1980, p. 64; Davies, 1982, p. 39). The claim often refers only to learning the essential operation of the AL. A great deal more learning and practice-time is required before the user is able to exploit the capabilities of the language at a moderately sophisticaed level (Wyatt, 1983-a, p. 36). The simplicity and ease of learning which was initially proposed as representing one of the foremost arguments for using an AL over a GPL (Merrill, 1982, p. 76) has been traded for continued increases in sophistication. This has resulted in ALs which are as complex as GPLs and take as long to learn to use effectively (Hazen, 1982, p. 22; Jensen, 1982, p. 51; Merrill, 1982, p. 76; Wyatt, 1983-a, p. 36).

It is beyond dispute that it takes less time to produce CALL materials using an AL rather than a GPL.

Wyatt (1983-a, p. 36) warns, however, against overstating the speed advantage. The programming time will be less, but of the same order of magnitude. Stevens (1983-a, p. 294) cites 30 hours of programming per hour of finished product using an AL, which can be compared with the 50 hours and upwards cited in section 2.3.2.

The claim often made by proponents that ALs are for those who do not know how to program and that programming using an AL is, somehow, not really computer programming is also disputed (Jensen, 1982, p. 51). Paloian (1974) reminds us that, while ALs may facilitate the actual coding of a CAL program, they do not contribute much towards shortening the program-planning stage or the debugging of a program once it has been entered into the computer. Hardy & Elliot (1982) say of MicroPilot that it "demands the same level of ability and skill [as that] required of a BASIC programmer". The difficulty lies not so much in the programming language itself, as in the need to understand programming concepts like variables, counters, flags, matching, conditional and unconditional branching, etc. (Davies, 1982, p. 39).

Indeed, the problem is often hightened by the need, in any case, to program certain sections of a CALL program in a GPL. Many ALs allow for the inclusion of routines written in the host language (the computer language in which AL was originally written). The author is thus given powers of operation unforeseen by the designers of the AL. Merril finds, in reviewing a PILOT program, that almost half of the statements were in the host language (Merrill, 1982, p. 76).

The second major cristicism of ALs is their lack of flexibility. An AL attempts to simplify the author's task by reducing the number and complexity of commands, but "the restriction in the domain of commands creates a restriction in the range of possible outcomes or applications" (Merrill, 1982, p. 76). Teachers, through the lack of flexibility of the given AL in fostering development of truly innovative activities, will either produce only simple exercises (Wyatt, 1983-a, p. 37) or they will recognize the limitations of the AL quickly and will wish to move on to GPL programming (Kenning and Kenning, 1983, p. 13). Davies (1982, p. 44) sees an analogy in the "paint-by-number" kits which please the newcomer to oil-painting but cease to satisfy the creative needs of those who wish to carry their artistic endeavours to a higher level.

The inflexibility builds upon itself. Teachers will be limited by their lack of programming experience and may decide that certain things that they wish to do are impossible on the computer. This situation will then be compounded by the inability of ALs to offer avenues for innovative teaching techniques and by their tendency to push teachers towards those pedagogical strategies which are the easiest to program. Otto & Pusack (1983, p. 27) have concluded, by virtue of this inflexibility, that ALs are not suitable to most forms of language practice.

The tendency of the software to influence the selection of pedagogical strategies is inescapable. Inherent in any very high level software such as ALs are the assumptions and decisions made by the original designers within which (or against which, as the case may be) the user has to work. Dean (1978, p. 23) makes the point that even a basic sequence like "presentation followed by decision" implies an instructional logic. ALs are criticized because they lead novice authors to follow a particular, restrictive instructional strategy since the restriced domain of commands makes this strategy easiest to implement (Hazen, 1982, p. 21; Merrill, 1982, p. 76).

There are technical considerations which limit the pedagogical utility of some ALs for CALL applications. Some ALs have been observed to have an immense "overhead" (in terms of computer memory and resources) which acts on microcomputers to trivialize the type of exercise that can be designed for the little remaining room in memory. This memory waste is compounded by the tendency of ALs to guide would-be programmers away from strategies of sound, economic programming such as structured programming or the re-use of generic procedures. Often only parts of the AL, or the program, or both, can be loaded into the microcomputer at one time, leaving students to face long delays as the computer loads data from external storage media. "Students are lulled by the endless whirring of the floppy disk" (Gillespie, 1983, p. 7).

Another technical consideration is portability, the ability to use material on computers other than the one for which they were designed. When good courseware is produced using an AL, it is often not available to the entire language teaching community because the AL in which it was written is supported only by certain computers or must be purchased separately. This is also, of course, a problem for GPL programs. It becomes more acute when an AL is involved, because there is less chance of finding a version of an AL compatible with a given machine than there is of finding GPL versions (Hazen, 1982, p. 21; Otto & Pusack, 1983, p. 27).

2.3.3.3 Production using authoring systems (ASs)

Many of the criticisms levelled against ALs can also be made against many authoring systems (ASs), with the notable exceptions of 1) the initial learning burden, 2) the time taken to produce material, 3) and the need to understand computer programming. There is general agreement that ASs offer a dramatic reduction in the time necessary for CALL courseware development, both in the initial training of prospective authors (Wyatt, 1983-a, p. 37) and in the time needed to produce an exercise (Pogue, 1980, p. 62). By definition, use of an AS does not require an understanding of computer programming.

Merrill complains that ASs "reduce the cost and effort by reducing variety in much the same way that cost and effort are reduced in fast-food restaurants, tract homes, and formula television shows." (Merrill, 1982, p.77) To carry this analogy further and to relate it to the present discussion, it can be argued 1) that the above did lead to a rapid expansion in the number of restuarants, homes, and television shows and 2) that not all fast food restuarants, tract homes, or formula television shows are uniformly bad.

While Merrill (1982, p. 77) concedes that ASs can enhance the quality of courseware produced by novices to CALL, he, and others (Edwards & Tillman, 1982, p. 20; Davies, 1982, p. 35) nevertheless hold that ASs restrict the quality of the courseware that could be produced by truly creative authors. Few would dispute Davies' (1982, p. 35) observation that ASs can never produce the same standard of CALL software as a team of expert teachers, CALL experts, and expert programmers using a GPL.

The reasons why the team approach has not satisfied the demand for material have been explored above (in section 2.3.3.1). Without such a team, Merrill's "creative authors" would probably find the range of possibilities in a GPL or AL environment no more satisfying or unique than an AS. The details in programming CALL materials tend to overwhelm print-oriented authors. Rarely do they have to concern themselves, when writing a book, with such issues as paper choice, page design, typesetting, binding, and a myriad of other activities involved in the production of their work. The details of converting a manuscript into a book are left to the publishing firm (Edwards & Tillman, 1982, p. 20).

After a fashion, an AS provides analogous support during CALL courseware design. The degree to which the support becomes a strait-jacket depends on the AS in question. "If the system is reasonably flexible, the teacher will have a fair degree of latitude on any given point" (Kenning and Kenning, 1983, p.12).

Wyatt (1983-a, p. 38) claims that ASs will not respond to the demand for quality CALL courseware because it is impossible to use them for the production of tutorial exercises or "open-ended or communicative activities of a collaborative sort." This is one of the key questions considered in this thesis.

Despite misgivings on the subject, Wyatt reluctantly concedes that ASs offer the only short-term solution to the courseware shortage dilemma:

> "Although there is no general agreement on which approach to take (to address the courseware shortage problem), there does seem to be a significant movement towards the adoption of authoring systems... It appears likely that sound and worthwhile courseware can be developed in this way at far greater speed and far less expense." (Wyatt, 1983-b)

Chapter Three: Research Design and Findings

This chapter examines those language learning activities which might be applicable to a computer-assisted language learning environment and which could be reproduced using an authoring system.

As an introduction to the data base for this study, a definition is suggested for the type of activities to be examined with an analysis of their nature. The method of gathering the data is explained and the data is presented.

The data can be separated into two categories, activities used to exemplify an analysis of language learning activities from an operational point of view and those used to exemplify an analysis from a pedagogical point of view. The data pertaining to operational typologies is examined first and the results of this examination are subsequently applied to the data pertaining to pedagogical typologies.

The aim of the first part of the analysis is to establish and codify the notion of "flexibility" introduced in Chapter Two by isolating those operational factors of the activities under discussion which might affect an authoring systems's ability to reproduce them on the computer. The three distinct, but related, problem areas cited by Kearsley in section 2.5.2 are examined: The presentation of discrete interactive elements of the activity to learners, immediate feedback to learners based on an analysis of the their response to the stimuli offered, and the global decision as to how to continue the activity with subsequent interactions.

In the second part of the analysis the relationship between operational flexibility and pedagogical effectiveness is examined. The discrete pedagogical factors used to categorize activities are isolated and the interaction between the key operational factors established and these pedagogical factors is determined.

3.1 CALL and its role in the language learning process

In section 2.1, a working definition of CALL was developed which excluded from the discussion uses of the computer as a tool or as a passive medium, and restricted it to considerations of the computer as an active participant and interlocutor in the language learning process. The discussion was further limited to uses of the computer for language practice, a point which will now be considered in further depth.

"Instruction" ultimately derives from a human resource person, normally the teacher. As teachers and other qualified resource persons tend to be in short supply, non-human surrogates which are derived from the teacher must often be utilized. Historically, the textbook has been the most widely used stand-in for the teacher. More recently, other more mechanical substitutes, such as language labs and computers, have been integrated into the students' learning experience.

No mechanical aid can ever complety replace the sensitivity to learners' varying needs which human teachers possess (Byrne, 1976, p.1). The most that one can expect of a mechanical surrogate is that it model, however imperfectly, some of the minor roles of the human teacher. A book presents information in a static manner, may pose questions, and can even provide a model correct answer. It cannot, however, modify the presentation of information, nor actively correct a learner's particular errors. The language lab offers the same capabilities and suffers from the same restraints as the printed page; it simply provides the sensory experience via the spoken rather than the written medium. The computer, on the other hand, is a much more sophisticated device. It can make judgements and thus has a greater potential to model the role of the teacher. Byrne's caveat still holds, nevertheless. The computer can serve as a surrogate teacher only in those areas where the full scope of the human teacher's sensitivity and subtlety are not called into play.

The language-learning process has been broken down into three segments: presentation, practice, and production (Dakin, 1973, p. 4; Byrne, 1976, p. 2).

In the presentation phase, new material is given to learners, either deductively through demonstration or inductively through example, which the learners assimilate through their own involvement in the discovery process (Dakin, 1973, p. 4). The learners' problem is that of understanding (Dakin, 1973, p. 6). The teacher's role is that of informant, carefully selecting material and choosing a method for presenting it in as clear and memorable way as possible, based on a thorough knowledge of the language and a feeling for the learners' individual needs (Byrne, 1976, p. 2). Computers serve in the presentation phase only in the same way as books, language lab tapes, films, and other media can serve - as passive presenters. To serve as even an imperfect surrogate for the teacher, the computer would require a knowledge of the language and a way to gauge the subtle individual reactions of students, which is currently beyond our ability to codify, let alone program. The computer does not seem, therefore, to offer any special advantages in the presentation stage of language learning.

In the production stage, control over students' performance is relaxed and students produce their own language (Dakin, 1973, p. 4). Now the learners' problem is that of communication (Dakin, 1973, p. 6) and the teacher's role is that of guide. Students are directed towards using the language freely, to express their own ideas in their own words for their own purposes (Byrne, 1976, p. 2). Of all the support media, the computer alone seems to offer the potential of actively participating at this stage. Ventures in this area, such as the program "ELIZA", have been only experimental, however. Current computer programs, developed from scratch by professional programmers, are ineffective in matching the linguistic abilities of even the least advanced learners, let alone acting as their guide. Little immediate application is therefore seen for authoring systems in the production phase of language learning.

The practice stage falls in between these two extremes. Learners are called upon to produce pieces of language in a carefully constructed and controlled environment (Dakin, 1973, p. 4), that can be duplicated on the computer. The learners' problem at this stage is that of remembering and applying what they have learned (Dakin, 1973, p. 6), but not the communication of original ideas, which the computer could not handle.

Byrne (1976, p. 2) likens the teacher's role at this stage to that of an orchestra conductor. In expanding Byrne's analogy, the conductor can be imagined as serving two roles. First, the conductor must orchestrate the music, constructing the controlled environment. This is the role of the teacher/author in designing a language practice activity, a role where the computer cannot serve as surrogate. The second role of the conductor is to guide the individual musicians in the reproduction of the orchestrated music, the corresponding role of the teacher being to lead students in using pieces of language within a controlled interaction. Students' responses at this stage are likely to be unexpected and confused, to the point of causing a breakdown in the interaction. The teacher must be sensitive to this, restructuring or temporarily interrupting the course of the interaction with feedback and guidance so as correct it (Dakin, 1973, p. 4).

Dakin (1973, p.7) considers that even the more rudimentary technology represented by the language lab could assume some of the roles of the teacher within this prescribed, controlled environment. The computer, with its far greater capabilities, though still feeble by human standards, could be expected to play a much greater part in assuming the teacher's role.

While interest recently has focused on the production phase of language learning, the necessity and importance of the practice stage is underlined by a variety of authors (Stevick, 1971, p. 391; Byrne, 1976, p. 32; Robinett 1978, p. 206). Learners need focused practice where they can build up confidence, because they are given something to say in an environment where confusing distractions are kept to a minimum (Byrne, 1973, p. 32).

3.2 Forms of language practice

In the general context of language learning, any verbal activity whatsoever could be considered practice. The definition of CALL outlined in Chapter Two limits the discussion, however, to a particular kind of language practice – that which is interactive, structured or "focused", and mediated.

Drills and exercises have traditionally been seen as a form of focused practice (Stack, 1971, p.161; Stevick, 1971, p.393; Rivers & Temperley, 1978, p.398; Robinett, 1978, p.206). A drill allows the learner to focus on aspects of language which are not normally the object of attention during normal language use or are seldom repeated several times in a row during the course of a normal conversation (Stevick, 1971, p. 393).

The use of the term "drill" evokes strong reactions among language teachers because of the origins and connotation of the term. Drill has traditionally been closely identified with a behaviourist philosophy of language learning. Drills have been narrowly defined as activities which focus on grammatical practice and which only allow for a single correct response by the learner (Stevick, 1971, p.393; Cook, 1972, p.121; Dakin, 1973, p.91; Rivers & Temperley, 1978, p.122; Robinett, 1978, p.47). The content of the drill needs to be unambiguously designed to lead the learner inexorably towards this single acceptable response (Cook, 1972, p.121; Rivers & Temperley, 1978, p.122). In order to escape the rigid confines of the above definition, some experts have adopted the term "exercise" to identify those activities where the focus may not be grammatical, which are less rigidly controlled and where more than one response might be acceptable (Stevick, 1971, p.393; Dakin, 1973, p.91). One could escape this confusion entirely by adopting a totally neutral term such as "activity". Activity is too general an expression, however, and does not adequately describe the particular form of practice traditionally associated with drills. Indeed, "exercise" itself suffers from the same generality of connotation.

The words "drill", "exercise", and "activity" here are used interchangeably, and divorced from their respective theoretical connotations, to refer to a particular format for language practice. Excluded from consideration are those formats identified as drills, exercises, or activities which do not fit the criteria set out below.

Drills are universally seen as a form of interactive practice. The interaction consists minimally of two parts. The first part has been called the "stimulus", "prompt", "cue", or "input" (to the learner), while the second part has been called the "response" or "output" (from the learner) (Stevick, 1971, pp. 393-394; Cook, 1972, p.121; Dakin, 1973, p.49; Rivers & Temperley, 1978, p.121).

Drills are seen as structured practice. The interaction proceeds algorithmically according to a predictable pattern (Dacanay, 1967, p.107; Rivers, 1978, p.398). It is the algorithmic, predictable nature of drills which allows their presentation by non-human interlocutors, such as books, language labs, or computers.

Drills are mediated practice; there is a set of expected responses and the learner's response can be examined in consequence of this set. To include the act of mediation, the minimal stimulus-response interaction must be expanded to include a third phase, the reaction to the learners' response, or "confirmation of response" (Stack, 1971, p.126; Rivers & Temperley, 1978, p.121) Reaction to the learner's response, or "feedback" in today's terminology, assumes an implicit phase which might be called "analysis of response".

Explicit reaction to each response may be dropped to foster a more natural interchange (Rivers & Temperley, 1978, p.121; Hardin, 1983). The teacher will still, nevertheless, be analyzing the learners' responses and the global course of the drill, the selection of material and the number of interactions, will be a adjusted as a consequence of the responses given. In artificial environments, the learner is often called upon to make a self-evaluation, based on exposure to acceptable responses contained in an answer-key at the back of a textbook or on a language lab tape. In correcting the learner's written work, or in monitoring the language lab interaction, the teacher can still make global alterations to the drill sequences.

The computer is capable of adopting the teacher's role as mediator. It can examine the learner's response, using tools that the teacher has given it, and can provide immediate feedback and make global decisions according to the options that have been programmed.

Activities of the drill format, then, being language practice which is intereactive, structured, and mediated, satisfies the working definition of CALL provided in Chapter Two.

3.3 The nature of drills

Drills can be considered as consisting of individual interactions, or drill "items". Each discrete item may be broken down into the following essential stages:

1. the "stimulus" or "prompt",
2. the "response" to the stimulus by the learner,
3. the "analysis" of that response in terms of a set of expected or acceptable responses,
4. the "feedback" to the response, based on the analysis,
5. "branching", or a global decision as to the subsequent course of the drill, based on the analysis and other factors.

This use of the computer term, "branching", must not be confused with the standard term adopted in computer programming, where it means executing any computer instruction other than the subsequent one. In computer programming terms, the presentation of immediate feedback within a drill item might involve branching while the decision as to the next item to be done might very well be accomplished without any branching at all. When viewed in terms of the present definition, the "branching" powers of a given authoring system may turn out to be considerably weaker than if described according to the standard programming sense of the word. considerably weaker when viewed with our definition.

The "prompt" can be further broken down into the "model", the piece of language upon which the learner will be working, and the "cue", the element or instruction which tells the learner how the model is to be used or modified (Dacanay, 1967, p.107; Rivers & Temperley, 1978, p.123). Stages three through five of the model described above would normally be the domain of the teacher. Ignoring these, then, for the moment, a traditional drill could be presented as illustrated below:

3.0 Example: Basic Pattern

#1 Model: I go every day.
Cue: yesterday
Response: I went yesterday.

#2 Model: I eat every day.
Cue: yesterday
Response: I ate yesterday.

(Dacanay, 1967, p.107)

(Original drills are uniquely numbered according to the order in which they were considered for inclusion in the data. There are gaps in the numeration as not all drills considered were included in the final data.)

Dacanay (1967, p.107) describes three formal variations on this basic pattern: interlocked drills, formalized drills, and telescoped drills. In interlocked drills, the response to one item becomes the model for the next:

3.1 Example: Interlocked Drill

#1 Model: He wrote the report for me
Cue: letter
Response/ He wrote the letter for me

#2 Model:
Cue: She

(Dacanay, 1967, p. 107)

In formalized drills, the explicit cue is replaced by a standard instruction for modifying a series of models:

3.2 Example: Formalized Drill

Instruction/Cue: Pluralize the noun.

#1 Model: I brought the letter.
Response: I brought the letters.
#2 Model: I brought the note.
Response: I brought the notes.

(Dacanay, 1967, p. 108)

In telescoped drills, there is no separate, explicit cue. Rather, the cue is contained within the model:

3.3 Example: Telescoped Drill

#1 Model/Cue: Would you like tea or coffee?
Response: I'd like tea.
#2 Model/Cue: Would you like cake or ice cream?
Response: I'd like ice cream.

(Dacanay, 1967, p. 108)

The purely textual representation of drills above is not designed to exclude from consideration the use of other media. A drill can be greatly enhanced by replacing or supplementing an explicit verbal cue with, for example, a picture. One illustration can quickly make clear a complex context which would take too long to explain verbally. Since a picture is a representation of reality, its use enhances the communicative nature of an activity. (Dacanay, 1967, p. 152)

Many simple exercises, indeed, depend on visual stimuli. In the verbal cueing of a simple vocabulary exercise, for example, it would be necessary either to paraphrase the item in the target language or to provide a translation in the learner's language. Many simple vocabulary items would be difficult to paraphrase with language appropriate to the level of the learner, and translation is only applicable to homogeneous groups of learners:

3.19.1	Model:	These are -----.
	Cue:	You see with them.
3.19.2	Model:	These are -----
	Cue:	des yeux

The use of a simple picture renders this very mundane exercise much more interesting and useful:

3.19.3	Model:	These are ----- .
	Cue:	(picture of eyes)

The computer can provide pictures as cues in a variety of ways. It can draw the pictures on the screen using its graphics capabilities. It can activate remote devices such as slide projectors, videotapes, or videodisks. It can prompt the user to refer to "off-line" media, such as a printed text. The presentation of audio or "live action" sequences is no different for the computer than the presentation of a single slide or videodisk frame.

Many authoring systems excel in providing mechanisms by which non-verbal cues can be included. Some provide powerful graphics editors, others provide easy to use interfaces to peripheral devices such as tape recorders or videodisks. A glance at 3.19 will show, however, that these have little effect on the structure of the exercises. The cue is simply the cue, regardless of whether it is merely a written one or, more dynamically, a recorded visual sequence. dynamic filmed one.

3.4 The drill sample

In an effort to collect a sample that would be a representative cross-section of drills and drill-type activities, examples were taken from drill typologies produced by a number of accepted specialists. This approach was adopted in order to establish a wider and more complete range of examples, and within a more reasonable time, than would result from a search of original sources such as actual classroom drills, textbooks, language lab tapes, and similar material.

Table 3.1 of Appendix A lists the eleven authorities consulted. The works span a period of fourteen years and represent the range of language teaching philosophies, from the behaviourist/structuralist approaches of the 1960's to the cognitive/communicative approaches of the 1980's.

In total, 127 examples were finally included in the data base, as listed in tables 3.2 and 3.3 of Appendix A. In examining a work, an effort was made to include at least one example of the drills and activities that were being proposed in support of the author's discussion. In most cases the drills were explicitly shown. In a few cases, all or part of the drill had to be extrapolated from the discussion. Where there were numerous examples categorized by the author as being of the same "type", usually only one was taken. The exceptions were when it was evident that two "like" drills were obviously different in features that might prove important.

A preliminary screening of examples was undertaken to exclude those that did not fit into the parameters of this study. These included activities intended primarily for the presentation or production phases of language learning as well as those which were not of an interactive, structured, and mediated nature.

Each author consulted presents the drill examples in a different fashion. In order to standardize the data, all examples were fitted into Dacanay's uniform frameworks, as shown in examples 3.0 - 3.4. So as to illustrate possible relationships between discrete items, two representative items were chosen from each drill example, wherever possible. In cases where parts of the drill had to be assumed, these are enclosed in square brackets [ ]. The original examples are listed in Appendix A.

The data consists mostly of oral drills designed originally for classroom or language laboratory use (See table 3.3). It may seem odd to use oral examples as the basis for a discussion of CALL, typically viewed as a written medium, and there are several points to consider in this regard:

1. A much larger and more varied selection of examples can be found in discussions of oral practice than in discussions of written practice. Most of the written drills examined were found either to be essentially derived from oral drills or to be activites better suited to the

production stage of language learning rather than the practice stage.

2. It is assumed that, while each medium will accentuate certain qualities of a drill, the essential nature of different types of drills will be independent of the medium used to present them. Cook (1972, p.121) and Rivers and Temperley (1978, p.278) support this approach in their discussions. Rivers and Temperley (1978, p.276) warn, however, that some adjustment in the presentation of the drill will be required as one transfers it from its original medium to another.

3. CALL can be considered a written medium in the sense that the learner interacts with the computer by reading written stimuli from the screen and responding via the keyboard. In other respects, however, it much more closely reproduces the environment of the oral drill: The responses made by the learner tend to be short, they are evaluated immediately and with a subtlety far exceeding any written answer-key, and some sort of pertinent feedback is provided to the learner. The "written" nature of CALL exercises can be moderated with a variety of non-standard output devices (tape recorders, slide projectors, videodisks, or voice synthesizers). and input devices (touch sensitive screens, light pens, "mouse" pencils, or

voice recognition devices), all of which would have little effect on the essential nature of the drill. Table 3.4 shows how CALL provides an environment which can be likened to that of the oral classroom, the language lab, and the written drill.

4. The one other major teacher-surrogate technology is the language lab. Drills designed for the language lab, like many written drills, can be seen to be derived from oral classroom drills and exhibit nearly all of the same drill types.

This study is, therefore, based on the premise that CALL drills will exhibit the same basic types of practice activities as one finds in the classroom or in the language lab.

3.5 Bases for drill typologies

The authorities consulted have organized their drill typologies according to two main gradients, which might be called categorization by "operational" distinctions and categorization by "pedagogical" distinctions. In general, operational distinctions refer to the operation that the learner must perform in executing the drill, while pedagogical distinctions refer to the drill's overall utility in the learning process. Table 3.5

shows that, on the whole, equal weight has been given to both schemes, although some authors exemplify only one or the other approach. The need to examine drills according to both dimensions was underlined by almost all those consulted (See Tables 3.6 and 3.14 for references).

It is a premise of this study that operational differences primarily reflect variations in the format (or algorithm) of a drill while pedagogical differences reflect primarily variations in content. An authoring system will provide for, and may limit the expression of, a range of drill formats, but should not have any influence on the choice of content. There may, however, be some aspects of drill format which will limit the range of pedagogical expression and some aspect of content which will enforce the choice of a more elaborate drill format.

Those drills used to exemplify operational typologies were analyzed with a view to establishing the essential factors used to differentiate them. Table 3.2 lists the 71 examples, from the works of 5 different authors, that were so examined. The effect of different operational factors on the possible computerization of the examples were then examined in detail.

Those drills used to exemplify pedagogical typologies were similarly examined to establish their essential pedagogical features. Chart 3.3 lists the 56 examples, from the works of 8 different authors, that were so examined. The possible effects of the operational factors on the expression of the pedagogical factors in a computerised version of the exercise were then explored in detail.

3.6 Traditional categorization by operational type

Operational type is the framework by which drills have traditionally been categorized. Table 3.6 lists the major operational types as discussed by the authorities consulted for this study. Paulston (1972, p.131) cites Johnson in postulating two underlying factors upon which these categorizations are based: 1) the amount and type of restructuring of the stimulus required to produce the response and 2) the amount of information the learner must bring to bear to produce the response.

Table 3.7 correlates the different terms and, judging by the names of the categories alone, there seems to be a general consensus as to the basic operational types:

A. Repetition/imitation drills
B. Substitution/replacement drills
C. Transformation/mutatation/conversion drills
D. (Other non-standard categories)
E. Question/response drills
F. Translation drills

Each author delimits the categories somewhat differently, however, and so what one author might consider to be a drill of type X, another might see as a drill of type Y.

Repetition/imitation drills are on the borderline between presentation and practice. The learners repeat the model given them by the teacher, without modification.

These instruments allow for practice in pronunciation and prosody (Robinett, 1978, p.47), but can only be used as presentation devices for higher-level features such as grammar and lexicon (Cook, 1972, p.128; Rivers & Temperley, 1978, p.126; Robinett, 1978, p.47). The computer is not currently an appropriate instrument for evaluating pronunciation. Repetition/imitation exercises must, therefore, be excluded from the present treatment.

Similar to repetition exercises are those in which there is no possibility of error. Candlin offers the following example, which was not included in the analyses:

<u>Example 3.83</u> <u>Example: Switchboard (B1)</u>

Instructions: [Choose an element from each list to form a complete utterance]

Model:
- A) Some/A lot of
- B) store detectives/store owners/ shopkeepers/store managers
- C) use
- D) TV security systems/Two-way TV/ TV-cameras/photo-scan systems
- E) to
- F) catch/observe/check up on
- G) thieves/"old hands"/"regulars"

(Candlin, 1981, p.82)

Any combination of the elements given produces a correct utterance. With no possibility of error, there can be no mediation. Hence, this must be considered a type of presentation drill and does not qualify under the present definition as a possible CALL activity.

Substitution/replacement drills assume a model with one or more fixed slots, the content of which can be replaced by any word that fulfils a similar function (Rivers, 1968, p.101-102; Dakin, 1973, pp.49-51). Different semantic elements are practiced within the same pattern (Stevick, 1971, p.394).

There is a variety of sub-categories under the heading "substitution". These depend essentially on factors such as whether the substitution slot is given or the substitition causes a correlated change to the element in another slot. Although the deciding factors are more or less standard, the resulting subcategories are not standard, as a glance at the authors' characterizations of the various drills listed in table 3.2 will show, nor are they applied in the same way. River's simple substitution (3.21), for example, would be considered by Dacanay to be a correlative substitution.

Whereas in substitution exercises a pattern is given, in transformation/mutation/conversion exercises the pattern itself is what the student is asked to reproduce (Dakin, 1973, pp.49-51). The same semantic elements are used in different patterns or surface realizations (Stevick, 1971, p.394). Transformation basically practices syntax (Cook, 1972, p.128; Rivers & Temperley, 1978, p.130). Any change in word-order can be considered sufficient to make a substitution exercise into a transformation exercise (Stack, 1971, p.144).

Rivers (1968, p.101-102) originally considered that the use of substitution vs. transformation drills reflected differences in basic approaches to the analysis of syntax, with transformation exercises stemming from transformation grammar. However, in a later report (Rivers & Temperly, 1978, p.130) she explicitly uses the term "conversion" so as to differentiate the drill format from any relationship to underlying theories. Robinett (1978, p.50), too, warns that "although (transformation drills) sometimes resemble the manipulative operations in certain rules of transformational grammar, they are not meant to be equated with such rules."

There is a divergence of viewpoints on the identification of subcategories withing transformation exercises, similar to the disagreement on subtypes of substitution practice. What some authors consider to be subcategories are treated by others as separate categories, hence the D. entry in table 3.7. All of the distinctions involve four possible basic factors, often in combination: a change in word order, the insertion of new elements, the deletion of existing elements, or the substitution of one element for another.

With question/response drills, the operational definition becomes obscured by pedagogical considerations. Dacanay (1967, pp.133-134) sees response drills as having two possible functions, the exchange of information (pedagogical) or the practice of the grammar of the response (operational). Rivers and Temperley (1978, p.143) feel that response drills must concern themselves with information exchange and not grammatical forms so that they are not reduced to conversion drills. In example 3.37, however, they cite, as the purpose of the drill, the elicitation of the desired tense (Rivers & Temperley 1978, p. 146), clearly a grammatical consideration. Robinett's response drills concern themselves almost entirely with grammatical practice (Robinett, 1978, pp.52-53).

There exists no clear and accepted operational definition of response drills. Stack (1971, p.134) mentions that these require the introduction of material not present in the stimulus. There is also the "you-I" conversion. The question model "Are you going to the movies?" could be transformed into the statement "You are going to the movies", but would have to be responded to as "I am going to the movies".

Translation will involve, of course, lexical substitution and, very probably, some changes in word order. From an operational point of view, translation drills could be viewed as a complex combination of substitution and transformation. Translation, as an operation, differs from the others operations in content alone; a factor not being considered at this point.

For the purposes of the present study then, the list in table 3.7 can be reduced to three major, traditional operational distinctions: substitution, transformation, and response. It will be seen that these categories do not immediately correlate with the realities that must be considered in reproducing the various drill examples on the computer, where one must consider "operation" from a formal or "algorithmic" viewpoint rather than from a linguistic one. The lack of general agreement as to the scope and application of each term, and the myriad unrelated subcategories, make it difficult to use the traditional categories as a point of departure for the current analysis.

3.7 Drill categorization based on cue and response-model

Those drill examples used to illustrate operational typologies were analyzed from the point of view of the actual mechanical operation taking place, an approach which differs from the traditional analysis of a drill in terms of the linguistic relationship between the stimulus and the response. Cook (1972, p.122) offers a basis for this alternative approach to analyzing a drill, where a basic framework for the response is postulated and the changes that the response undergoes between one drill item and the next is examined. The resulting desription parallels much more closely the algorithmic description required to prepare a drill for for possible presentation on the computer, than does the traditional analysis.

In the analysis shown in table 3.9 of appendix B and summarized in table 3.8, drills are examined in terms of the cue (C:), that part that actually triggers a change, and the model (M:), where "model" is now re-defined to mean the "response-model" or basic framework for the response. In those cases where drills could be analyzed in more than one way, the way that seems most productive in terms of the question of computerization of the drill, has been adopted.

Chapter 3: Research Design and Findings

A major division between "simple" drills (those drills which can be accomplished with one discrete operation) and "complex" drills (those which require more than one simple operation) is evident. Rivers & Temperley (1978, p.125) calls the latter "mixed" drills, a term which is here employed in a more restrictive sense.

The fundamental operations identified are:

I) "insertion" of an element into the model,
II) "removal" of redundant elements from the model,
III) "re-order" of model elements, and
IV) "production" of new elements, based on the cue alone (where the response-model or models are implicit).

I. Several different types of insertion are evident:

Type A: where the primary operation is the insertion of the cue into the model. This can be shown schematically as:

C: --> M:[]

where:
C: stands for the cue
M: stands for the response-model
[] stands for an insertion point in the model.
--> indicates that the operation is the insertion of C: into M:[]

Example 3.4(Dacanay) can, thus, be described as follows:

C: away
M: I'm going [].

Type B: Where the cue could be presented as part of the model, triggering the insertion of a secondary element into the model:

E:-->M:C:[]

where:
E: stands for a secondary element
C: stands for the cue imbedded within the model

Example 3.5(Dacanay) can be described as follows:

C:+M: Where [] José live?
<---

where:
C:+M: stands for the model with the imbedded cue.
José therefore is the imbedded cue
<-- indicates that José determines the choice of insertion element for the slot [] in the model.

Type C: Where the cue itself is not inserted into the model, rather, it triggers the choice of another element which is inserted in its place:

C:-->E:-->M:[]

where:
C:-->E: indicates that the cue determines the choice of secondary element
E:-->M:[] stands for the insertion of that element into the model

Example 3.20(Rivers) can be illustrated as:

C: uncle --> E: he
M: Yes, [] 's across the road.

where:
The cue, "uncle", determines that the insertion element is "he". "he" is then inserted into the response model.

Type D: where the cue is deleted from the model, triggering the insertion of a secondary element:

M:+C:-->M:-- <-- E:

where:

M:+C:	stands for the model, containing the cue
-->	stands for the implicit deletion operation
M:--	stands for the model with the cue deleted
<--	stands for the insertion operation
E:	stands for the secondary element to be inserted

Example 3.33(Rivers) can be represented in the form:

C: n't
M: They have -- [] coffee.

where:

n't	is the cue
They have -- [] coffee	is the response-model with the cue already deleted.

Type E: where there is no cue. The choice of insertion element is determined by the learner.

II. The <u>removal</u> of redundant elements was not found in any of the simple drills, but was a component in several of the mixed drills.

M:+Rd: --> M:--

where:

Rd:	stands for the redundant element.
M:+Rd:	stands for the model, containing the redundant element
-->	stands for the operation of removing Rd:
M:--	stands for the resultant model

This operation can be demonstrated in example 3.13(Dacanay):

M:+Rd:	The noisy children <u>who were making a lot of noise</u> were sent to bed.
M:--	The noisy children -- were sent to bed.

where:
The underlined portion of the model is the Rd:

III. With the <u>re-order</u> of model elements, the cue is formalized and there is no real distinction between the prompt and the response model:

M:[1][2] --> M:[2][1]

where:

M:[1][2]	represents the model with elements in the original order.
-->	stands for the operation of re-ordering the elements
M:[2][1]	represents the resultant model

Example 3.7(Dacanay) illustrates the typical case:

```
M: [The girl] [is] [ready]. --> [2][1][3]
      [1]      [2]   [3]
```

IV. In exercises involving the <u>production</u> of new elements based on the cue, it is not possible to predict a single response model, as there are many possible variations This operation could be symbolised in the formula:

C:-->R:-->(M:)

where:
C: stands for the cue
--> represents the operation of responding to the cue
R: stands for the learner's response
(M:) stands for the implicit response model.

"Complex" drills can be categorized as:

I) "multiple" operations, those in which the same simple operation is repeated;
II) "mixed" operations, those in which different simple operations must be brought to bear; and
III) "choice" of operation, where succeeding drill items require different types of operation.

All of the drills which cannot be analyzed in terms of one simple operation respond to an analysis based on more than one simple operation.

I. Three variations on "multiple operations" are evident. With "insertion + insertion", the cue causes insertion of secondary elements into more than one slot. With "insertion, then insertion", the insertion of the primary element creates an environment which triggers the secondary insertion. Similarly with "re-order, then re-order". (The reader is referred to table 3.9 for examples of each of the above situations.)

II. "Mixed operations" call for the application of different kinds of simple operations. With "insertion, then removal", the insertion of an element into the model forces the removal of other elements which have become redundant. The data contains one example of the need for "insertion, then removal, then insertion", where the removal of the redundant elements opens up the opportunity for more insertions into the model. In two examples, the learner has to formulate a question and then answer it. In terms of the present analysis, this could be seen as a "re-order, then production of new elements". In another example, the learner has to formulate the answer to a question, and then insert the cue, or "production of new elements, then insertion".

III. In drills calling for a "choice of operation", there is only a single operation to be accomplished each time, but the learner's task involves a judgemental decision as to the operation to be selected and accomplished. Examples are found of the choice between "insertion or re-order" and of "insertion Type A" or "insertion Type C"

It remains probable that there are many other complex combinations of simple operations that are not evident in the current sampling of drills and exercises.

3.7.2 Fixed and variable insertion slots

Insertion operations can have as their goal the selection of the proper insertion element, the selection of the proper insertion slot, or both. The purpose of the exercise involving an insertion will directly affect its presentation in a computerised form as it will condition the manner in which the response-model can be displayed and will determine whether the exercise is a "simple" or "complex" one. computer.

Where the goal of the exercise is the selection of the proper insertion element alone, the insertion slot will be fixed and presented to the learner. Drills of this type can easily be presented with the use of fill-in or simple multiple-choice response elicitation techniques. (A complete discussion of the various response elicitation techniques can be found in section 3.10.6.)

3.5 (Dacanay) Selection of insertion element:
Fill-in

Where ---- José live?
?

3.5 (Dacanay) Selection of insertion element:
Simple Multiple Choice

```
Where ---- José live?
   a. do
   b. does
```

Where the goal of the exercise is exclusively the selection of the proper insertion slot, the form of the insertion element will likely be provided to the learner. The resultant drill becomes, de facto, a multiple-choice activity. It can either be explicitly multiple-choice, with the learner typing a letter or number to indicate the slot choice, or it can be a fill-in activity where the learner chooses, using cursor control, the slot in which to type the response and the computer records only the slot choice.

3.11 (Dacanay) Selection of insertion slot:
Simple Multiple Choice

```
He [1] 's [2] on time [3].
Choose (1-3)
```

3.11 (Dacanay) Selection of insertion slot:
Fill-in = Multiple Choice

```
He --- 's --- on time ---.
Type the word in the correct blank.
```

In cases where both the insertion element and the correct insertion slot are to be selected, the learner is dealing with a complex drill. The learner may correctly choose the element, but place it into the wrong slot, or vice-versa, or both. If n represents the degree of feedback possible on each point, then there would be n x 2 feedback possibilities.

On authoring systems that only offer simple multiple choice, it would be necessary to complete such drills in two steps, one for each discrete point. With the "fill-in = multiple choice" technique shown in the above example (3.11b), both responses could be elicited in a single step, though they would still have to be analyzed separately.

The insertion operations isolated in table 3.9 were subsequently re-examined to determine whether the insertion was to be made in a fixed location in the model or whether the learner had a choice as to the point of insertion of the required element. Table 3.10 lists the results of this examination. In complex drills, only the insertion portion of the activity was considered. Where there were multiple insertions, the likelihood of the learner making a mistake in the selection of the insertion slot was considered and, if such an error seemed unlikely, the insertion slot was considered as fixed.

It is only with insertions of Type A, including those complex drills where the discrete operation is an insertion of Type A, that one finds a choice of insertion slot. In the others, notably in insertions of Type B & C, the insertion slot is fixed.

A few of the drills illustrate complications. In River's example 3.28 & 3.29, if allowance is made for a selection of slot, the slot would determine the form of the insertion element. In River's 3.32, the selection of slot is left up to the learner and its validity cannot be judged mechanically.

What of those insertions of Type A where the insertion slot is fixed? Examples 3.4 (Dacanay) and 3.40 (Robinett) represent such simple operations that they would barely qualify as CALL exercises according to the working definition established previously, as there is a restricted possibility of error. In the other four examples (3.61 - 3.144) either the isolation of the cue or changes to the cue based on its insertion into the model are the primary goals of the exercise.

In summary, it is in insertions of Type A (insertion of cue) that a choice of insertion slot is to be expected, although this pattern is not invariable. In other drills the insertion slot is likely to be fixed.

Where there is a fixed insertion slot, fill-in or simple multiple choice elicitation techniques probably represent the best response elicitation technique to use for the computerised form. Where there is choice of insertion slot, multiple choice is the preferable. Where there are both, neither fill-in nor simple multiple choice is adequate. "Fill-in=multiple choice", "holistic multiple choice" or "free response" (to be fully discussed in section 3.9) must be used if the author of the computerised drill desires that the learner complete the exercise in single step.

3.7.3 Variations in cue presentation

The manner in which the cue is to be presented to the learner is also of importance to the computerization of an exercise.

The cue can be explicitly presented, or it can be implicitly understood by the learner, as is the case in formalized drills. A drill may be partially formalized. In other words, there may well be an explicit cue, but further formal instructions may be required to identify, to determine how to apply it.

Where the cue is explicitly presented, it can be imbedded in the response model, or it can be presented as a separate element. Where the cue is a separate element, it can be presented alone or be imbedded within a cue framework. In the latter case, the learner must be able to isolate the cue from the context of the cue framework. The cue is the discrete part of the cue framework that the learner must consider in order to formulate a response. The cue framework is that part of the cue that plays no direct part in determining the response.

The response-model used in the current classification usurps most of the function of Dacanay's original idea of "model". Sometimes, however, it is not possible to provide all the required information in the cue or response model. For those cases where the learner will require supplementary data that is present in neither the cue nor the response model, it is possible to define an element called the "prompt". Prompts can serve either an operational purpose, providing the learner with information required to complete the task, or a pedagogical purpose, setting up a situation or a reason for the task. At this juncture, consideration is given only to the operational need for a prompt.

An examination was made of the variations in the presentation of the cue, and of the prompt where its inclusion is essential, in the drills used to illustrate operational typologies. Seven discrete factors were studied:

1) whether no explicit cue was to be presented because the cue was to be implied by the instructions (in other words, a formalized drill),

2) whether there was no explicit cue because of some other reason,

3) whether the explicit cue was imbedded within the response model,

4) whether the cue was a separate element, presented in isolation,

5) whether the cue was a separate element presented within a cue framework,

6) in cases of explicit cues, whether more instructions were necessary to isolate the cue from the cue framework or to determine how the cue was to be applied,

7) whether a separate prompt was operationally required.

Chart 3.11 lists the results of this evaluation, along with a detailed analysis of the cue and prompt presentation in each case.

Some correlation between operational drill type, as established in 3.7.1 and presentation of the cue is evident. Much of this may stem from the definition of the types, however. With simple insertions of Type A (C:-->m:[]), for example, where the goal is the insertion of the cue into the model, it is no surpise that the cue and model are presented separately.

It is interesting to note, nevertheless, that in most of the cases the separate cue is presented in isolation rather than as part of a more elaborate cue framework, which might be more interesting pedagogically. The use of formal instructions to isolate the cue would make an activity more challenging. One might expect to see the latter two situations see a larger use in the drills categorized pedagogically.

With insertions of Type B (E:-->M:C:[]) the most expedient manner of presentation is to imbed the cue within the response model. It is interesting to note that in the majority of cases, a separate prompt is also required, so that the identity of the secondary insertion element can be determined. In 3.5(Dacanay) for example, given only:

C:+M: Where [] Jose live?

the learner might provide responses such as "can", "may", etc. The prompt:

Where [does] Mary live?

closes the response set (the set of probable responses, a term to be discussed more fully in section 3.10.1) by limiting responses to forms of the element indicated. The prompt could be replaced by a formal instruction:

Use a form of the auxiliary "do"

or alternatively, possible responses could be explicitly displayed using a multiple-choice elicitation technique.

To eliminate the necessity for inclusion of a "prompt", the identity of the secondary element must be fixed by the grammatical and lexical environment provided by the cue and the response model. In 3.35(Rivers) the response set is fixed by the form of the response model and the lexical link between cue and response, and therefore no prompt is necessary. In 3.140(Stack) the response set is fixed by the grammatical situation.

With re-orders of the model elements, all of the examples represent formalized drills with no explicit cue, although it is possible to imagine that an author could devise an explicit cue to serve the same purpose.

With the production of new elements, the tendency is towards explicit, separate, isolated cues, accompanied by a formalized instruction as to how to apply the cue. There is no cue framework, as the nature of the cue is such that the entire element, and not just a part of it, must be considered by the learner in producing a response.

3.8 The realisation of exercises on the computer using an authoring system

In Chapter Two it was shown that the mechanisms or templates offered by an authoring system may be fixed, allowing the author the flexibility to provide merely the content of an exercise, or there may be a range of templates from which to choose or even the possibility of creating new templates using "macro-instructions".

The mechanism that an authoring system provides cannot be considered in isolation from the data required by the authoring system to generate an exercise. Once the global template for an exercise has been established, the authoring system must to elicit from the author the data needed for each interaction. The types of information that the authoring system is able to store influences the manner in which it must handle the presentation of an activity. An authoring system is not capable of operations for which it cannot store data. In proposing, below, various possible templates, some consideration must be given to their data-storage requirements.

In section 3.3 a drill item was analyzed into five component stages. To allow an author to realise a drill on the computer, an authoring system must provide a "global" template that permits the completion of each of these five steps. The enormous range of variations that must be considered when examining the complete drill defies systematization, however. It is necessary, therefore, to analyze the different stages of the drill interaction independently. Kearsley's division of an exercise into three functional stages (outlined in section 2.2.5.3), 1) the presentation of the drill and the elicitation of the learner's response, 2) the analysis of the response and the presentation of pertinent feedback, and 3) the decision as to the next item to be done, can serve as a framework for the analysis. The "global" template can then be described in terms of its component presentation templates, answer-analysis templates, and branching templates.

It must be kept in mind that the components are, of course, inter-related. The choice of a fill-in vs. multiple choice presentation, for example, will determine the nature of the answer-analysis mechanism and the data required for answer-analysis. Branching possibilities will be very much affected by the choice of presentation and answer-analysis templates.

3.9 Factors influencing the presentation phase of a drill

The first phase of a drill to be considered is the presentation of the stimulus to the learner, and the subsequent elicitation of the learner's response. While a collection of typical example templates is developed to describe the drills listed in the data, it is again impossible to systematize global presentation templates because of the large number of possible variations. Instead, the discussion is restricted to various discrete aspects of the stimulus presentation and response elicitation. Four basic approaches to eliciting a response are considered: fill-in-the-blank, multiple choice, holistic multiple choice, and free response, with a review of the response-elicitation mechanisms that can be adopted for each case. The presentation of the stimulus is examined in terms of the instructions, the prompt, the cue, and the response-model.

The response-elicitation mechanism, or REM, is the bridge between the presentation and the answer-analysis phases. A particular REM will delimit the range of answer-analysis possibilities. Conversely, the requirements of the answer-analysis phase may call for a particular REM, which, in turn, may influence the form of the presentation template.

With "fill-in-the-blank" exercises, the computer displays the response-model, with the appropriate blank or blanks and the learner is invited to enter the elements that would normally occupy the indicated slot or slots within the model.

Example of simple fill-in (3.4 Dacanay):

Response Model: M: I'm going -------.
?

The response-elicitation may be represented schematically as:

```
?-->rt         where:
               ?-->     represents the act of
                        soliciting a response.
               rt       stands for a short one- or
                        two-word textual response
```

Sometimes there may be two blanks to complete as in the following example (3.12 Dacanay):

```
C:+M:              He --- n't --- coffee.
?-->rt1,rt2           ?        ?

               where:
               rt1      stands for the completion of
                        the first blank
               rt2      stands for the completion of
                        the second blank
```

Where there is a choice of slots for inserting the response, this choice can either be made by a multiple choice or by allowing the learner to position the cursor within the appropriate blank and then giving the computer the ability to understand where it has been placed.

Example of multiple-choice 3.6(Dacanay):

```
I:              Choose a or b
C:              tonight
M:              The --- leaves --- at seven.
                     a.          b.
?->rm           ?

          where:
          rm          stands for a multiple choice
                      response
```

Example of "fill-in = multiple choice":

```
P:              The train leaves tomorrow at seven.
C:              tonight
M:              The --- leaves ---- at seven.
?->rl=m              ?          ?

          where:
          rl=m        stands for the situation
                      where the cursor location can
                      be translated into a multiple
                      choice response
```

For the cases involving both the choice of insertion slot and the form of the element to be inserted, it is appropriate to postulate a REM of the following type:

```
?->rl=m,rt      where two response would, effectively,
                be generated by the learner's single
                action.
```

Three different multiple choice presentations can be postulated - simple multiple choice, holistic multiple choice and re-order multiple choice. The first would parallel very much the fill-in technique, in that the computer would display the response model with a blank. In this case, however, the response set would be explicitly shown:

Example of Simple Multiple Choice (3.5 Dacanay):

```
C:+M:          Where --- José live ?
                 a.  do
                 b.  does
```

With holistic multiple choice, the response set is included with the response model, to produce several complete responses. This is most appropriate with many complex drills, since the multiple or mixed operations can be reduced to a single operation, and for situations where the response set is excessively open-ended.

Example of holistic multiple choice (Dacanay 3.12)

```
P:             He likes coffee.
C:+M:          a.   He doesn't like coffee.
               b.   He don't like coffee.
               c.   He doesn't likes coffee.
               d.   He don't likes coffee.
?->rm
```

Re-order multiple choice differs from the other REMs in that a series of discrete responses is entered, representing, in fact, the entire explicit response set. What is significant is not the elements provided, but their order. The REM required for re-order exercises as:

?->ro.

Chapter 3: Research Design and Findings

Multiple-choice is the most powerful of the REMs provided by authoring systems in the sense that the simplest authoring system can still present complex exercises by presenting them as multiple-choice activities. Pedagogically, multiple-choice may not be the most appropriate template for an activity as the learner need only recognize and not produce the correct response. Its use is limited to those situations where the set of correct answers can be predicted.

Free-response is the easiest REM to structure, from the point of view of presentation and response-elicitation, but can be the most difficult for the computer to handle in terms of response-analysis. There is no explicit response model, rather, the response model (or models) are used internally during the evaluation phase. For free response to be possible, there must be some systematization, however complex, of the responses.

Example of free response (3.37 Rivers):

C: Why didn't they come home before midnight?

?->rs

Where rs stands for a sentence response.

The first part of the stimulus is the instructions and other supplementary data (I:). For the most part, the instructions are to help the learner to understand the nature of the task and how to perform it, although they could contribute to the motivation of the learner by establishing a pedagogically rich environment. Under the category of instructions, it is appropriate to include supplementary text, drawings, and so forth, which offer pedagogical support, but which do not figure operationally in the activity (a paragraph establishing the situational frame for a drill, for example). Instructions and other support material that does not figure operationally in the activity can be represented schematically with the symbol: "TTT", standing for neutral text. In formalized or partially formalized drills, the cue is implicit in the instructions and so the instructions become operationally important. Instructions that figure operationally in the activity are symbolized below by the symbol: "CCF, standing for formalized cue. Example 3.34(Rivers), for example, could be treated as a formalized drill, in which case the general instructions might be:

TTT Vary the final segment from future to conditional, according to the first segment.

To ensure the desired response, further, more explicit instructions would be required:

CCF Use " 'd " or " 'll " in your response.

Chapter 3: Research Design and Findings

Material presented to the student can be universal (i.e. the same across all items in a sequence) or item-specific. Lower case is used below to represent item-specific components and UPPER CASE to represent universal components. Thus, the patterns TTT or ttt or both may be encountered. A drill sequence could have universal instructions, TTT, and item-specific situational settings, ttt.

An authoring system need only elicit universal material once for an entire sequence but must elicit item-specific material separately for each item. Universal material can always, of course, be treated as item-specific material. The learner would see no difference in the presentation, but the author would be asked to enter the same material consecutively for each item. Universal frameworks, however, can be used in the generation of item-specific material and, as is demonstrated below, there are cases where the use of item-specific data instead of a universal framework reduces certain options, notably in the case of branching possibilities with interlocked drills. There are even cases in which item-specific data cannot be predicted and entered in advance, as prompt, cue, or response models will depend in part on information that the learner has provided.

The specific prompt for each item can thus be stored in advance, as ppp, or a prompt generating algorithm or rule can combine a universal prompt framework, PPP[], with some other element to derive ppg, the item-specific prompt. In some cases, a drill may require the same universal prompt for all items, PPP. Similarly, the cue can be stored as ccc, CCC[] + ? --> ccg, or CCC and the response model as mmm, MMM[] + ? --> mmg, or MMM. Insertion response-models are represented as mmm[], mmg[], or MMM[], and non-insertion models without the square brackets, []. In the case of multiple choice templates, the explicit response models are represented as by rr1...rrn, or rrn for short, with RRN for universal response models.

3.9.2 Presentation options in the data: Simple operations

This section relates the analyses contained in tables 3.9, 3.10, and 3.11, along with other variations not yet discussed, to the possible computerization of the drills in the sample.

3.9.2.1 Simple insertions of Type A

Simple insertions of Type A require, at least, ccc & mmm[]. We will assume TTT to be present in all cases. Example 3.4 (Dacanay) could then be presented using the following template:

P-template #1

I:	TTT	Complete the sentence with the word provided
C:	ccc	away
M:	mmm[]	I'm going ----.
?:	?->r	?

where the author would have to provide TTT * 1 plus ccc,mmm[] times (*) the number of items in the sequence or n-times.

Since the response models are identical across items, some authoring systems may allow the entry of a universal response model:

P-template #1b

I:	TTT
C:	ccc
M:	MMM[]
?:	?->r

where the author would have to provide TTT,MMM[] * 1 plus ccc * n.

There is no operational difference between the two. The only benefit foreseen would be to economize the author's time and to save storage space within the computer. While these factors of user-friendliness and programming efficiency are important considerations in judging authoring systems, they are not under discussion here.

Therefore mention of the possibility of using PPP, CCC, or MMM is only made where such use can have operational implications.

Example 3.40 (Robinett), 3.118(Stack), and 3.144(Stack) could also be presented using the above formula. They have in common that they are Type A insertions with the insertion slot fixed, where the cue and the response model contain all the information required to complete the operation.

Examples 3.118 and 3.144 differ from 3.4 & 3.40 in that the learner's task is more complex. In 3.4 & 3.40 the cue is inserted without change, while in 3.118 and 3.144 the cue must be changed. At first glance this may seem to be a function of the drill content alone. It does, however, have one operational implication. Whereas 3.118 could be presented as a simple multiple choice, for 3.4 and 3.40 such explicit listing of the responses would make the activity very trivial:

P-template#2 3.118(Stack)

I:	TTT	Choose the correct completion
C:	ccc	Il se repose
M:	mmm	Si Paul est fatigué ----.
	rrn	a. il se reposera
		b. xxx
		c. xxx

* 3.4(Dacanay)

```
I:    TTT          Choose the form which is the insertion
                   of the cue into the model.
C:    ccc          away
M:    mmm          I'm going ----.
      rrn                    a. away
                             b. xxx
                             c. xxx
```

Example 3.9(Dacanay) would require a multiple choice template, either fill-in=multiple choice, assuming the authoring system provided that capability, or simple multiple choice, if not.

P-template #3

```
I:    TTT          Insert the cue at the correct point in
                   the sentence
C:    ccc          for me
M:    mmm[]        Mother made [] a dress [].
?:    ?->rl=m                  ?          ?
```

P-template #4

```
I:    TTT          Type 1 or 2 to indicate the correct
                   place in the sentence to insert the cue
C:    ccc          for me
M:    mmm          Mother made [1] a dress [2].
?:    ?->rm        ?
```

Like 3.9, 3.11 (Dacanay) could also use the above approaches. Besides being Type A insertions, they have in common that they both have a choice of insertion slot, there is no change to the form of C:, and C: and M: contain all the information required to complete the operation.

Example 3.6(Dacanay) differs from the above in that the cue replaces one of the words that present in the original sentence. If the exerccise is done as fill-in=multiple choice, therefore, a prompt will be required so that the learner can see the original sentence:

P-template #5

```
I:   TTT        Insert the cue at the correct point in
                the sentence
P:   ppp        The train leaves tomorrow at seven.
C:   ccc        tonight
M:   mmm[]      The [] leaves [] at seven.
?:   ?->r1=m          ?          ?
```

Where done as a simple multiple choice, C: and M: will contain all the required information:

P-template #4

```
I:   TTT        Type 1 or 2 to indicate the correct
                place in the sentence to insert the cue
C:   ccc        tonight
M:   mmm        The [train] leaves [tomorrow] at seven.
                      [1]             [2]
?:   ?->rm      ?
```

Example 3.6 differs in another important aspect from 3.9, in that it is an an interlocked drill. The result of the insertion made in item(x-1) is reflected in either P:(x) or or M:(x), depending on the approach used. If ppp is used, then the order of the items will be fixed in the data and no computer-controlled branching will be possible. To provide a flexible branching option with an interlocked drill, the authoring system must offer a mechanism whereby an element of drill item(x-1), in this case C:(x-1), can be inserted into a universal prompt framework, PPP[], replacing an earlier element. The P:(x) can thus be generated, based on whatever the previous item was. This algorithm may be represented in the form:

PPP[] + ccc(x-1) --> ppg(x)

which gives rise to the following alternate template:

P-template #6

P-Rule:		PPP[] + ccc(x-1) --> ppg(x)
I:	TTT	Insert the cue at the correct point in the sentence
P:	ppg	The train leaves tomorrow at seven.
C:	ccc	tonight
M:	MMM[]	The [] leaves [] at seven.
?:	?->r1=m	? ?

whose surface realization is identical to the presentation using P-template#5 above. If done on the model of P-template#4 above, the discussion above about the interlocked nature of the drill would hold. For computer controlled branching, one would need a mechanism whereby mmm would be replaced by mmg based on the P-Rule:

MMM[] + ccc(x-1) --> mmg(x)

Like 3.6 are 3.10(Dacanay), 3.21(Rivers), 3.42(Robinett), and 3.60(Cook). These Type A insertions have in common that they are interlocked drills, in which the cue replaces an element already present in the model, and there is a choice of insertion slot.

Examples 3.120(Stack) and 3.125(Stack) are formalized drills with no explicit cue. Only M:, and possibly P:, need be displayed:

P-template #7

```
I:    CCF       Make the statement negative
P:    ppp       They're reading.
M:    mmm[]     They [] reading.
?:    ?->rt     ?
```

Example 3.61(Cook) is a partially formalized drill where both I: and C: are functionally required:

P-template #8

```
I:    CCF       Insert the second item mentioned in the
                cue into the model
C:    ccc       I can't decide whether I like swimming
                or skating best.
M:    mmm[]     Oh, I prefer ----.
?:    ?->rt     ?
```

If one could take advantage of the universal elements that are present in the original, the computer could offer a great deal more flexibility with this drill than is shown in the above template. A cue framework CCC[] could be defined into which the item discrete cues cc1 and cc2 could be placed. An algorithm could place the cues into first and second slots randomly so that on each presentation of the item, the order (and the answer) might be different. Alternatively, or in addition, the formal cue CCF could become a framework CCF[] into which universal cues CC1 ("first") and CC2 ("second") could be placed at random, changing the instruction:

P-template#10

```
P-Rules        1.  CCF[] + CC1,CC2 --->   ccfg
               2.  CCC[] + cc1,cc2  -->   ccg
I:   ccfg      Insert the first item mentioned in the
               cue into the model
C:   ccg       I can't decide whether I like skating or
               swimming best.
M:   MMM[]     Oh, I prefer ----.
::   ?->rt          ?
```

This subtlety of presentation assumes an answer-analysis mechanism capable, in the same way, of generating the current correct response.

So far the discussion has focused on presentations in fill-in or simple multiple choice modes. Most of the drills discussed could also be presented as holistic multiple- choice or free-response activities. Wherever simple multiple-choice could be used, holistic multiple choice is also possible, with the difference that the separate ccc or mmm[] can be suppressed if desired.

P-template #11   3.9(Dacanay)

```
I:   TTT      Choose the correct sentence
M:   rrn      a.  Mother made a dress for me.
              b.  xxx
              c.  xxx
?:   ?->rm    ?
```

As holistic multiple-choice is almost always an available option, especially for those authors using a weak authoring system, It is discussed only in those extraordinary cases where it might not be possible or where it is the only option. Likewise, from simply the presentational point of view, free response will always remain possible. The more complex answer-analysis requirements of free response will make it an uneconomical option in many cases. In other situations, it will be the only possible presentation mechanism. Mention of the free-response option is henceforth only made in the latter context.

3.9.2.2 Simple insertions of Type B

With simple insertions of Type B, the cue and the response model constitute one unit. In its simplest form, this could be represented in the form mmm[], as C: and M: are abstractions that do not always require separate, concrete, realisation on the computer. Example 3.35(Rivers) exemplifies the simple case:

P-template #12

I:	TTT	Complete the statements with the appropriate occupational term
C:+M:	mmm[]	A person who builds houses is a -----.
?->rt		?

Example 3.140(Stack) is similar. In both cases, the relatively closed lexical or grammatical nature of the task is such that the range of response possibilities is limited.

In all the other Type B insertions in the data, some additional information is required in order for the learner to produce the intended response. This can take the form of either a prompt or of formalized instructions. Example 3.34(Rivers) provides a typical case. With a prompt, it can be presented in the form:

P-template #13

I:	TTT	Vary the final segment from future to conditional, according to the first segment
P:	ppg	If I see him, I'll tell him.
M:	mmg[]	If I saw him, I -- tell him.
?:	?->rt	?

3.34 is also an interlocked drill, which is why ppg and mmg are adopted. The algorithm to produce ppg is more complex than in insertions of Type A, as both the mmg(x-1) and the correct answer to the previous item, aaa(x-1), are needed to produce ppg(x):

PPP[] + mmg(x-1) + aaa(x-1) --> ppg(x)

As discussed earlier, the drill could, of course, be done with ppp and mmm[], but the order would become fixed by the data.

3.34 can also be presented as a formalized drill:

P-template #14

I:	TTT	Vary the final segment from future to conditional according to the first segment
	CCF	Use 'd or 'll in your response.
M:	mmm[]	If I saw him, I -- tell him.
?:	?->rt	?

As a formalized drill, the interlocked quality of 3.34 can be maintained without the computer generation of elements. In the situation where an authoring system could not generate ppg and mmg and computer-controlled branching was nevertheless required, the drill could always be re-cast in a formalized format.

In the multiple-choice mode, rr1...rrn would achieve approximately comparable results to those of CCF in the above example. Multiple-choice might, then, be a possible alternative to fill-in for many drills of this type.

Most of the other examples of insertions of Type B, 3.5(Dacanay), 3.41(Robinett), 3.117(Stack), 3.132(Stack), 3.142(Stack), and 3.143(Stack), would submit to the findings outlined in the discussion of 3.34, with some minor adjust Only the first two are interlocked drills. In the case of 3.117, both ppp and CFF are required. 3.142 and 3.143 are best served with a prompt. As formalized drills, they would require item-specific instructions. In 3.142(Stack), for example:

```
ccf(1):     "Use ' read '"
ccf(2):     "Use ' eat  '"
```

3.9.2.3 Simple insertions of Type C

One would expect insertions of Type C to be fairly similar to insertions of Type A in their presentation. Again, the cue and the response model must be presented separately. A major difference is that whereas with Type A insertions the cue tends to be isolated, in Type C insertions it tends to appear within a cue framework, which opens up the possibility of computer-generated cues where the cue framework is universal (CCC[] + ccc --> ccg). Use of computer-generated cues would provide for economies but, in most cases, would not greatly affect the operation of the drill. Example 3.20(Rivers), which is typical, could be presented as a fill-in exercise:

P-template #1

```
I:   TTT      Insert the correct pronoun
C:   ccc      Do you see my uncle over there?
M:   mmm[]    Yes, ---'s across the road.
?:   ?->rt       ?
```

or as a multiple-choice exercise:

P-template #2

```
I:   TTT      Choose the correct pronoun
C:   ccc      Do you see my uncle over there?
M:   mmm[]    Yes, ---'s across the road.
     rrn           a.  he/b. xxx/c. xxx
?:   ?->rm    ?
```

Similar to 3.20 are most of the other examples in this group: 3.46(Robinett), 3.47(Robinett), 3.62(Cook), 3.63(Cook), 3.64(Cook), 3.65(Cook), 3.66(Cook), 3.106(Stack), 3.111(Stack), and 3.114(Stack). In 3.106 and 3.111, the cue framework and the response model are identical. In 3.66 the response model is so simple amd the nature of the task is such that free-response would serve equally well in soliciting the response as would fill-in or multiple-choice.

Examples 3.28(Rivers) and 3.29(Rivers) present complications. Both are interlocked drills. In 3.28 there are instances in which the learner should have the option of choosing either insertion-slot, as they are both correct, and other instances where no such choice is possible. The authoring system must offer a mechanism for switching presentation templates (and answer analysis templates). To maintain the interlocked nature of the drill, the template for item(x) select which cue to present, based on the learner's choice in item(x-1):

P-template #15

```
I:   TTT
C:   ccc(x-1)  Janet read her mother the letter.
M:   mmm(x-1)  Janet read -- the letter --.
?:   ?->rl=m,rt          ?              ?
------------------------------------------
I:   TTT
C:   ccc(x)-a  Janet read her the letter.
     ccc(x)-b  Janet read the letter to her.
M:   mmm(x)    Janet read -- to her.
?:   ?->rt               ?
```

In 3.29 the learner is free to provide a wide range of responses. Again, the learner should be given a choice of insertion slot. In order to maintain the interlocked nature of the drill, the cue and response model for item(x) would have to be generated on the basis of a choice of template, and on the learner's previous response to item(x-1). There would be no way to pre-determine either the word provided in item(x - 1), or the slot chosen.

P-template #16

```
I:   TTT
C:   ccc(x)     I gave it to her.
M:   mmm(x)     I gave -- to her  --.
?:   ?->rl=m,rt       ?          ?
------------------------------------------
I:   TTT
C:   ccg(x+1)-a     I gave [aaa(x)] to her.
     ccg(x+1)-b     I gave her [aaa(x)].
M:   mmg(x+1)       I gave -- [aaa(x)] --.
?:   ?->rl=m,rt           ?           ?
```

3.9.2.4 Simple insertions of Type D

There is only one example of a Type D insertion in the data, 3.33(Rivers). The way that 3.33 is best handled depends on whether the focus is to be only the correlative change or it should include the learner's ability to isolate the negative element as well. In the former case, the formalized cue is not really necessary. Simply a prompt and the response model need be provided, as in the example below:

P-template #17

```
I:   TTT       Insert the correct form into the  model,
               based on the change observed
P:   ppp       They haven't any coffee.
M:   mmm[]     They have -- coffee.
?:   ?->rt             ?
```

It would be difficult to handle the latter case, of isolation and removal of the negative element coupled with the correlative change, as a fill-in exercise because the presentation of mmm[] would reveal the answer. Free response would probably be the most suitable presentation method. On authoring systems that could not handle free-response, holistic multiple choice would be the second and less attractive possibility.

P-template #18   3.33(Rivers) Free Response

```
I:   CFF       Delete the negative elements
P:   ppp       They haven't any coffee.
?    ?->rs     ?
```

P-template #19

I:	TTT	Choose the correct negative form
P:	ppp	They haven't any coffee.
M:	rrn	a. They haven't any coffee/b. xxx/c. xxx
?:	?->rm	?

3.9.2.5 Simple insertions of Type E

In Type E insertions, the learner has a great deal of freedom to choose the form of the insertion. In such cases, the needs of the answer-analysis phase will be the determining factor in choosing which presentation template to use. In example 3.32(Rivers), the learner can insert any response which is grammatically and semantically correct, into any desired slot. Furthermore, the drill is interlocked, so that the learner's choices are reflected in the following item. A sophisticated authoring system could conceivably have the potential to generate a response model: mmg based on a complicated framework, MMM[], in conjunction with the learner's material provided in the previous item, aaa(x-1). It is difficult to see how aaa(x) could be analyzed for correctness without the involvement of a grammatical parser endowed with semantic interprative capabilities. Fill-in or free-response would thus be ruled out. The exercise could be done as a simple multiple-choice, but the learner would lose the freedom to choose the insertion slot. The most faithful reproduction of the original on the computer would, therefore, be provided by holistic multiple-choice.

P-template #20

I:	TTT	Choose the correct expansion of the given sentence.
P:	ppp	The man crosses the street.
M:	rrn	a. The tired old man crosses the busy street.
		b. xxx
		c. xxx
?:	?->rm	?

Example 3.36(Rivers) is somewhat more rigid. Formal instructions are given as to the nature of the insertion, and the learner has freedom in areas that do not directly affect the goal of the drill. This could be performed as an insertion, with an answer-analysis algorithm capable of checking for "to" or "not to" in the learner's response. The rest of the learner's response would have to remain unanalyzed, a situation which many teachers would find unnacceptable. If total control over the correctness of the learner's response were required, the best option, again, then, would seem to be multiple choice. As the insertion slot is fixed, simple multiple choice would probably be adequate.

P-template #21

I:	TTT	Choose the ending that provides a correct infinitive construction
M:	mmm[]	She had decided []
	rrn	a. to marry him./b. xxx/c. xxx
?:	?->rm	?

3.9.2.6 Re-order of model elements

Example 3.7(Dacanay) is typical of the exercises where the goal is the re-order of model elements:

P-template #22

```
I:   CCF     Re-order the parts of the statement so
             as to make a question .
M:   mmm     The girl is  ready.
               [1]    [2] [3]
?:   ?->ro   ?
```

Examples 3.8(Dacanay), 3.14(Dacanay), 3.127(Stack), and 3.129(Stack) are similar. All are formalized drills. All require that the model, mmm, be presented. Multiple choice is really the only way to handle exercises of this type. While with some authoring systems it might be expedient to present them in free response mode, they would still essentially be multiple choice activities. Instead of typing the numbers directly, the learner would be typing word(1), word(2), etc. There would still be no opportunity for insertion or creation of new elements.

3.9.2.7 Production of new elements

The last of the simple drill types represent activities where the learner must produce new elements in response to a cue, but without any explicit response model. This results in a simple presentation template:

P-template #23

```
I:   CCF        Answer the question truthfully .
C:   ccc        Why didn't they come home before midnight?
?:   ?->rs      ?
```

All of the examples of this type, 3.31A(Rivers), 3.37(Rivers), 3.38(Rivers), 3.39(Rivers), 3.54(Cook), and 3.145(Stack), could all be presented in a similar format.

With authoring systems that could not handle free response in the answer-analysis phase, the only alternative for the presentation of this type of exercise would be multiple-choice, as in the following example from 3.31A:

P-template #24

```
I:   CCF        Which of the following best represents
                the result of following the instructions
                given ?
C:   ccc        Tell George your name is Ronald.
M:   rrn        a.   George, my name is Ronald.
                b.   xxx
                c.   xxx
?:   ?->rm      ?
```

At first glance, many teachers would probably find the multiple-choice version of these activities to be weaker than free-response one. The question of which approach is superior evaluated further when the answer analysis phase is considered in section 3.10. With some examples, of course, such as 3.38, multiple-choice is not a viable option because the learner must provide original material.

3.9.3 Presentation options in the data: Complex drills

The analysis of complex drills has led to their categorization into those involving the multiple application of the same simple operation, the application of different operations, or the choice of operation. Where a response may involve several factors, or where more than one discrete response is required, it might be simpler to formulate templates that would execute the drill as two separate and subsequent simple drills. It seems pedagogically advantageous, however, to structure the learner's task so as to preserve the unified nature of the activity. Computer realisation of the drill should not require the learner to provide separate, discrete responses where this does not seem natural. Internally, of course. the learner's single response may have to be analyzed according to multiple factors, and so result in a range of feedback options much wider than might derive from a simple drill.

3.9.3.1 Multiple operations

The examples of insertion + insertion contained in the data, 3.12(Dacanay), 3.22(Rivers), and 3.134(Stack) all exhibit the need for separate prompts (P:), have the cue imbedded in the response model (C:+M:), and have fixed insertion slots. In other words, they resemble very much simple insertions of Type B, except that there are two elements to be inserted into two discrete slots. The form of the insertion element for each slot, rather than the choice of element to place into either slot seems to be the major motivation in all three examples.

The possible fill-in template shown below for example 3.22, which seems most typical, resembles very much P-template#13, which was applicable to most of the simple insertions of Type B. The only difference is in the response elicitation mechanism, which in this case must be capable of eliciting two fill-in elements for two separate blanks, leading, in all probability, to a more sophisticated response analysis and feedback template.

```
P-template #25

I:   TTT       Make the necessary changes
P:   ppg       He brings his lunch
M:   mmg       You --- --- lunch
?:   ?->rt1,rt2    ?   ?
```

Holistic multiple-choice could be adopted to reduce the learner's input to a single item, so reducing the corresponding number of feedback possibilities:

P-template #11b

```
I:   TTT        Choose the correct sentence
P:   ppg        He brings his lunch
M:   rrn        a.  You bring your lunch.
                b.  xxx
                c.  xxx
?:   ?->rm
```

(The prompt is not operationally required this example. The use of ppg and mmg reflect the interlocked nature of example 3.22.)

In those examples of consecutive insertions in the data, the first is a Type A insertion and the second a Type B insertion. Two possibilities present themselves. In example 3.26(Rivers), the insertion slots for both are fixed and the drill can be presented in a manner very similar to the previous examples:

P-template #26

```
I:     CCF          Convert the statements to questions.
C:     ccc          Peter has a new car.
C:+M:  mmm            [] Peter   [] a new car?
?:     ?->rt1,rt2     ?          ?
```

P-template #11b

```
I:   CCF        Choose a correct question
C:   ccc        Peter has a new car.
M:   rrn        a.  Does Peter have a new car.
                b.  Has Peter got a ner car.
                c.  xxx
?:   ?->rm
```

Example 3.36(Stack) also shows fixed slots for both insertions, and could be handled similarly, although, in this case, C:+M: could not be used and a prompt is needed:

```
.cp6
    P-template #26b

    I:   CCF          Answer the question
    P:   ppp          The boy is polite
    C:   ccc          How does he speak?
    M:   mmm          The boy  [] []
    ?:   ?->rt1,rt2         ?  ?
```

Holistic multiple-choice and free-response are also possible. The relatively predictable response model makes free-response an ideal, and probably more natural, possibility.

Example 3.31b is the third example with fixed slots. As a fill-in, this would be extremely clumsy, given that four discrete responses would have to be elicited. Free-response is theoretically possible, as a fixed-response model could be established. The response is quite long, however, making the learner's task difficult and error-prone. A very complex response-analysis template can be envisioned, which might well be beyond the reach of most authoring systems. The best approach to 3.31b seems to be holistic multiple-choice.

Examples 3.24(Rivers) and 3.43(Robinett) exhibit a choice of insertion slots for the first insertion. With 3.24, the slot for the second insertion is also variable.

There would be so little in the way of explicit response model that the fill-in mode would almost, de facto, become free- response, a very viable option here, given the fixed nature of the implicit response model:

P-template#27

I:	TTT	Insert the cue and make other necessary changes
P:	ppg	She brings too many pencils to school.
C:	ccc	You
?:	?->rs	?

By dealing on the level of word units, the re-order, then re-order example, 3.16 (Dacanay), could be treated as a single re-order multiple-choice, and the discussion and template shown in 3.9.2.6 would apply:

P-template #22

I:	CCF	Combine the two questions, putting the yes/no question first.
M:	mmm	Who is he? Do you know? [1] [2] [3] [4][5] [6]
?	?->ro	

The possible combinations of six elements far exceed those of the three elements shown in the example of the simple re-order operation shown in 3.9.2.6, however and a more complex response-analysis-and-and feedback template could be predicted.

3.9.3.2 Mixed operations

Leaving aside the use of holistic multiple-choice or free-response, which are everpresent possibilities and which can reduce the learner's task to a one-step operation, the insertion + removal exercises represent a new kind of problem. Simple removal was not discussed in 3.9.2 because no examples were found which involved removal of redundant elements alone.

The learner's task in a removal is to determine what needs to be removed and to indicate this. No new REMs are required for this operation, which can be performed either as a fill-in/short answer or as a multiple-choice response to an explicit question or instruction. In 3.13(Dacanay) the initial insertion of the cue and the subsequent identification of redundant elements would have to be done separately. The second operation could be presented in the form:

```
P*template#28

I:     TTT        Indicate the words to be removed
M:+Rd  mmm        The noisy children who were making a lot
                  of noise were sent to bed
?:     ?->rt
```

```
P-template#29

I: .   TTT       Indicate the range of words to be removed
M:+Rd mmm        The noisy children who were making a lot
                 [1] [2]    [3]      [4] [5]  [6]   [7][8]
                 of noise were sent to bed.
                 [9] [10] [11] [12][13][14]
?     ?->rm
```

The removal portions of the rest of the examples, 3.16(Dacanay), 3.30(Rivers), and 3.131(Stack), could all be handled in the same way.

Robinett provides the only examples of re-order, then production of new elements. These exercises, 3.44 and 3.45 are naturally two-step operations and would respond to being handled as two separate simple drills.

While there are two factors involved in the response to exercise 3.146(Stack), it would best be treated, presentationally as a simple production-of-new-element drill with a CFF such as "Answer the question using ' always ' ". The answer analysis and feedback could be addressed towards separate consideration of the correct formation of a response and the correct use of "always".

3.9.3.3 Choice of template

Exercises 3.27(Rivers) and 3.130(Stack) are not complex in themselves, but require a more complex authoring system, in that different items require the application of different simple templates. Conceptually, this does not seem difficult. Each discrete drill item could be flagged as to which template should be applied. These have been considered as separate category and treated as complex drills because many authoring systems are incapable of switching templates within a sequence.

3.10 Factors influencing the answer-analysis phase of a drill

There are several factors influencing the answer-analysis phase which must be examined before possible answer analysis templates for the drills in the data can be postulated. The content of the drill, largely irrelevant to the choice of P-template, becomes an important factor in answer-analysis as it directly determines the response-set and the answer-set of a drill. Response-set and answer-set must be defined and their expression in the data examined. The approaches to answer-analysis currently in use, or proposed for CALL, must be discussed and those not applicable to the current discussion disgarded. The concept of response possibility must be defined, and the effects of the response-elicitation technique chosen for a particular presentation template on the range of response possibilities examined. Finally, techniques for controlling the range of response possibilities require examination.

3.10.1 An Examination of the response-sets in the data

The response-set can be considered the set of predictable or expected responses which derive naturally from the content of the stimulus, and from which the learner chooses the correct response. If, for example, the stimulus calls for the learner to provide the correct subject pronoun, then it would be reasonable to consider the response-set to be the set of all subject-pronouns. Responses that were outside the response-set would be invalid or unrecognizable responses rather than simply wrong responses. The symbol {} has been adopted to designate the concept of a set, and {R} or {r} to denote response-sets.

The nature of the response-set affects mostly the type and range of feedback that can be given to the learner after an incorrect response. Where there is no response-set, for example, there are only two possible outcomes of the answer analysis: the response matches the correct answer or it does not. With a minimal response-set, three outcomes at least are possible: correct; not correct, but within the response-set; and not within the response-set.

A response-set can be universal, {R}, meaning that all items draw their responses from the same set, or item-specific, {r}, meaning that each drill item has its own unique response-set. In 3.5(Dacanay), for example, all of the items relate to the choice "do/does", while in 3.142(Stack), item one relates to the choice "read/reads" and item two to the choice "eat/eats". When the response-set is universal, it can be larger and more structured than when it is item-specific.

The response-set can also be implicit or explicit. In the latter case it will be apparent to the learner, from the nature of the task or from the presentation template chosen, what the response possibilities are. Multiple-choice activities, for example, will automatically have explicit response-sets. The symbol #{r} denotes explicit response-sets.

There are some variables which are particular to the multiple-choice mode. With multiple choice the actual response-set differs from the true response-set. {R} may be {do/does}, but the learner's response options may be {a/b} or {1/2}. We will symbolize the actual response-set as @{R} to differentiate it from the true response-set {R}. The explicit response set may encompass all of the natural, implicit, response-set, @r = {r}, or it may only represent a manageable portion of the larger response-set, @r < {r}.

The size of the natural response-set and its degree of "closedness" is determined by the content of the exercise. Where the natural response set is larger than desired, it can be reduced by the presentation template. In

3.5(Dacanay), for example, the prompt:

P: Where does Mary live?

reduces the possible forms that could be inserted into the model:

M: Where [] Jose live?

With the prompt, {R} = {do/does}, without the prompt it would be necessary to include {can/ might/ may/ should/ would/ shall/ ...}. In cases where there is no natural limit on the number of plausible, expectable responses, where there is, in other words, no finite response-set, an artificial limit may be imposed by using multiple-choice.

The response-set can be of very limited in size, containing perhaps from two to half a dozen elements. This can be called a "very small" response-set and can be shown symbolically as:

{r} = {aaa/bbb}
{r} = {aaa/bbb/ccc}
{r} = {aaa/bbb/ccc...}

where aaa represents the correct response and bbb, ccc represent incorrect but predictable responses.

3.61(Cook), for example, has an item-specific response set of two: {r} = {aaa/bbb} = {swimming/ skating}. 3.132(Stack) has a universal response set of four: {R} = {aaa/bbb/ccc...} = {will/ 'll/ would/ 'd}. 3.62(Cook) has one of seven elements, the days of the week. While all the size terms to be discussed are relative, those response-sets where all the elements could be displayed simultaneously for the learner, if desired, could be considered as "very small". In cases of multiple-choice, the explicit response set, #{r}, would equal the actual response set {r}.

Where the set of responses is larger than what could comfortably be displayed to the learner, but still small enough to be handled without resorting to techniques designed for large databases, it can be refered to as relatively "small" and be represented by the symbols:

{r} = {aaa/bbb/.../nnn}

In 3.118(Stack), for example, the possible re-statements of "il se repose" is limited, but still represent a substantial number. Similarly, in 3.144(Stack), the number of ways, correct and incorrect, to respond affirmatively to the question "Do you like music?" is considerable.

A patterned or structured response-set is one where types of responses are grouped together. In 3.117(Stack), it is possible to phrase responses that illustrate the choice of the correct, but improperly formed, tense, "est commande"; the wrong tense, "commandait"; or improper verb agreement, "avez commande". While the total number of possible responses may be considerable, they can be categorized into a few general types of error. Errors of the same type can be treated in a similar manner.

Where the set of responses to too large to handle using normal techniques, but is still, nevertheless, finite, it can be termed relatively "large" and represented as:

{r} = {database}

In the original, non-computerised execution of the drill, the learner would have to choose from among a large set of possible responses stored as part of "general knowledge". On the computer, database storage and retrieval techniques would be required, to determine if a response were in the acceptable set. In 3.29(Rivers), a database would be postulated containing a large list of names of objects which for which "it" could substitute, a list of proper names for which "her" could substitute, and so forth.

Table 3.12 examines the variations in response-set that are evident in the data. Almost all of the examples exhibit closed response-set, although there is no apparent pattern in the variation between universal and item-specific response-sets. 3.66(Cook) and 3.111(Stack) are shown as having no finite response-set because the desired response is not part of a larger set. The insertions of

Type E, 3.32(Rivers) and 3.36(Rivers), and some of the production-of-new-element drills, 3.31A(Rivers), 3.37(Rivers), and 3.39(Rivers), have response sets too large to be defined. All the drills classed as re-order of model elements seem, by their nature, to exhibit universal response-sets.

Most of the insertions of type A and all of the re-order drills exhibit inherently explicit response sets. Explicit response sets will have to be very small and so, in the table, a star was not placed in column 5 if there was one in column 4. Taking columns 4 and 5 together, we can see that most of the drills in the data, both simple and complex, with the exception of those that involve the production of new elements, have very small response sets.

3.10.2 An examination of the answer-sets in the data

The answer-analysis operation involves more than simply determining if the learner's response matches the "correct" response. The the learner's response must be identified by matching it against one of the elements in the response-set, of which the "correct" response represents only a single possible case. For those cases where there is a degree of correctness, perhaps a "best" answer and a "second-best" answer, the approach is to consider this situation as identifying a correct answer and one of the "incorrect", but predictable answers.

From the definition given of a CALL exercise, it is possible to define at least one "correct" or acceptable response, which can be symbolized as aaa or AAA. It is also necessary to assume a finite set of correct responses, because an infinite set be equivalent to having no "correct" (as opposed to "incorrect) responses at all and anything the learner supplied would therefore be acceptable. Normally the correct response would be assumed to be item-specific, aaa, although a universal correct response, AAA, exists in one example from the data, 3.111(Stack).

There may be a single correct response, aaa, or a set of correct responses {a}. As in the case of response-sets, considered above, where there is {a}, it may either be "very small" set:

{a} = {aa1/aa2}
{a} = {aa1/aa2/aa3}
{a} = {aa1/aa2/aa3...}

a relatively "small" set:

{a} = {aa1/aa2/.../aaN}

or a relatively "large: set:

{a} = {aa1/aa2/.../aaZ}

The data is examined in table 3.13 with regard to variations in answer-set. Most of the examples have only one correct response per item. To single out a few examples of a very small answer-set, it is possible to mention 3.11(Dacanay), where "now" can be inserted into one of two slots; 3.120(Stack), where the form can be "are not" or "aren't"; or 3.144(Stack), where the response can be "do", "like it", or "like music". A larger, but still relatively small answer- set can be found in examples such as 3.31A(Rivers), where one can expect answers such as "George, my name is Ronald.", "My name is Ronald.", "I'm Ronald.", etc. Large answer-sets are exemplified by exercises such as 3.32(Rivers), where virtually any grammatically and semantically correct insertion is acceptable, or 3.29(Rivers), where any noun that can replace "it" is acceptable.

The set of {a} may represent discrete, separate elements or may simply be alternative forms of the same basic element. The latter case is represented in the formula:

{a} = {aax/aay...}

This can be exemplified in 3.34(Rivers), where {a} = {"would"/ "'d"}, both of which are forms of the same word.

For most cases, the correct answer can be predicted by the author and entered into the data as a separate, discrete element:

aaa < data

Where the response-set is structured and relatively small, it may be possible to generate the correct answer based on other elements in the data, such as the cue, in combination with a universal rule or set of rules:

aaa < RULE

This possibility may be used, even where not needed, to save time for the author and to save on data-storage space. Such cases are not treated here. In 3.63(Cook) and 3.114(Stack), however, answer-generation by rule is liable to be the preferred avenue, because of the ease with which it could be handled by the computer.

There are other instances, however, where the correct response cannot be predicted, but may depend on some choice that the learner has already made. Two answers may be "linked" in such a way that the first response determines the second. In 3.28(Rivers), for example, the choice of insertion slot made by the learner determines which form of the answer is correct.

Response-processing can be holistic and attempt to relate the learner's response, as a whole, to the correct response. Alternatively, it can be partial and match pieces of the response against pieces of the correct answer, to produce feedback based on the discrete elements. With longer and more complex responses, partial analysis becomes the only feasible avenue for analysing a textual response. In 3.120(Stack) processing the response on a word-by-word basis would allow the computer to realize that, in a case of "not are", the words are reversed, whereas holistic processing would simply see the response as wrong. In cases such as 3.36(Rivers), if performed in the fill-in mode, holistic matching is not possible, as the entire learner-response cannot be predicted. All that is known is that the first word may be either "not" or "to" and that if the first word is "not", then the next word should be "to".

With simple one-word reponses, partial analysis is probably not required. Indeed, in cases such as the re-order of two elements, partial-response analysis may not be possible.

In the discussion of response sets, @{R} was defined as the actual response-set as opposed to the true response-set {R}. Similarly, the symbol @a is defined to stand for the actual response to a multiple-choice. Thus, in 3.6(Dacanay), the learner's choice of [2] stands for "'tonight' replacing 'tomorrow' in the slot between 'leaves' and 'at seven'".

3.10.4 General approaches to computer-based answer analysis

Pusack(1983) outlines five basic mechanisms for answer analysis in CALL: Non-evaluation, right/wrong evaluation, pattern mark-up, error anticipation, and parsing.

This author (Pusack, 1983, p.55) finds non-evaluation to be a powerful pedagogical approach in many circumstances. Typically, the learner is asked to formulate an answer mentally, the correct answer is displayed, and the learner is asked to compare the two. For free-form responses, where only a likely model for the correct answer can be provided, non-evaluation may be the only viable technique. The disadvantages lie in the possibility that the learner may not detect an error, or may not understand an error that has been detected. There is very little potential in these circumstances for computer-controlled branching. The rigorous interpretation of CALL activities used for this study calls for them to be mediated by the computer, and therefore self-mediation or non-evaluation must be excluded from the discussion.

Right/wrong evaluation involves the matching of the learner's response against a "correct" response, or set of "correct" responses. Techniques can be applied which reduce mismatches due to deviations such as errors in spacing, capitalization, punctuation, and minor spelling variations (Pusack, 1983, p.55-56).

Pattern mark-up involves a non-linguistic matching of the learner's response against the response model. Missing, extraneous or misplaced letters or words are indicated, using some system of proofreading symbols. Pusack(1983, p.61) offers the following illustration from the PLATO system:

```
Response Model:
The quick brown fox jumps over the lazy dog.

Learner's response and mark-up:
the brown quick fox jumpd baerr the big lazy Dog.
*          <          = = = xxxxx       xxxx *

      where  *  stands for     capitalization errors.
             ^                 missing element
             <                 element to shift leftward
             ===               mis-spelled word
             xxx               unrecognizable  or  extra
                               word.
```

While a pattern mark-up algorithm can be applied to fill-in and multiple choice exercises, it should be evident that its true power lies in dealing with sentence-length free response.

Several serious limitations to pattern-mark up have been identified, some of which stem from the mechanical nature of the method of analysis involved. Word-order errors are described without reference to grammatical norms. Inflection errors, root spelling errors, diacritic errors and run-on words are all described in a similar manner (Hart, 1981, p.9). Since there is no syntactic model within the computer, no explanation of an error can be given to the learner (Pusack, 1983, p.59). It is difficult to imagine how the information obtained from a pattern mark-up analysis could contribute to computer controlled branching, modelled on the learner's performance. Furthermore, the algorithm itself often manifests shortcomings. Morphological analysis tends to take precedence over syntactic analysis (Pusack, 1983, p.59). The resulting mark-up is based on the hidden response model, often betraying the correct syntax, which may be the point of the exercise. Pattern mark-up algorithms tend to "hypercorrect", registering elements or patterns which were not intended by the learner. Pusack

(1983, p. 59) offers the following example:

Response Model:	Ich gehe in die Stadt.
Learner Response:	Ichn gehe die Stadt.
Mark-up:	Ich< gehe ^n die Stadt.

where the algorithm picks up the extra "n" at the end of "Ich" and interprets it as part of the missing word "in". The possibility that the the learner simply forgot to include "in" seems more likely than the possibility that it was mispelled and placed at the end of the subject, "ich". Finally, pattern-mark up assumes that a static response model can be predicted. There is no accommodation built in for cases in which, for example, the choice of a word in one part of the sentence pre-determines the choice of a word later in the sentence.

For the purposes of the present survey, the computer is allocated the role of surrogate teacher. Pattern mark-up does not offer the subtlety of analysis that could be displayed by a human instructor. Furthermore, with authoring systems that are based on this approach, there is no mechanism for teacher-authors to encode and include their expertise. It must be recognized that pattern mark-up is a viable technique for authoring systems to employ for answer-analysis. It must be excluded, however, from consideration here, as the model being developed cannot be applied to it.

Pusack's (1983, p. 61) fourth category, error anticipation, is an extension of right/wrong analysis and deals with possible avenues to follow, once it has been determined that the learner's response is incorrect. Essentially, an attempt is made to match the learner's response against one of a list of expected wrong answers. For each wrong answer there may be a particular diagnostic message as feedback, or a particular branch to another item may be made. In keeping track of error types, the computer may be building up a model of the learner's global performance, which can also be used to initiate particular branches.

Pusack (1983, p. 61) draws attention to two drawbacks to this technique. First, it is necessary to encode all of the possible errors that may occur, a task which the author may find unmanageable. Second, a bold leap is made from the existence of an error to its probable cause, assumed in the explanation for the error which is provided as feedback. Such explanations must be carefully designed and must take into account all possible causes for a given error.

The final answer-analysis technique, that of full grammatical and semantic parsing of the learner's response, will eventually overshadow all the others. Some theoretical point in the future can, perhaps, be postulated in which the computer will have the same linguistic and pedagogical expertise as the human teacher, at which point authoring systems will no longer be necessary. Currently, however, parsing is an imperfect technique, and requires resources beyond what would normally be available in a CALL environment (Pusack, 1983, p.63). Pusack coins the term "Pseudo-parser" to cover algorithms which match parts of the learner's response against parts of the response model or models, providing some insights into the nature of the differences that become evident.

To exclude parsers from the discussion, while including "pseudo-parsers", an arbitrary distinction must be made. The term "parser" is reserved for any algorithm which must grammatically and semantically label elements, in order to perform the analysis. Pseudo-parsers will be able to perform their analysis by matching elements of the learner's response against lists of possible matches for the same slot. Pusack's example, introduced above, can be used to illustrate the two:

> Given a learner's response: Ich gehe die Stadt., a parser would begin by labelling the elements as Subject+verb+object. It would tag "gehen" as a verb of motion and realize that it could not take

a direct object. Of the list of prepositions that could follow "gehen", only "in" could collocate with "die Stadt". It would then signal the missing "in".

The pseudo-parser would have a stored response model:

M:word(1)="Ich"
M:word(2)="gehe"
M:word(3)="in"
M:word(4)="die"
M:word(5)="Stadt"

The learner's input would be broken down into words:

L:word(1)="Ich"
L:word(2)="gehe"
L:word(3)="die"
L:word(4)="Stadt"

The computer would then register that there was a word missing. It would find that L:(3) did not match M:(3) and L:(4) did not match M:(4). A trial phase-shift of one element would obtain a match for L:(3+1) and L:(4+1). The missing element would then be identified as M:(3) or "in".

In the relatively closed, structured environment of the CALL exercise, as it has been defined, the pseudo-parser can appear to the learner to possess a grammatical "understanding" of the input and to have para-human flexibility. As the environment becomes more open and less structured, the pseudo-parsing technique becomes too complex and must be replaced by a true parser. Assuming that most authoring systems are incapable of providing that option, the author must introduce "structure" of a non-linguistic nature by forcing the activity into another mode, such as multiple choice.

In the discussion of answer-analysis, there is a concentration on what Pusack terms "error anticipation", realizing that what he calls "right/wrong" analysis may be the only applicable approach with drills that have no finite response-set or with authoring systems that do not allow for error-anticipation. The pseudo-parser approach essentially emerges as a variation of error-anticipation. Instead of examining the response holistically, each component is treated separately, but still in an error-anticipation aspect.

3.10.5 The effects of variation in response-possibilites, and preprocessing of input

The open/closed nature of the response- and answer-sets of activities is largely an inherent feature which does not depend on the medium of presentation. An entirely separate variable is the open/closedness of the range of response possibilities. While the response-set depends primarily on the content of the drill, the range of response possibilities is a function of the presentation template and of the response-elicitation technique in particular. Only to the degree that these affect the presentation template do the form and content of the drill influence the range of response possibilities.

The nature of this new variable may best be illustrated with an example. Consider possible responses to a simple yes/no question. From the point of view of content, such a question is completely closed. The response set contains only two elements, "yes" and "no", only one of which can be correct.

If the question is posed as follows, allowing for free input:

Is that correct ?

the range of possible responses is wide, and the corresponding complexity of the answer-analyzing algorithm will have to be very great. Some possible responses are:

YES, Yes, yes, yES, yyes
yis, yea
sure, I think so, right on
I don't think so, I think not, no way
naw
no, nno, No, NO.

The range of response possibilities would render virtually useless an attempt to directly match the response against the words "yes" or "no". If the response does not match with "yes", the conclusion that the learner intended to signal "no" does not follow automatically.

The application of pre-processing techniques can greatly reduce the range of response possibilities by cancelling out, or preventing, mechanical variations such as upper/lower case and mis-spellings.

One of the most important such techniques of this kind is that of converting all letters to a single case, either upper or lower. YES, yes, Yes, yES, etc. can then all be treated as YES or yes.

The computer's reaction to extra spaces in the input frustrates many learners. A computer may find that " YES" does not match "YES". Many programming environments are sufficiently sophisticated to ignore leading and trailing blanks, but extra blanks between words can still cause problems. If the desired response is "aren't you?", then "aren't you?" is likely to be rejected. A pre-processing algorithm is required that can strip leading and trailing blanks and reduce inter-word blanks to single spaces. (The problem posed by "aren't you" can also be dealt with using partial processing, as is demonstrated below.)

The problem of spelling errors and alternative forms may be dealt with during the actual answer-analysis. A pre-processing alternative, however, is to freeze the keyboard and only allow certain letters to pass. In the "yes/no" example above, the letters Y,E,S,N,O could be programmed as the only active letters. The learner could type "Yesno", but would be unable to type "I think so" or "no way!".

All answer-analysis by computer is ultimately performed by determining if two elements match. The subtlety with which this basic operation can be completed affects the range of response possibilities that can be accommodated.

Where the learner's response can be matched against more than one possible element, variations such as "YES", "SURE", "SI", "OUI", "JA" etc. can be accepted. This is not a good approach for handling the majority of mis-spelling situations, as all possible combinations would have to be predicted in advance: yes, yyes, yees, yess, yesss, ysys, etc.

Many authoring systems allow for some degree of "fuzzy matching". The simplest approach is to look at only part of the learner's response as compared with the relevant part of the model. In the "yes/no" example, the computer could compare the first letter of the response with "Y" and thereby accept a much wider range of responses as being affirmative. Some systems allow for "dummy" or "wild card" characters to be placed in the answer, allowing for a match with "Y##". Other systems allow can calculate the percentage of agreement between the two elements. Thus, "EES", which the algorithms shown above would miss, would be found to have 66% compatibility with "YES". Pusack (1983, p. 56) discusses pseudo-phonetic fuzzy matches, whereby "ROOSEVELT" could be matched with "ROSEVELD" or "ROSEFELD".

Fuzzy matching is a useful tool for suppressing the errors due to spelling variations that so often frustrate learners are working on the computer, but it visibly demonstrates that the surrogate cannot entirely replace the teacher. The technique is useful only in some situations, as there will always be a certain amount of uncertainty and guesswork involved in planning the outcome to a response. A simple fill-in exercise illustrates this point:

C: talk
M: He ____.
?: ?

The learner's response, ???, must now be identified. Using wild-card characters for the storage of the correct response, the situation could arise where the learner's response, ???="tawks", would be considered "correct" when matched with the correct answer, aaa="####s", while "talk" would be considered incorrect.

However sophisticated the response-analysis techniques, it is impossible to preclude the "grey area" of learner input that cannot be identified. While the handling of this "grey area" in a pedagogically acceptable fashion can be considered part of the skill in designing CALL activities which authors must possess, their task can be made easier on systems where uncertainty is reduced to a minimum.

The "grey area" phenomenon can cause the computer to make "mistakes" in analysing learners' responses to certain stimuli. In other situations, authors must resign themselves to the fact that certain parts of the learners' responses simply cannot be analyzed.

One avenue towards solution is to reduce the range of response possibilities. The question from the earlier example could be posed as follows:

Is that correct (Y for yes/ N for no) ?

reducing the response possibilities to two, a closed set. Any input that does not match the set of response possibilities can be immediately rejected, regardless of its acceptability or non-acceptability from the viewpoint of content.

Chapter 3: Research Design and Findings

Multiple-choice type questions inherently have a closed set of response possibilities. Yes/no, true/false, yes/no/maybe and other similar question formats can be considered as variations on multiple choice.

Even in situations where the student is called upon to type an entire sentence, the set of response possibilities can be considered as closed. Compare: response possibilities. Consider:

(a) Write these words in the correct order:

BOOK GIVE I THE HER

and (b) Write the numbers in the order corresponding to the correct sentence

```
BOOK  GIVE  I    THE  HER
[1]   [2]   [3]  [4]  [5]
```

The only feature that format (a) from format (b), above, is the additional possibility of errors in spelling and reproduction in the former case. The response-set can be considered closed when it is possible to determine whether the answer is composed of the required, albeit possibly malformed, components.

As illustrated in the "yes/no" example shown earlier, the complexity of any response, from the point of view of open/closedness of content, can be further increased by allowing a looser range of response possibilities, or simplified by reducing the range.

The availability of some of the techniques described above is an important consideration in discussing CALL applications of an authoring system. These are treated separately, however, and for the rest of this analysis, it is assumed that the applicable pre-processing and fuzzy matching techniques are available.

3.10.6 Basic answer analysis templates

It has been indicated above that while the choice of P-template for particular exercise is relatively independent of content, the choice of answer-analysis-and-reaction template, or A-template, depends both on P-template requirements and upon content, in particular upon the response-set, {r}, and the answer-set {a} .

In the examination of A-templates in terms of the data required, the number and type of operations, and range of possible outcome situations that can be identified, it will be assumed that some reaction to all learner input is involved. The reaction may take the form of immediate feedback, in terms of a message, or some particular branching decision may be made. Reaction to any situation requires, first of all, that the A-template allow the computer to realize that the situation exists.

The link between P-template and A-template is the response-elicitation mechanism or REM. The REM will also, to some extent, determine the range of response possibilities. It can be assumed that the multiple choice REM's (?->rm, ?->rl=m, and ?->ro) lead to a closed set of response possibilities while the fill-in REM (?->rt) and the free response REM (?->s) lead to an open set of response possibilities.

Wherever applicable, the existence of the pre-processing and fuzzy matching techniques described in 3.10.5 is further assumed. Only valid multiple choice responses will be considered. In the case of fill-in, the assumption is that all responses are in the proper case, that there are no extra spaces, and that minor spelling errors are excluded. For free-response, it is assumed that the response has already been broken down into discrete words for partial processing.

The symbol, ???, is used to stand for the learner's response and the symbols, ?1?, ?2?, ?3?, ... ?n?, to stand for that response, broken up into discrete word units, where applicable. The symbol @? will stand for a multiple-choice response.

The simplest A-template that can be postulated is the algorithm required to handle the situation where there is no response set, {r} = 1,[60] and where there is a single correct answer, {r} = {a} = aaa. In this case, aaa would have to come from data. There would be only two possible outcomes, ??? would or would not match aaa. The A-template could therefore take the form:

```
A-template #1

?:    ?->rt ---> ???
R:    {r}=1
A:    {a}=aaa  <---data

MO:   1.   If aaa=??? then situation I
                      else situation II

F:    Situation I:    Learner's response is correct.
      Situation II:   Learner's response is unrecognized
                      (and presumed incorrect).

      where:     ?:   REM leading to response
                 R:   Response Set
                 AGO: Answer Generating Operations (not
                      present in this case).
                 A:   Answer Set
                 MO:  Matching Operations
                 F:   Situations leading to feedback or
                      other reaction
```

Where there is a response-set of two elements, another possible outcome is added:

```
A-template #2

?:    ?->rt ---> ???
R:    {r}={aaa/bbb} <---data
A:    {a}=aaa       <---data

MO:   1.    If aaa=??? then situation I
                       else 2.
      2.    If bbb=??? then situation II
                       else situation III

F:    Situation I:     Learner's response is correct.
      Situation II:    Learner's response is incorrect.
      Situation III:   Learner's response is
                                         unrecognized.
```

Additional elements in a very small response set simply multiply the number of operations and the number of possible outcomes:

```
A-template #3

?:    ?->rt ---> ???
R:    {r}={aaa/bbb/ccc/ddd} <---data
A:    {a}=aaa               <---data

MO:   1.    if ??? is element of {r} then 2.
                                  else situation III
      2.    If aaa=??? then situation I
                            else 3.
      3.* ({r}-1)     If bbb=??? then
                                 situation II * {r}

F:    Situation I:     Learner's response is correct.
      Situation II * ({r}-1):
                       Learner's response, nnn, is
                      incorrect because...reason(nnn)
      Situation III:   Learner's   response   is
                                        unrecognized.
```

Multiple-choice activities inherently have a very small response-set, so the answer analysis would be very similar to the above, with the difference that much less pre-processing of the learner's response would be required and the "grey area" would be reduced to nil.

A-template #4

```
?:    ?->rm ---> @?
R:    @{r}={@a/@b/@c/@d}       <---data
A:    @{a}=@a                  <---data

MO:   1.    If @a=@?    then situation I
                              else 2.
      2.* (@{r}-1)      If @b=@? then
                                  situation II * @{r}

F:    Situation I:      Learner's response is correct.
      Situation II * (@{r}-1):
                        Learner's response, nnn, is
                       incorrect because...reason(nnn)
```

Cases where there are multiple correct answers forming part of a very small answer-set introduce only slightly more complexity.

A-template #5

```
?:    ?->rt ---> ???
R:    {r}={aa1/aa2/aa3/bbb/ccc/ddd} <---data
A:    {a}={aa1/aa2/aa3}             <---data

MO:   1.    if ??? is element of {r} then 2.
                                      else situation III
      2.    If ??? is element of {a} then situation I
                                      else 3.
      3.* ({r}-1)       If bbb=??? then
                                    situation II * {r}

F:    Situation I:      Learner's response is correct.
      Situation II * ({r}-1):
                        Learner's response, nnn, is
                       incorrect because...reason(nnn)
      Situation III:    Learner's   response   is
                                          unrecognized.
```

Where there are two responses elicited, however, the complexity increases greatly:

A-template #6

```
?:    ?->rt1,rt2 ---> ?1?,?2?
R:    {r1}={aaa/bbb/ccc/ddd} <---data
      {r2}={aaa/bbb/ccc/ddd} <---data
A:    {a1}=aaa               <---data
      {a2}=aaa

MO:   1a.  if ?1? is element of {r1} then 2a.
                                   else situation IIIa
      2a.  If aaa=?1? then situation Ia
                           else 3a.
      3a.* ({r1}-1)      If bbb=?1? then
                                  situation IIa * {r}
         ...
      1b.  if ?2? is element of {r2} then 2b.
                                   else situation IIIb
      2b.  If aaa=?2? then situation Ib
                           else 3b.
      3b.* ({r2}-1)      If bbb=?2? then
                                  situation IIb * {r}

F:    Given the same situations as in the previous
      templates, the complexity of the feedback
      increases radically:

      Situation Ia + Situation Ib
      Situation Ia + Situation IIb * bbb
      Situation Ia + Situation IIb * ccc
      Situation Ia + Situation IIb * ddd
      Situation Ia + Situation IIIb

      Situation IIa * bbb + Situation Ib
      etc.

      In effect, 2+({r1}-1) * 2+({r2}-1) possibilities
```

The range of situations requiring a reaction is much greater. It should be evident why the elicitation and simultaneous analysis of three or more responses has not been considered a feasible option.

Templates #5 and #6 could be combined so that multiple correct answers could be accepted, for either or both of the responses elicited.

The limits to the ability to check individually each and every element of {a} and/or examine {r} for a possible match will depend on the authoring system in question. In making the distinction between "very small" and "relatively small", a distinction has been attempted between a first category of cases which it would be reasonable to expect the majority of authoring systems to be capable of handling, and a second group which only very powerful authoring systems would be expected to handle on the same pattern. fashion. An authoring system called Egg, for example, has the power to handle up to 60 discrete matches, each with its own particular reaction (Peuchot, 1983). Where the response and/or answer set is too large to handle as above, different techniques, such as partial analysis or conversion to multiple choice, must be explored.

3.10.7 Answer-analysis Options in the data: Simple drills

On the basis of factors examined in tables 3.12 and 3.13, it is apparent that most of the drills in the data can be covered by the basic answer-analysis-and-reaction templates outlined in 3.10.6. For those cases not covered by these basic templates, more complex templates will be postulated below. As in the case of the presentation templates, the components of the answer analysis templates are the critical factor to be considered. The actual global templates preposed proposed are likely to be very specific to particular situations that are presented in the data.

3.10.7.1 Simple insertions of Type A

3.6(Dacanay) is typical of many of the insertions of Type A present in the data, in that there is a small, explicit response-set and a single correct response. Although the answer may be elicited in the fill-in mode (?->r1=m), 3.6 is essentially a multiple-choice activity and could, therefore, be accommodated by the basic A-template#4.

A-template #4

```
?:    ?->rm ---> @?
R:    @{R}={[1]/[2]}            <---data
A:    @{a}=[2]                  <---data

MO:   1.   If [2]=@?   then situation I
                            else 2.
      2.* (1)           If [1]=@? then
                                situation II

F:    Situation I:    Blank [2] = "...leaves [tonight]
                      at seven", which is correct.
      Situation II:   Blank [1] = "[Tonight] leaves
                      tommorrow at seven", which is
                      incorrect because...
```

Similarly, 3.9(Dacanay), 3.10(Dacanay), 3.21(Rivers), 3.42(Robinett), 3.60(Cook), 3.61(Cook), and 3.125(Stack) could all be handled as above. 3.11(Dacanay) adds the complication of the possibility of two correct answers in some cases, and so would require the multiple-choice equivalent of basic A-template#5.

3.4(Dacanay) and 3.40(Robinett) have a response-set which is very large. Theoretically, such a large response-set could be handled by a computer program. In the case of these rather simple activities, however, the complex answer-processor that would result would probably not be worth the effort. More than likely, an infinite response-set would be assumed and a template such as A-template#1 would be applied.

3.118(Stack), 3.120(Stack), and 3.144(Stack) have response sets that, while larger than those discussed for cases 3.4 to 3.125 above, are still relatively small. The latter two add the complication of allowing for multiple correct responses. Most authoring systems should be able to handle these exercises with basic templates such as A-template#3 and A-template#5. The response-sets show certain patterns, and so the number of different feedback messages required can be reduced to a number far fewer than the discrete number of {r} elements against which the learner's response would would have to be checked.

The responses to these drills could also be handled, and perhaps more efficiently, through a partial analysis by a pseudo-parser, as illustrated in 3.10.4. We can take 3.44(Stack) as an example:

```
A-template#7

?:    ?->rs ---> ?1?, ?2?...
R:A:  {r}:{a}    word(1)  do,like
                 word(2)  like,it,music
                 word(3)  the,music,a,very
                 word(4)  lot,much,
AGO:  if do(1) then (2)=0
      if like(1) then (2)=it,music
      if it(2) or music(2) then  (3)=0,a,very
      if a(3) then (4)=lot
      if a(3) then (4)=much
MO:   Check each word.  Set decision flags.
      Evaluate all decision flags and come up with a
      reaction.
F:    A potentially high number of discrete reactions.
```

The above treatment of 3.44 is not complete. The number of branches and possibilities could be extended to any desired extent. No matter how complex the pseudo-parser, there would always be a "grey area" of incorrect responses that would slip through and of correct responses that would fail to be recognized. Still, a much larger number of correct answers and error-situations could be handled than would be possible with template#5, given even a very powerful authoring system.

While, in some cases, each drill item may bring its own item-specific data to the structure established in the pseudo-parser, it is highly unlikely that the rule structure itself would be item-specific, given the amount of time and effort that would be required to design, encode, test and elaborate it.

With authoring systems that could not handle the three drills above, with a template such as A-template#5, and which did not offer the possibility of encoding pseudo-parsers, the only option certain of handling all correct responses and providing adequate feedback for incorrect ones would be multiple-choice, whereby the response possibilities would be reduced to a very small set.

3.10.7.2 Simple insertions of Type B

Type B insertions tend to demonstrate very small implicit response-sets and a single correct response. 3.5(Dacanay) provides the typical example, using the basic template#2:

```
?:    ?->rt ---> ???
R:    {r}={do/does} <---data
+A:   {a}=do        <---data

MO:   1.   If ???=do then situation I
                     else 2.
      2.   If ???=does then situation II,
                     else situation III

F:    Situation I:    Learner's response is correct.
      Situation II:   Learner's response is incorrect.
      Situation III:  Learner's response is
                                    unrecognized.
```

3.35(Rivers), 3.41(Robinett), 3.140(Stack), 3.142(Stack), and 3.143(Stack) are all similar, though in some cases the response set is greater than two and so A-template#3 should be used. 3.34(Rivers) and 3.132(STack) differ only in that there is an answer set of two, requiring the use of a basic template such as A-template#5.

The only exercise that might deviate from this pattern is 3.117(Stack), where the learner must supply a verb form. This case, too, could be handled by the basic templates described above. An analysis by parts might be more appropriate than a holistic match, however, as the answer clearly has three functional components: auxiliary

+ stem + ending. To cover adequately all possible mistakes by use of a holistic analysis, the author would have to enter a lengthy list of possible variations. With a partial analysis, an adequate analysis could be provided, even without considering alternative possibilities for each component:

```
A-template#8

?:    ?->rs --->  ?1?,?2?,?3?
R:A: {r}/{a} =    part(1) = a
                  part(2) = command
                  Part(3) = é

M:   1.   if ?1? = a  then Situation Ia
                      else Situation IIa
     2.   If ?2? = command  then Situation Ib
                            else Situation IIb
     3.   If ?3? = e  then Situation Ic
                      else Situation IIc

F:   Situation Ia:  The auxiliary is correct
     Situation IIb:   The auxiliary is missing or
                      incorrect.
     Situation Ib:  The participle stem is correct
     Situation IIb: The participle stem is incorrect
     Situation Ic:  The participle ending is correct
     Situation IIc: The participle ending is incorrect.
```

Going the extra step of supplying a larger response-set for each of the separate components might allow the computer to generate feedback messages such as:

Form:	Message:
"est" ->	You cannot use "être" as an auxiliary in this case
"sav-" ->	The past participle has an irregular stem
"ée" ->	The ending does not make agreement with the subject

A-Template#8 assumes a response pre-processor capable of breaking up the learner's input into its meaningful parts.

3.10.7.3 Simple insertions of Type C

The answer-analysis requirements of the Type C insertions are essentially the same as for Type B. Thus 3.20(Rivers), 3.47(Robinett), 3.62(Cook), 3.64(Cook), and 3.106(Stack) could use either basic A-template#2 or #3. 3.63(Cook) and 3.114(Stack) could also use the same approach. The nature of their response-sets, however, would also allow for a more generative approach:

```
A-template#9   3.63(Cook)

?:    ?-->???
R:    {R} = {days of the week}  day(1)="Sunday", etc.
AGO:  1.   aaa=ccc+1.   Wednesday(4)=Tuesday(3)+1
A:    {a} =  Wednesday.
etc. as with template#3...
```

An authoring system that allowed for generation of items, both in presentation and answer-analysis, would free the author from having to encode the entire list of items in the data. As it could present items at random, each learner could be presented with a different set.

3.46(Robinett) is similar to 3.117(Stack) in that, while the response could be analyzed holistically, greater flexibility of response might be obtained with a partial analysis using A-template#8.

Examples 3.66(Cook) and 3.111(Stack) have a single answer, and no finite response-set and so would use basic A-template#1, the right/wrong analysis. If a multiple-choice approach were adopted for 3.111(Stack), at a level beyond:

a. True
b. False

(the multiple choice equivalent of A-template#1), then the author would be forced to supply some sort of response-set, presumably a meaningful one.

3.28(Rivers) resembles 3.144(Stack), discussed earlier, in that the learner has a certain element of choice in selecting the response and the choice made at one point determines the nature of the remainder of the correct response. In 3.144, for example, the choice of "do" as the first word in the response precludes the choice of "it" or "music" as the second word, while the choice of "like" as the first word forces their use as second word. 3.28 would require the same sort of analysis, but on a much simpler level:

```
A-template#10

?:   ?->rl=m,rt ---> [n],???
R:   {r}={aaa/bbb}    {her/to her}
AGO: if [1] then aaa=her
     if [2] then aaa=to her
A:   aaa = her/to her
etc.  as with template#2...
```

3.29(Rivers) and 3.65(Cook) represent cases that have not yet been examined. Both have very large response-sets and smaller, but still very large, answer-sets. It would be impossible for the computer to cover all possible responses. 3.65, for example, has a response-set that includes all imaginable proper names and an answer-set that includes all masculine names. Learner's could supply foreign names, like "Abdul" or strange spellings of English names, such as "Merija" for "Maria". A teacher could, nevertheless, distinguish these from common nouns such as "table" or nonesense combinations like "xxx". The computer is not as versatile as a human instructor. A response-set, {R}, consisting of a long list of valid names would have to be entered into a database. Each name could be tagged as to whether it was masculine, feminine, or both so that the answer for a particular item could be determined.

```
A-template#11

?:   ?->rt ---> ???
R:   {R} = {database: list of valid names}
A:   {a} = {those names that can be masculine}
MO:
1. Is ??? an element of {R}    if yes then 2.
                               else situation III
2. Is ??? an element of {a}    of yes then situation I
                               else situation II

F:   Situation I:  The response is correct.
     Situation II: The response is incorrect.
     Situation III:  The response is invalid.
```

The only other method of handling such an exercise would be to close the response possibilities by reducing the drill to a form of multiple-choice or by presenting a short of list of valid names on the screen.

3.10.7.4 Simple insertions of Type D

3.33(Rivers) is the only example of this type. In 3.9.2.4 it was shown that this exercise could be handled in two ways. If the instructional intention is limited to the ability of the learner to make the correlative change correctly, then there would be a single response with a dual response-set, and a single correct answer, clearly a case for A-template#2.

If the aim of the exercise, on the other hand, is to force the learner to identify the element to be deleted, as well as to make the resultant correlative change, then this is clearly a two-response situation and A-template#6 would normally be applied. It was pointed out in 3.9.2.4, however, that the basic fill-in presentation is not applicable to this example. Multiple-choice or free-response were shown to be available options. The multiple-choice approach could be handled by A-template#4. With free-response, the range of response possibilities widens, to include options that the designer of the activity may not have intended:

P: They haven't any coffee.
?: They have some coffee.
They've some coffee.
They've got some coffee.
They have some.
They do have some.

With such a lengthy input, holistic evaluation would be unsuitable. Fuzzy matching, to reduce errors due to spelling, could only be properly applied one word at a time. A template such as A-template#8 would be capable of handling each word separately. To allow for an even wider range of variations, a pseudo-parser such as in A-template#7 could be employed.

3.10.7.5 Simple insertions of Type E

It was anticipated in section 3.9.2.4 that 3.32(Rivers) would have to be performed as a multiple-choice as it exhibits no finite-response set and the answer-set is extremely large. In the multiple-choice form, the aspect of the activity that allowed the learner to provide original input would be lost. If this were a key goal of the activity, then it is unlikely that it could not be accomplished within the limits of an authoring system. The same finding would hold true for 3.36(Rivers), if only holistic answer processing were available. While there is a part of the answer that is fixed, there is another part which involves creative response by the learner. A partial- processing template such as A-template#7 could handle the answer-analysis, but a portion of the response would remain unanalyzed and would allow the passage of unreported errors. If this situation were not pedagogically acceptable to the author, then exercises similar to 3.36 would not be realisable through the use of an authoring system either.

3.10.7.6 Re-order exercises

The re-order exercises represent a special kind of multiple-choice. They all uniformly exhibit very small, explicit response-sets with a single correct response. The pre-processing of the learner's response will be different from that involved in a standard multiple-choice. With ?->rm, the computer must verify that ?a is an element of @{r}. With ?->ro, the computer must determine that all elements of @{r} are present, that none is repeated, and that there are no extra elements. The response-set, @{r}, represents not only the elements themselves, but all possible recombinations of the elements.

Whether or not the learner has supplied the correct order can be easily be established. In fact, there is no reason why more than one correct order could not be accommodated, although this is not evident in the data. The problem lies in what to do with responses involving incorrect orders. So long as the number of elements is small, the computer can identify each order using exact match techniques, and the author can provide a unique feedback for each one. In the two-element cases such as 3.8(Dacanay), 3.14(Dacanay) and 3.129(Stack), there are only two possible outcomes, the correct one, [2]-[1], and the incorrect one, [1]-[2]. Responding in terms of the incorrect order, in this case, amounts to doing nothing at all. Where there are three elements, as in 3.7(Dacanay) and 3.127(Stack), six possible outcomes present themselves. Four of these (excluding the correct order and the original order) represent actual errors that the learner might make. The number of combinations increases factorially, at an alarming rate:

```
4 elements = 4 factorial combinations = 1*2*3*4       =  24
5 elements = 5 factorial combinations = 1*2*3*4*5     = 120
6 elements = 6 factorial combinations = 1*2*3*4*5*6   = 720
```

Beyond three or four elements, a complete error anticipation type of analysis cease to be feasible. It would be necessary to identify certain key combinations which were typical errors, or some method of partial analysis would have to be devised.

3.129(Stack) may provide a possible solution. Here, only the words involved in the goal of the activity can be moved, while the others are held constant, thereby reducing the number of elements to be treated. This possibility is considered at greater length below in the discussion of complex re-ordering exercises.

3.10.7.7 Production of new elements

In the discussion of section 3.9.2.7, free-response and multiple-choice were offered as possible modes for production of new element exercises. The handling of multiple-choice needs no further discussion. The use of a multiple-choice approach may defeat the some of pedagogical goal of this category of activity, especially the intention of inducing the learner to "produce" a response. Only the free-reponse answer-analysis possibilities are therefore reviewed here.

3.39(Rivers) exhibits a response-set which is not finite, but an answer-set which is. The correct responses are short, so that a holistic matching technique should be feasible. A right/wrong analysis such as in A-template#1, but with the added ability to handle multiple correct answers could be applied to the exercise. There would be no manner of offering any discussion of incorrect choices.

The response-set could be artificially limited by displaying a selection of, say, 20 rejoinders and asking the learner to select only from those displayed. Explanations could afterwards be given as to the applicability of each of the rejoinders to the cue.

The effectiveness of partial-analysis, on the model of A-template#7 or A-template#8, is determined by the existence of one, or a set, of relatively fixed response models. In the case of 3.39, the number of different possible response models is enormous and therefore these approaches cannot easily be adopted.

3.54(Cook) and 3.145(STack), on the other hand, exhibit relatively small response- and answer- sets; with fixed response models. A-template#7 should handle these exercises very effectively. In 3.31A(Rivers), there is still a closed set of predictable syntactic patterns, although much larger. The same type of analysis could be used, but the answer-processor would require much more power and subtlety than the one assumed for template#7.

Exercise 3.38(Rivers) builds a database interactively with the learner and also applies it. During the building phase, the learner adopts a role similar to that of an author interacting with an authoring system. This is the phase where the learner can supply new material, and it is an unmediated phase. No analysis is made of the questions that are entered. As the computer is applying the database, asking questions such as "Does it live on a farm?", the learner's yes/no answers are still not being evaluated for correctness. It is only at the end of the tree when the computer asks "Is it a xxx?" that the learner's yes/no response can be mediated. At this point the computer knows either that it has the answer or that the tree needs another branch. This activity seems to be very complex, but from the point of view of answer-analysis, it is one of the simplest encountered in the data.

3.37 (Rivers) requires a partial analysis technique that has not been discussed yet, namely "key-word search". The number of possible ways to respond to the question "Why didn't they come here before midnight?" is enormous. The question does have a very small set of correct responses, however: "fireworks", "Fourth of July" If the computer looks only for these words, and ignores all other elements, then it is possible to determine, to some degree, if is the learner's response is correct, from the point of view of content:

A-template#12

```
?:   ?->rs   --->  ?1?,?2?,?3?,....?n?
R:   0
A:   {a} = {fireworks  and/or  Fourth of July}

MO:  1.  Is {a} element of {?n?}?

              If yes, then situation I
                  else situation II
F:   Situation I:   You appear to be correct.
     Situation II:  Your response does not seem to be
                    correct.
```

This technique involves controlled guesswork and this must be reflected in the reactions to the responses. Many teachers would reject the total lack of grammatical analysis. Sentences leading to situation I could include, for example, such unnaceptable formulations as:

They is not come cause them fireworks.

When the correction operation is strictly centred on a check of learner-comprehension, on the assumption that learners will produce grammatically correct responses, key-word search can still lead to ambiguous results, as with the following sentence leading to situation II:

They didn't come because they took part in the patriotic celebrations.

Nevertheless, for exercises such as this one, key-word search may be the only alternative to multiple choice.

3.10.8 Answer analysis options in the data: Complex drills

In section 3.9.3 the point was made that, while it may be possible to lessen the number of discrete steps the learner has to perform in completing a complex drill item, each point must nevertheless be analyzed separately, so as to be reflected in the feedback reactions. Holistic multiple-choice represents the only method of response elicitation method that would allow several discrete points to be analyzed with a single analysis per drill item.

Section 3.7 described how all complex drills could be analyzed in terms of the application of more than one of the simple drill-options defined. It is, then, to be expected that analysis of the responses would entail no more than the multiple application of some of the answer-analysis techniques described in the previous section for simple drills. No fundamentally new techniques should be required.

3.10.8.1 Multiple operations

Basic A-template#6 was defined in section 3.10.6 to handle the simple case of two simultaneous fill-in responses, as was proposed for the insertion + insertion exercises such as 3.12(Dacanay) and 3.22(Rivers). Each of the discrete insertions demonstrates very small response-sets and a single correct response and, as was mentioned during the discussion of presentation template in 3.9.3.1, represent simple insertions of Type B. The feedback reactions are additive, resulting in a dramatic increase in the feedback possibilities over those in a simple drill.

In 3.134(Stack), each discrete insertion, while still having only one correct response, has a larger response-set, and is made up of components that should be examined using a partial analysis. Each resembles the simple Type-B insertion shown in 3.117(Stack), which uses A-template#8. A-template#8 provides for a wider range of feedback reactions than the simpler templates, a range which would be doubled in the case of two discrete inputs. It would be reasonable to anticipate detailed feedback such as:

C: a. etudié b. rentré
M: Alain [@ quand je suis [].
?: est etudié rentrait

F: The first verb must be in the imperfect and the second in the passé composé.
Your passé composé uses the wrong auxiliary in this case
You have used the wrong verb ending on the imperfect

The insertion-then-insertion exercises, in which the insertion-slots are fixed, can be handled in the same way as 3.12 and 3.22. A-Template#6 may not be adequate, however, as the possibility arises that the first insertion will determine the second. Such is the case in 3.26(Rivers), where the choice of "Does", for the first insertion, requires "have" as the correct answer for the second, while the choice of "Has" for the first necessitates "got" in the second. While the two responses can be entered at the same time, the first must be analyzed, using a standard template like A-template#2, before consideration of the the second, the template for which must have an answer-generating capability as shown in A-template#10. The importance of the link between the two responses is not as evident in 3.136(Stack). Nevertheless, the answer-analysis mechanism should be given the subtlety to realize that "politely" is only correct in slot#2 if the learner has followed instructions and inserted "speaks" in slot#1. Mindless feedback such as the following can then be avoided:

P: The boy is polite
C: How does he speak?
M: The boy ---- -----.
?: is politely
F: "is" is wrong but "politely" is correct

Where the first insertion can be made in a variable slot, there would be no alternative, in the fill-in mode, to presenting the two operations in sequence as a simple Type-A insertion, followed by a simple Type-B insertion. Errors in the first part would have to be be processed before proceeding, as the display of the response model for the second insertion would have to show the cue inserted in the correct slot. Leaving it in the wrong slot would result in confusion on the part of the learner and would, in any case, be possible only if the second response-model were generated. 3.24(Rivers) illustrates the situation:

```
     P:    She brings too many pencils to school.
     C:    paper
     M:    She brings too many pencils to school.
           [1]                     [2]        [3]
     ?:    [1]
   * C:+M: Paper ---- too many pencils to school.
     ?:         brings
     F:    Correct!
```

Free-response would allow for the learner to complete the activity in a single step and in a much more natural fashion. As the implicit response-model is relatively rigid, a template such as A-template#8 could handle the response analysis on a word-by-word basis. In the case of the above response:

Paper brings too many pencils to school.

the feedback could be:

1. "Paper" cannot substitute for "She".
2. "pencils" should be replaced by "paper"
3. ...which would require "too much" instead of "too many"

In 3.9.3.1 it was suggested that the re-order-then re-order exercise, exemplified by 3.15(Dacanay) could be handled in one step by treating it as an ordering of six discrete word-units rather than two separate orderings, first of two sentence-units, and secondly of two word-units. As was indicated in 3.10.7.6, only some of the 6-factorial possible combinations could be treated as errors, the rest falling into the "not recognized" category.

3.10.8.2 Mixed operations

The insertion+removal exercises, 3.13(Dacanay), 3.16(Dacanay), 3.30(Rivers), and 3.131(STack), all exhibit fairly predictable response-models. Free-response, using a template such as A-template#8, joins holistic multiple-choice as being an option which allows the learner to accomplish the task in a single operation. Other approaches would require two discrete actions on the part of the learner, one for the insertion and one for the removal. 3.9.3.2 showed that the removal action could be accomplished as a two-answer (first and last word of sequence to be removed) or multiple-answer (all the words to be removed) fill-in or multiple choice operation. Variations of basic A-templates#4 and #6 would handle the response analysis.

Removal exercises illustrate a case where the elicitation of a whole series of responses may be feasible. In the case of 3.13(Dacanay), where up to seven discrete responses might be required, the additive nature of the feedback that was discussed in relation to template#6 is not a problem. For each word in the model there are only two possibilities, it is to be removed or it is to be left in. Considering all the words in the model of 3.13, there would be only 28 possible messages.

3.44(Robinett) and 3.45(Robinett) are naturally done as consecutive simple drills. The re-order portion could be handled as a re-order multiple-choice or, given the small number of elements and the fixed response model, as a free response using A-template#8. Once all reaction and corrections to the first part were completed, the second part could easily be handled by A-template#7 as a free-response.

3.10.8.3 Choice of operations

No new answer-analysis mechanisms are required for these essentially simple drills, other than the ability of the authoring system, as was earlier discussed with reference to presentation templates, to switch between templates.

3.11 Factors influencing branching between drill items

The flexibility that the author can allow to the computer, in its choice of the next drill item to present to the learner, is a function of the authoring system's available presentation and answer-analysis templates. Before examining the effect of these on branching, some consideration is necessary of the different branching options that are available on the computer.

3.11.1 Branching options in CALL activities

Eisele (1978, p.17) defines "branching" as a method of decision as to the sequence of component segments and outlines four basis branching schemes: Linear branching, looping, remedial branching, and voluntary branching.

A. Linear branching occurs when the learner must continue on to the next item, regardless of the outcome of the current item:

Item 1 ---> Item 2 ---> Item 3

Some further qualification of this category is necessary. The situation in which the computer simply proceeds from one item in the data to the next can be considered as "fixed branching", while the situation in which the computer randomly selects items from the data, or generate items according to a rule, can be considered as "random branching."

B. Looping occurs when the learner is made to repeat an item when immediately after having made an error. Under the present scheme, the term "branching" is restricted to links between items and so the repetition of an item must be considered as one of the immediate feedback options possible WITHIN the item. The "looping" that Eisele discusses would be considered as a special case of branching scheme A.

```
[ Item 1  y  ---->] ---> [ Item 2 ...
  <------ n
```

C. In Remedial branching the next item is chosen on the basis of the learner's performance in the current item: This is the interpretation most frequently applied, in CALL, to the term "branching". Several patterns may be identified:

c1: Remedial branching based on one item, where, if the learner misses an item in the main stream, the next item to be performed is a remedial item of the same sort as the one missed. There may be one of these, or the chain of remedial items may continue over a number of presentations. Once "mastery" of the point is demonstrated, through providing a correct response to one or more of the remedial items, the learner returns to the main stream. Those learners who do not make an error are not diverted from the main stream and receive no presentation of the remedial items, although some authors may prefer to offer them the option of proceeding through the remedial exercises in any case.

```
[Item 1]  n ---> [Item 1b] n ---> [Item 1c] n --->etc.
   :                   y:                y:
[Item 2]  <-------------------------------
```

c2: Remedial branching, based on several items, is similar, except that more than one error is required to trigger the remedial stream. A count must be kept of the number of errors that the learner is making.

```
[Item 1] n ---> x=x+1
   :               :
   : <-------------
   :
[Item 2] n ---> x=x+1
   :               :
   : <-------------
   :
[Item 3} n ---> x=x+1: if x=3 then ---> [Item R1]...
   :               :
```

c3: Skipping forward is a variation on c2. A count is kept of correct answers, and the student is branched out of a sequence (or given the option to leave) after a specified number of correct answers. The same process may be in the case of incorrect answers.

```
[Item 1]
   :
   : if y then y=y+1
   :
[Item 2]
   :
   : if y then y=y+1
   :
[Item 3]
   :
   : if y then y=y+1: if y=3 then EXIT
   :
[Item 4]
```

c4: <u>Remedial branching based on multiple factors</u> is a more complex case of C1, where the correct answer still moves the student on in the sequence, but there is more than one possible avenue of remediation, depending on the nature of the learner's errors. Where more than one error is possible at the same time, a hierarchy may be established so that the learner does not have to perform all of the indicated remedial items.

```
[Item 1]
  :y          n --->a.  ---> [Item A] ...    :
  :                 b.  ---> [Item B] ...    :
  :                 c.  ---> [Item C] ...    :
  :                                          :
  : <----------------------------------------
[Item 2]
```

c5: <u>Remedial Branching based on multiple factors across more than one item</u> is a combination of c3 & c4.

```
[Item 1]
 :y      n ---> a.----> a=a+1 ---->:
 :              b.----> b=b+1 ---->:
 :              c.----> c=c+1 ---->:
 :                                 :
 :<--------------------------------
[Item 2]
 :y      n ---> a.----> a=a+1 ---->:
 :              b.----> b=b+1 ---->:
 :              c.----> c=c+1 ---->:
 :                                 :
 :<--------------------------------
[Item 3]
 :y      n ---> a.----> a=a+1 ---->if a=3 then ...
 :              b.----> b=b+1 ---->if b=3 then ...
 :              c.----> c=c+1 ---->if c=3 then ...
 :                                 :
 :<--------------------------------
[Item 4]
```

c6: <u>Repeating incorrect items at a later point</u> is not identiacal in nature to "looping", which is not considered here as a true form of branching, because in this case, the program will have already moved past the item in question and must now return to it. This step is usually accomplished by flagging items as having been done correctly or not.

c7: <u>Undirected branching</u>, where individual items do not, necessarily have "correct" answers. Each item leads to multiple paths, which lead on, in turn, to multiple paths. Eventually, certain paths lead to correct conclusions and it is in this way that the activity is mediated.

```
                         [Item 1]
                            :
    :-----------------------:-----------------------:
    :a                      :b                      :c
    :                       :                       :
 [Item A]                [Item B]                [Item C]
    :                       :                       :
 :-------:               :-------:               :-------:
 :       :               :       :               :       :
[Ai]   [Aii]           [Bi]    [Bii]           [Ci]    [Cii]
```

This is a tree-structure, such as one might be found, for example, in a game of "Twenty Questions". It is also possible that branches of the tree be cross-linked and back-linked, as in an "Adventure"-type simulation.

D. <u>Voluntary branching</u> opportunities are offered in many CALL activities. These may include options to 1) skip an item, 2) skip to the beginning of the next series, 3) re-do an item that was not completly understood, 4) or exit completly.

Voluntary branching is certainly important to the learning process. The computer should be responsive to the needs and desires of the learners, and should not unduly restrict their flexibility. With a book, for example, the learner always has the option of turning the page forward or backward and of going to the next or previous chapter. A computer-based activity should offer, at least, the same flexibility. This type of branching does not derive from the computer's mediation of the activity, however, and therefore must be considered a "user-friendliness" variable rather than a topic for to be considered in the current investigation.

Computer-invoked branching (or branching decisions made by the computer on the basis of the learner's activities) may also be entirely voluntary, of course. The computer may ask, for example:

> Your progress indicates that you should do some remedial activity. Would you like to do that or continue ?

Of Eisele's four categories, then, only two really are of concern here, namely Type A and Type C. The three possible types of branching to be considered are: "fixed branching", "random branching" and "variable branching". The latter term refers to the schemes outlined under Type C, which are neither fixed nor random.

3.11.2 The Effects of presentation and answer-analysis factors on branching

In the discussion of P-templates for interlocked drills, it was shown that, without the ability to generate presentation elements, the order of the items would be fixed by the data. Random branching would be impossible and variable branching would quickly become very difficult, as the author tried to predict and provide for all possible branchings in the data. The choice between interlocked vs. independent drill items appears to be the only formal presentation factor that directly affects branching.

Branching possibilities are much more directly affected by the sublety of the answer-analysis templates. Remedial branching is only possible to the extent that the computer can identify the learner's errors.

The remedial branching possibilities of an authoring system which only offered right/wrong analysis would be restricted mostly to simply repeating incorrect items (branching scheme: c6). Branching to a stream of remedial exercises (c1) could only be done where where the response possibilities were limited to dual response-set. Otherwise the author could not ensure that the error committed by the learner matched the remedial meterial offered. Even with only right/wrong analysis, one can provide for skipping forward based on correct or incorrect responses (c3), albeit with less certainty in the case of incorrect answers. Tree structure branching (c7) can also be handled with a right/wrong (in this case Yes/no or A/B) answer-analysis. Such tree-structure activities are often more interesting to the learner, however, where there is more than one branch from each node.

Where the authoring system allows for multiple evaluations, it is possible to take advantage of branching schemes based on remediation of particular errors (c1 & c4). There is a direct link between the subtlety of answer-analysis and branching flexibility. The greater the number of possible incorrect incorrect responses that can be identified, the greater the possibilities for branching. It is possible, and in fact quite probable, however, that the answer analysis mechanism be more subtle, be able to identify far more response variations, than the system allows branching paths.

3.11.3 Independent factors affecting branching

Branching subtype C2 and its derivatives, c3 and c5, amount, in effect, to building up a model of the learner's performance and taking action based upon that model. To be capable of building such a model, the authoring system must have some facility for keeping records over more than one drill item. Most, in fact actually, do offer this facility. It is often manifest as a score-keeping mechanism, although score-keeping is only part of the necessary resource. The system must be capable, not only of presenting the score to the learner, but also of allowing the author to cause action to be taken, based upon the score.

The number of correct or incorrect responses, and its derivatives, such as percentage correct over a certain number of items or the learner's percentile standing, can indicate only the crudest model of performance. In order to frame an accurate gauge of linguistic performance, many more intangible factors must be assessed. An authoring system must be capable of providing a wide range of different types of scores. As in the case of branching, however, the subleties of score-keeping are directly related to those of the available answer-analysis mechanisms.

3.12 Categorization by pedagogical factors

It was shown in section 3.5 that most authorities categorized activities according to a second gradient which reflects pedagogical, rather than formal, features. Almost half of the data represent activities presented to exemplify these pedagogically based typologies (see table 3.3 for list).

There is less agreement among authorities as regards pedagogical classifications than was the case for the operational categories discussed in section 3.6. Table 3.14 lists the authors and the variety of approaches, categories, and terms employed to differentiate activities along this gradient. Even where the same or like terms are used, they are presented often with differing connotations. As a general pattern, "meaningful/ contextualised" activities are universally preferred over "mechanical/ manipulative/ meaningless/ non-contextualised/ non-communicative" activities". The latter are condemned as useless or even counter-productive to the learning process. There is much less agreement as to what the term "meaningful activity" actually represents. The example cited by one author as a "meaningful" drill matches another author's example of a "meaningless" one.

Where more than two categories are postulated, "Communicative/ situational" activities are proposed as being even better than simple "meaningful/ contextualised" ones. Table 3.15 shows that uniform criteria for separating these two groups are even more difficult to find, than for distinguishing "meaningful" drills from "meaningless" ones. Each author uses a completely different scheme for sub-classifying activities in the "communicative" sphere.

It is difficult to isolate, from the categorization schemes exemplified in tables 3.14 and 3.15, a uniform set of factors by which to evaluate drills pedagogically. Nevertheless, for the teacher, this pedagogical classification is in all probability the most important. Teachers' questions about authoring systems most often focus on their ability to create activities of the "meaningful" or "communicative" type.

Such questions cannot be addressed without having concrete criteria by which to judge an activity as being "meaningful" or "communicative" rather than "meaningless". These criteria should reflect pedagogical realities, but must also relate to possible implications, in terms of the use of an authoring system to produce computer-based activities of a similar nature. Once the factors contributing to an activity's "meaningfulness" are isolated, it should be possible to examine each factor individually.

Nine such possible factors have been postulated and are discussed below. They are based only in part on points explicitly raised by the authors in setting forth their typologies. The primary basis for extraction of the nine factors was an initial examination of the authors' supporting example drills, in which their implicit assumptions about pedagogical effectiveness become evident:

I. Realistic interchange: The stimulus-response pairs represent sequences that might be produced during normal language interaction (Stack, 1971, pp. 135, 137-138; Dakin, 1973, p. 62). Also included as positive were those activities in which learner is working closely with realistic interchanges, even though the actual activity does not simulate real language interchange.

II. Extended context: Subsequent items are presented within a global context, or contribute to building up such a context (Dakin, 1973, p. 85). A secondary factor, perhaps the more important, is whether the context plays any part in determining correct responses.

III. Truth value: Responses are judged in relation to a given context or domain on non-grammatical, non-lexical grounds, such that the learner cannot produce a correct response without an understanding of the context (Stack, 1971, p. 134; Dakin, 1971, p. 8-9; Robinett, examples 3.48-3.50; Paulston, 1972, p. 134). Possible complications arise out of the relationship of this factor to II (Extended Context) and VII (Lexical Understanding). The former becomes clear as the factors are applied to the data. The latter must be defined: Truth Value represents more than simply responding correctly to a lexical cue, such as a picture. It consitututes, rather, responding correctly to a given situation.

IV. Information gap: There is information that the learner does not have, but the interlocutor does. This factor was suspected of being related, in some way, to factor III (Truth Value).

V. Function practice: There is focus on the functional aspect of the language being used, as opposed to the grammatical or lexical aspect. The immediate complication in defining this factor was that all realistic interchanges (I.) would clearly expose the learner to some illustration of the functional use of language. As defined here, the isolation of one functional aspect must be made much more explicit.

VI. More than one response: More than one correct response to the stimulus is possible (Paulston, 1972, p. 134; Dakin, 1973, p. 85). Extremely narrow cases, such as close synonyms or variants like "plane" and "airplane" are not considered.

VII. Lexical understanding: The understanding of the meaning of a word, phrase, or idiom is required to produce a correct response (Dakin, 1973, p. 8-9). In its narrow sense, this can be defined as a case where an exercise cannot be completed if the text is replaced by nonsense words (Paulston, 1972, p. 135). This criterion alone ceases to be productive with more advanced drill-types, where a general understanding of the language is always required. In such cases, "lexical understanding" is considered a factor only where there is focus on the meaning of a word or expression.

VIII. <u>Elicitation of new, creative elements, within constraints</u>: New, creative semantic elements (not present in the overall context or in the stimulus) must be provided by the learner. The learner's choice is constrained, to some degree, however, by a given component of the activity. Completely free creation is, therefore, not possible. To eliminate confusion with factor VII (Lexical Understanding), this process must involve more than simply the definition or re-phrasing of a word or expression.

IX. <u>Elicitation of new, creative elements, without constraint</u>: There is opportunity for the learner to include original (i.e. personally-chosen) semantic elements in the response, without constraining factors. Obviously, this must only refer to creative additions to an otherwise constrained response. Completely free expression would fall outside our definition of CALL activity.

The list of these nine factors is not exhaustive. Other important factors are often cited in the context of communicative drills, such as motivation and relevance of the situation to the learner. In the case of Robinett (examples 3.49-3.50), for example, the relevance of the situation to the learner is the sole deciding factor between a meaningful and a communicative drill. The decision was taken to include only those factors which could have a forseeable effect on the computerisation of the drills examined.

Table 3.16 examines the relevant data in terms of the nine discrete factors postulated. Activities were then grouped according to individual factors (Tables 3.21 through 3.28) to observe any possible inter-relationships between the factors.

Table 3.21 shows all the cases in which there is a realistic interchange. Many of the cases (75%) also exhibit an extended context. Originally, the expectation was that all examples where the stimulus-response pair were of a question-answer nature would automatically be realistic interchanges. Those examples representing a question-answer interchange are listed in table 3.20, and this phenomenon is shown to be generally true. Beile, however, shows with example 3.156, that it is possible to create a question-answer exchange that does not represent a realisitic interchange.

Table 3.22 shows all the cases in the data in which there is an extended context and is further subdivided, according to the role of the context in determining the correct answer. Many of the cases (70%)

also exhibit a realistic interchange. It seems clear that extended context and realistic interchange tend to be associated, but still independent factors. The fact that 66% of the cases where context directly affects the correct response also involve the truth-value factor is not surprising, as these two factors would seem, in the nature of things, to go together. How, then, can the 33% of cases providing the exception be accounted for? 3.71(Byrne) and 3.73.1(Byrne) exhibit an information gap, which, as is suggested below, might be considered an alternative case of Truth Value. The others call on the learner's lexical skill, which differs from Truth Value only according to the arbitrary definition given above.

Table 3.21 lists those examples which involve Truth Value or Information Gap and separates them as to whether the role of the learner is to answer a question (Question-Answer), pose a question (Answer-Feedback), or neither (no question). Truth Value and Information Gap and listed together because it was felt that the difference between the two factors was simply a function of the direction of information flow (from the learner or to the learner). It is difficult to postulate a general rule based on the small number of cases present in the data, but the trend seems to indicate that Information Gap and Truth

Value can be treated as different manifestations of the same basic factor.

What of the possible relationship between Truth Value/Information Gap, on the one hand, and the presence of an extended context affecting the choice of correct response, on the other? Those cases which exhibited the latter factor, but did not exhibit Truth Value were discussed above, and were shown not to be significant. Truth Value, without the presence of a corresponding extended context affecting the choice of correct response, is exhibited by 18% (3/16) of the examples listed in table 3.23. The three exceptional (3.69(Byrne), 3.95(Dakin), and 3.96(Dakin)) cases exhibit item-specific rather than extended contexts. It can be assumed, then, that where an extended context affects the choice of correct answer, Truth Value or Information Gap will also be factors, but not necesarily the reverse.

Table 3.24 presents the few cases in the data where there is explicit practice of language functions. In all cases, there is also a realistic interchange. In nearly all cases, there is also an extented context, but the one exception,3.56(Cook), indicates that this might not always be the case.

Table 3.25 examines all the cases in which there is more than one response. Cases can be imagined where a purely morpho-syntactic variation would lead to the possibility of multiple correct responses. In all the cases exhibited in the sample, however, some combination of factors VII (Lexical skill), VIII (Elicitation of new, creative elements, within constraints), and IX (Elicitation of new elements, without constraints) is also evident, indicating that the variation in correct answers must be of a semantic nature.

Table 3.26 groups all the examples where Lexical Understanding is a factor in determining the correct response. In 63% of the cases, more than one correct response must also be accepted. A possible significance of this trend is explored, below, in the discussion of the difference between "meaningful" and "meaningless" activities.

Table 3.27 shows all examples where creative new semantic elements are added, within constraints. As might be expected, all such cases lead to the possibility of more than one response. With only one exception, they also require lexical skill. Table 3.28 shows those examples where some degree of free creative input on the part of the learner was permitted. Again, all must allow for more than one correct response.

It is necessary to postulate some manner of distinguishing between those activities characterized by their authors as being "mechanical", "manipulative", "meaningless", "non-contextualised", or "non-communicative" (hereafter referred to simply as "meaningless" activities) and those characterized as being pedagogically more appropriate (hereafter referred to simply as "meaningful" activities). (The difference between "meaningful" and "communicative" activities is not important in this context.) In some cases, authors are in direct conflict. Stevick (example 3.93) classes a drill which demonstrates only a realistic interchange as a "meaningless" drill while Stack (1971, p. 134) explicitly defines a "meaningful" drill as one that has, at least, a realistic interchange. For Robinett (1978, p. 208), a meaningless drill becomes meaningful when there is reference to truth value. All other factors are incidental.

It was decided to explore how the nine factors postulated, which had their ultimate basis in the authors' contrasts of "meaningless" and "meaningful" activities, might contribute to a common, working understanding of the difference between the two types. All the drills identified as being of the "meaningless" variety are grouped together in table 3.19. It is clear that in all but three cases, none of the nine factors considered are involved. In the three exceptional cases only one factor is evident, either realistic interchange or lexical skill. In table 3.18 those examples characterized as "meaningful", but showing the influence of only a single factor, are shown. The list is quite short, as only four such examples are found in the data. Interestingly, the single factors are the same as for "meaningless" drills, realistic interchange and lexical skill.

Ninety-one percent of the examples in the data characterized by their authors as being "meaningful", exhibit the presence of at least two of the pedagogical factors. Realistic Interchange, alone, is evident in both "meaningless" and "meaningful" examples. It was shown above that in nearly all cases where there was Realistic Interchange one could also expect to find Extended Context and in many cases where Lexical Skill was involved one could expect more than one response to be possible. Beile alone seems to feel that the basic difference between "meaningless" and "meaningful" is the presence of a realistic interchange (examples 3.150 vs. 3.151-4). For Paulston and Stevick, the presence of a realistic interchange is not sufficient to make a drill meaningful (examples 3.51/52 vs. 3.53 and 3.93 vs. 3.94). 3.94).

It is also interesting to note that the drill examples supplied by those authors who postulate a single, deciding factor, actually evidence the presence of multiple factors (Stack, examples 3.160 vs 3.161; Robinett, examples 3.48 vs. 3.49).

It seems safe to say, then, that "meaningful" activities should exhibit at least two of the nine pedagogical factor postulated above. Should an activity exhibit only a realistic interchange, or the need for lexical skill, it is possible to rule that it does not qualify as a "meaningful" activity. There are no examples in the data of any the other factors in isolation and so it is impossible to speculate whether any of them, on their own, can render an activity "meaningful".

3.13 The relationship between pedagogical and operational factors

The operational classification developed in section 3.7 was applied to the pedagogical sample to see if there were any discernible relationships. The results are summarized in table 3.17b. The presentation and answer-analysis factors, developed and explored in 3.9 and 3.10, were also applied to the sample. A detailed analysis of this application is shown in table 3.17, and summarized in table 3.17c. Tables 3.20b through 3.28b consider the operational factors in relation to discrete pedagogical factors.

The breakdown of the pedagogical sample in terms of operational categories is shows similar results to that of the original operational sample upon which the categories are based. Most of the activities (82%) represent simple operations and, of those, most (71%) could be classified as insertions. All the complex operations in evidence can be described in terms of the simple operations. It was not necessary to postulate any new opertational types in order to accommodate the drills in this sample. There are, however, operational types which are not in evidence in the pedagogical sample. There are no deletion or removal operations and only one isolated case of a re-order operation. Nor are there any mixed operations, though this is understandable as most of the mixed operations in the operational sample involved removal.

A factor-by-factor examination (tables 3.20b-3.28b) revealed no evident pattern. Drills illustrating a particular factor seem to be evenly spread across all operational types. The only exception seems to be in the case of drills having more than one response, in which case no insertions of Type A are represented.

Table 3.17c illustrates the presentation and response-analysis factors as they apply to those drills which can be called "meaningful" by the criteria discussed in section 3.12. Some interesting global observations can be made. It might be anticipated that meaningful activities would require item-specific response models (mmm) because universal response models (MMM) would make the activities excessively rigid and mechanical. In fact, universal response-models could be set up for 60% (21/35) of those exercises with explicit response-models. One might expect meaningful activities to exhibit large or infinite response-sets, while meaningless activities exhibited small, closed response sets. In fact, 44% (20/45) of the examples exhibit response sets which could be termed "very small" (vs) and 28% (13/45) more exhibit response sets which are still small enough to be manageable using standard answer-analysis techniques. Finally, it might be expected that meaningful drills would require the acceptance of a large number of correct responses.

Thirty-three percent (15/45) if the activities only allow for one correct answer and an additional 40% (18/45) have answer sets that could be characterized as "very small".

Table 3.20b examines the operational factors for those drills which are question-answer interchanges. As might be expected, all, including the type B insertions (C:+M:) have separate, explicit cues. These all have very small or small response sets and single or very small answer-sets.

Table 3.21b lists the drills which exhibit a realistic interchange. No particular presentation or answer-analysis pattern is observable. Answer-sets, for example, range from a single correct response to an infinite set of correct responses. A similar case can be found in table 3.22b (extended context), 3.23b(truth value/information gap), 3.24b(function practice), 3.25b(more than one response), 3.26b(lexical skill), 3.27b(creative input, within constraints). In 3.28b(creative input, without constraints), there are no very small answer-sets, which is, of course, predictable.

Those drills with very large or infinite response sets or answer sets would require modification, before presentation on the computer, so as to limit the range of response possibilities in some way. This could be accomplished in a natural fashion, as in 3.96(Dakin) where, if the domain were restricted to musical instruments by contextualising the activity more fully, very small response- and answer-sets would result. Where natural limitations cannot be introduced, it is necessary to resort to multiple choice. Indeed, Candlin has already taken such a step in 3.76 and 3.77, where the large implicit answer set has been artificially reduced by providing a very small explicit response set with one correct answer.

Drills with large or infinite answer-sets that could not be delivered well as multiple choice activities would be difficult to accomplish on the computer without a full parsing-mechanism. An example of this case is provided by 3.89(Candlin). Such an activity borders on the production stage of language learning. Though structured and mediated, it is not inherently interactive.

Chapter Four: Conclusions and Implications

Three questions were raised at the beginning of this thesis: 1) whether a model of the operational features of language learning activities could be constructed that would predict authoring system requirements for producing these activities as CALL exercises, 2) what the range of activities that it would be possible to produce through the use of an authoring system might be, and 3) whether the features which make activities meaningful and communicative are affected by the flexibility of the authoring system used to realise them.

Three separate purposes of the model were proposed. It was to serve a) as a partial guide to teachers in choosing authoring systems, b) to assist teachers in determining how a particular activity might be realised through the use of an authoring system, and c) to give designers of authoring systems an indication of the type of flexibility that is important to CALL.

4.1 Examination of the questions raised, in the light of the study

In section 2.2, it was shown that an authoring system functions by establishing a template, or formula, for an activity and then soliciting the content data required for each discrete item. The formal feature model involves the categorization of exercises according to the behaviour of the cue and the response-model, outlined in section 3.7, and the factors influencing the presentation, answer analysis, and branching phases of the exercise, considered in sections 3.9, 3.10, and 3.11. The model permits the identification of the inherent formal features of an activity and the translation of these into a formulaic description. Where the model can reduce an activity to a formula, therefore, that activity should be theoretically realisable through the use of an authoring system. The question of whether existing authoring systems provide the degree of flexibility required by the formula obtained from the model's application is excluded from the present treatement.

The model presented here has been tentatively validated in two ways. First, some measure of internal consistency is demonstrated as the categorization established in section 3.7 reflects regularities in the presentation and answer-analysis phases discussed in sections 3.9 and 3.10. Second, in section 3.13, the model was successfully applied to the drill sample organized pedagogically. All of these latter drills could be accommodated according to the model.

A further process of validation, not conducted in this study, would be to examine existing computer-based activities, to determine whether they could all be adequately described in terms of the model, or whether there existed CALL activities that are not predicted by the model.

In section 2.2, CALL was defined as consisting of structured, mediated, interactive, practice activities involving the use of computers. In section 3.1, the role of CALL in the practice, rather than the presentation or production stages of language learning was re-iterated.

The model can be used to predict whether a given activity is of a suitable form to be converted into a CALL exercise. For the model to be applicable at all, however, the activity must be a structured one. An unstructured activity does not submit to any formulaic description. The activity must be mediated, as well. There must be, at least, the possibility of two divergent paths. In the example of a presentation activity as shown in section 3.6, the answer-set would be identical to the response-set. In other words, all valid responses would be correct. Another prerequisite to computer-mediation of an activity is the author's ability to provide an indication of how the correct response might be identified. Candlin's example 3.89, discussed in section 3.13, exhibits a very-large-to-infinite answer-set and, therefore, must be considered a production activity.

The relationship between the model and the drills selected as exemplifying meaningful and communicative practice was explored in section 3.13. All of the drills could be described in terms of the model, indicating that no new operational formulae are required for this second sample. The general pattern, as well as the analysis by discrete pedagogical features, suggests that meaningful and communicative exercises, as illustrated in the sample, are not _operationally_ different from meaningless, manipulative exercises.

In the discussion of quality materials in section 2.3, the question is raised as to the degree to which quality depends, on the one hand, on the background and expertise of the author and, on the other hand, on the flexibility of the software. Wyatt is quoted as indicating that authoring systems would fail to provide the quality desired because of their lack of flexibility. It appears from the present study that the sophistication of authoring systems, in general, should not have any appreciable negative effect on the production of quality courseware. (Individual differences in authoring systems will always affect the ability to realise particular types of exercises, however, as is pointed out below.)

The results would lead one to place the onus for the creation of quality, meaningful and communicative courseware on the shoulders of teachers and teacher-trainers, rather than on the shoulders of software developers.

4.2 The implications of the model to the purposes of the study

Teachers are normally cognizant of the pedagogical aspects inherent in the types of activities they wish to produce. Before they begin to examine authoring systems, they need to become cognizant of the operational aspects as well. Similarly, where an authoring system has already been selected, teachers must compare the operational features of an activity with the options available on the authoring system chosen in order to determine if the activity can be produced in an acceptable manner.

Chapter Four Conclusions and Implications

The study illustrates how slight changes to an activity can transform a simple operation into a complex one, or completely change the operational type. It was shown, for example, in section 3.13, how 3.96(Dakin) could be radically transformed simply by adding a few contextualising words. A seemingly straight-forward drill such as 3.29(Rivers), was shown in section 3.9, to involve many discrete factors: variable response model, elicitation of multiple responses, linked responses, and a large answer-set.

It was shown in sections 3.9 and 3.10 how even quite complex activities could be modified to fit the multiple-choice approach, allowing them to be realized on even the simplest of authoring systems. Understandably, teachers will demand alternatives to multiple choice. It is not always appropriate and its overuse would certainly lead to a negative reaction on the part of the learners. The second solution to handling complex activitities or those with large answer sets was shown to be the construction of a pseudo-parser.

Currently pseudo-parsers must be programmed from scratch. The fact that they can be described formulaicly indicates that they could be included in the repertoire of "tools" that an authoring system should provide. Their construction might be possible by the use of high level "verbs" discussed in section 2.2. Programmers would have to take care, however, not to make the resulting authoring system too difficult to use. As was proposed in section 2.2, various levels of interaction with the authoring system might be provided to teachers. The beginner could be content to choose from a variety of "matching" paradigms, while the more advanced user could create pseudo-parsers.

The need to generate item-specific stimuli, rather than read them from data, was shown in several cases, such as when the stimulus depends, in part, on material that the learner has previously provided. Similarly, there are cases where the correct response must be determined by the computer on the basis of choices that the learner has previously made. These features must be made available on authoring systems.

An answer-analysis mechanism capable of distinguishing a large number of discrete errors that the learner might make, coupled with a record-keeping mechanism that can keep track of a large number of different factors, was shown to be a necessary pre-requisite to allowing the computer to build up an adequate model of the learner's performance. Authoring systems must be given this flexibility.

4.3 The limitations of the study and directions for further research

The limitation of the study to the more or less traditional applications that are currently possible on the computer was necessary in order to construct a feasible model. As the intelligence and linguistic ability of computers is developed, there will be a corresponding decrease in the need for such a rigid, algorithmic structuring of activities. The model may have to be changed to allow for a wider range of exercise types, as the creation of presentation and production activities with an authoring system become possible. Truly intelligent authoring systems of the future may contain a model such as the one developed here, and free the author from having to consider all of these operational factors. A point may be reached in the not-so-distant future where authors can simply communicate to the authoring system the type of exercise they wish to accomplish and have it respond, "Okay, done." or "Sorry, cannot do."

In order to keep the present study within the bounds of a reasonable length, no attempt was made to embark on a review of existing authoring systems, to establish their capabilities and limitations in the creation of different types of exercises. Such an investigation would be the next logical step. The present theoretical model could then be further validated, and possibly adjusted to fit realities that are not foreseen at this point. While the model itself can be of some use to teachers in selecting an authoring system, an actual survey of existing systems, based on the model, would be much more readily applicable.

References

Allen, J. R. Individualizing foreign language instruction at Dartmouth. Foreign Language Annals, 1972, 5(3), 348-349.

Alatis, J. The application of instructional technology to language learning. CALICO Journal, 1983, 1(1), 9-12.

Ashmore, T. N. The Future of the microcomputer in the speech communication classroom. Paper presented at the annual meeting of the Western Speech Communication Association, Albuquerque, N.M., February 19-22, 1983. ERIC Document #ED228669.

Avner, R. A. Cost-effective applications of computer-based education. Educational Technology, 1978, 18(4), 20-25.

Bagley, C. A. CAI vs. CBE Languages: Authoring, how soon?. Minneapolis: Minnesota Educational Computing Consortium (MECC), 1974. ERIC Document #ED149713.

Barker, P. G., & Singh, R. Author languages for computer based learning. British Journal of Educational Technology, 1982, 13(3), 167-196.

Barker, P. G., & Singh, R. A practical introduction to authoring for computer assisted instruction, part 2: PILOT. British Journal of Educational Technology, 1983, 14(3), 174-200.

Beile, W. Typologie von Ubungen im Sprachlabor. Frankfurt: Diesterweg, 1979.

Bloch, J. Prewriting and composing on the computer. TESOL Newsletter: Special Insert on CALL, 1986, 20(1), p.4

Boyd, G. The appropriate level of sophistication of computer languages for the writing of tutorial modules and courses. Aspects of Educational Technology, 1971, 5, 449-455.

References

Boyd, G. Four ways of providing computer assisted learning and their probable impacts. Computers in Education, 1982, 6, 305-310.

Boyd, G., Keller, A., & Kenner, R. Remedial and second language English teaching using computer-assisted instruction. Computers & Education, 1982, 6, 105-112.

Braun, P. H., Reflection on CAI language design. Paper presented at the AEDS annual convention, 1973. Eric Document #ED087425.

Byrne, D. Teaching Oral English. London: Longman, 1976.

Candlin, C. (Ed.) The Communicative Teaching of English. London: Longman, 1981.

Chappelle, C., & Jamieson, J. Augmented BASIC, a better way to program ESL lessons. Paper presented at the TESOL annual convention, Toronto, 1983.

Cohen, V. B. Criteria for evaluation of microcomputer courseware. Educational Technology, 1983, 22(1), 9-14.

Cook, V. J., Some types of oral structure drills. In Allen H., & Campbell, R. (Eds.) Teaching English as a Second Language. New York: McGraw Hill, 1972, 120-129.

Crawford, S. A Standards Guide for the Authoring of Instructional Software Victoria, BC: Jem Research, 1981.
ERIC Document # ED211062.

Dacanay, F. R. Techniques and Procedures in Second Language Teaching. Quezon City, Phillipines: Aleman-Phoenix Publishing House, 1967.

Davies, G. Computers, Language, and Language Learning London: Centre for Information on Language Teaching and Research (CILT), 1982.

Dean, P. M., Computer-assisted instruction authoring systems. Educational Technology, 1978, 18(4), 20-23.

Dakin, J. The Language Laboratory and Language Learning. London: Longman, 1973.

References

Dimas, C. A strategy for developing CAI. Educational Technology, 1978, 18(4), 26-29.

Dowsey, M. W. Easy author - entry systems: A review and a prototype. International Journal of Man-Machine Studies, 1974, 6, 400-419.

Edwards, J. S., & Tillman, M. Computer-assisted instruction (CAI): A partner for PI. Performance and Instruction, 1982, 21(9), 19-21.

Eisele, J. Lesson design for computer-based instructional systems. Educational Technology, 1978, 18(9), 14-21.

Elliot, L., Gillette, D., & Brandt, W. A Guide for Authoring and Programming Computer Assisted Instruction, Salem, Oregon: Oregon Dept. of Education, 1982).
Eric Document # ED220661.

Fitzgibbon, N., & Grate, G. Computer-assisted instruction: Where do we go from here? Elementary English, 1970, 47, 917-921.

Funk & Wagnall's Standard College Dictionary. Montreal: Fitzhenry & Whiteside, 1976.

Gillespie, J. Striking the 'R': DeMythologyzing the computer revolution. NALLD Journal, 1983, 17(3-4), 4-12.

Goldes, H. Acronyms for education. PC Magazine, 1983 3(7), 353-357.

Grabinger, R. S. An Evaluation of three authoring language software packages. Techtrends, 1985, 30(4), 20-23.

Hardin, P. C., How to develop motivation in CAI-ESL materials: Ideas from Japan. Paper presented at the 1983 TESOL Convention, Toronto, March, 1983.

Hardy, R., & Elfner, E. S. Writing Courseware on the TRS-80. Creative Computing, 1982, 8(10), 34-37.

Hart, R. S. Language study and the Plato system. Studies in Language Learning., 1981, 3, 1-24.

References

Hazen, M. Computer-assisted instruction with PILOT on the Apple computer. Educational Technology, 1982, 22(11), 20-22.

Higgins, J. The use of the computer in English language teaching. In Davies, G. (Ed.) Computers, Language, and Language Teaching. London: Centre for Information on Language Teaching and Research (CILT), 1982, 25-30.

Higgins, J., & Johns, T. Computers in Language Learning. London: Addison-Wesley, 1984.

Holmes, G. Personal communication: July 1981.

Holmes, G. Computer-assisted instruction: A discussion of some of the issues for would-be implementors. Educational Technology, 1982, 22(9), 7-13.

Holmes, G. Creating CAL courseware: Some possibilities. System, 1983, 11(1), 21-32.

Holmes, G., & Kidd, M., Integrating CAI into the curriculum. In Smith, J. (Ed.) Proceedings of NECC-82. Kansas City: National Educational Computing Conference, 1982, 48-54. Eric Document # ED231349.

Hopper, C. Les applications pedagogiques de l'ordinateur en classe de francais. Colloquium held as part of the McGill Conference on Computers in Education, Montreal, December 12-14, 1984.

Hooper, R., Two Years On: The National Development Programme in Computer Assisted Learning, Report of the Director. London: Centre for Educational Technology (CET), 1975.

Jensen, J. D., A taxonomy of microcomputer authoring systems. Performance and Instruction, 1982, 21(6), 50-52.

Joiner, L. Microcomputers in Education. Holmes Beach, FL: Learning Publications, 1982.

Jonassen, D. H., The real case for using authoring systems to develop courseware. Educational Technology, 1985, 25(2), 39-41.

References

Kearsley, G. Some facts about CAI: A Quantitative analysis of the 1976 index to computer-based instruction. Journal of Computer-Based Instruction, 1976, 3(2), 30-36.

Kearsley, G. Authoring systems in computer-based education. Communications of the ACM, 1982, 25(7), 429-437.

Kenner, R., & Richards, G. A homemade CAI package for use in the language lab. NALLD Journal, 1980, 15(1), 31-34.

Kenning, M. J., & Kenning, M. M. Introduction to Computer Assisted Language Teaching. London: Oxford University Press, 1983.

Last, R. Computer assisted language learning: A Growth area in humanities computing. ALLC Bulletin, 1983, 11(3), 83-86.

Littlewood, W. Communicative Language Teaching. London: Cambridge University Press, 1981.

Locatis, C., & Carr, V. Selecting authoring systems. Journal of Computer Based Instruction, 1985, 12(2), 28-33.

Marty, F. Reflections on the use of computers in second language acquisition. Studies in Language Learning, 1981, 3, 25-53.

McCambridge, C. Uses of microcomputers in education. A Panel Discussion as part of Teleconference organized by the Ohio Dean's Network. October 29, 1982.

McGee, J. Uses of microcomputers in education. A Panel Discussion as part of Teleconference organized by the Ohio Dean's Network. October 29, 1982.

Merrill, P. F., The case against PILOT: The pro's and con's of computer-assisted instruction languages and authoring systems. Creative Computing, 1982, 8(7), 70-77.

Merrill, P. F., & Salisbury, D. Research on drill and practice strategies. Journal of Computer-Based Instruction, 1984, 11(1), 19-21.

References

Mydlarski, D., So you want to do Your own CAL program: The Experience at the University of Calgary. In Kenner, R. (Ed.) Proceedings of the Concordia Colloquium on Language Laboratories, 1981. Montreal, SPEAQ Journal, 1980, 119-128.

Olsen, S., Foreign language departments and computer assisted instruction: A Survey. Modern Language Journal, 1980, 64(3), 341-350.

O'Neal, F., & Fairweather, F. The evaluation and selection of computer based training authoring systems. CALICO Journal, 1984, 2(2), 27-31.

Orstein, J. Once more: Programmed instruction in the language field: The State of the art. Language Learning, 1970, 20(2), 213-222.

Otto, S., & Pusack, J. Stringing us along, Programming for foreign language CAI. CALICO Journal, 1983, 1(2), 26-33.

Paloian, A. Y. An interrogative authoring system. International Journal of Man-Machine Studies, 1974, 6, 445-463.

Paramskas, D., Courseware-software interfaces: Some designs and some problems. CALICO Journal, 1983, 1(3), 4-6.

Paramskas, D., Progiciels et matrices (langues secondes). Journal de l'APOP, 1984, 2(2), 14-15.

Paramskas, D., & Mydlarski, D. Prompt Authoring system software produced by the University of Guelph and the University of Calgary, 1985.

Pattison, L., Authoring package update: Software writing made easy. Electronic Learning, 1985, March, 30-36.

Paul, D. M., Handbook for the Identification and Assessment of Computer Courseware for the Adult Learner. Harrisburg, PA: Pennsylvania State Dept. of Education, 1982.
ERIC Document # ED228457.

References

Paulston, C. B., Structural pattern drills: A classification. In Allen, H., & Campbell, R. (Eds.) Teaching English as a Second Language. New York: McGraw Hill, 1972, 129-138.

Peuchot, M. La Finalite de l'enseignement assiste par ordinateur. Paper given at the First Annual McGill Conference on Computers in Education, Montreal, December 12-13, 1983.

Piepho, H. E. Establishing objectives in the teaching of English. In Candlin C. (Ed.) The Communicative Teaching of English. London: Longman, 1981, 45-62.

Pogue, R. E. The authoring system: Interface between author and computer. Journal of Research and Development in Education, 1980, 14(1), 57-68.

Pusack, J. P. Dasher. Authoring system software produced by Conduit Corp., Iowa City, Iowa, 1982.

Pusack, J. P. Answer-processing and error correction in foreign language CAI. System, 1983, 11(1), 53-64.

Raschias, R., & Lange, D. A discussion of the attributes, role, and uses of CAI materials in foreign languages. Journal of Computer Based Instruction, 1984, 11(1), 22-27.

Rivers, W. M. Teaching Foreign Language Skills. Chicago: University of Chicago Press, 1968.

Rivers, W. M., & Temperley, M, S. A Practical Guide to the Teaching of English. New York: Oxford University Press, 1978.

Robinett, B. W., Teaching English to Speakers of Other Languages: Substance & Technique. Minneapolis: University of Minnesota Press, 1978.

Sanders, D., & Kenner, R. Whither CAI?: The need for communicative courseware. System, 1983-a, 11(1), 33-39.

Sanders, D., & Kenner, R. Computer assisted language instruction symposium. Symposium held at the 1983 TESOL Convention, Toronto, Ontario, March 15, 1983-b.

References

Scanlan, R. Computer assisted instruction: PLATO in Latin. Foreign Language Annals, 1971, 5(1), 84-89.

Schuyler, J. Programming language for microprocessor courseware. Educational Technology, 1979, 19(10), 29-36.

Stevens, V. Review of English lessons on PLATO. TESOL Quarterly, 1983-a, 17(2), 293-300.

Stevens, V. A Report of a project illustrating the feasibility of video/computer interface for use in ESL. CALICO Journal, 1983-b, 1(1), 27-30.

Suppes, P., & Mackey, E. The historical path from research and development to operational use of CAI. Educational Technology, 1978 18(4), 9-12.

Stack, E., The Language Laboratory and Modern Language Teaching--Third edition. London: Oxford University Press, 1971.

Stevick, E. W. Adapting and Writing Language Lessons. Washington: Foreign Service Institute, 1971.

Villeneuve, R. Langages-auteur, systemes-auteur. Bulletin de l'APOP, 1984, 2(2), 5.

Weible, D. The foreign language teacher as courseware author. CALICO Journal, 1983, 1(1), 62-64.

Wilson, J. Uses of microcomputers in education. A panel discussion as part of Teleconference organized by the Ohio Dean's Network. October 29, 1982.

Wright, A. Microcomputers in the Schools: New Directions for British Columbia. Victoria: British Columbia Dept. of Education, 1980. ERIC Document #ED208846.

Wyatt, D. Personal communication: October, 1982.

Wyatt, D. Three major approaches to developing computer assisted language learning materials for microcomputers. CALICO Journal, 1983-a, 1(2), 33-38.

Wyatt, D. Computer assisted language instruction: Present state and prospects. System, 1983-b, 11(1), 3-11.

References

Zinn, K. L. Requirements for effective authoring systems and assistance. _International Journal of Man-Machine Studies_, 1974, 6, 381-400.

Appendix A: Table 3.1

Sources of the Original Drill Examples:

Year	Author	Source	Number of Examples
1967	F. R. Daçanay	classroom drills	14
1971	E. Stack	language lab drills	23
1971	E. Stevick	classroom drills	2
1972	V. J. Cook	classroom drills	13
1972	C. Paulston	classroom drills	3
1973	J. Dakin	language lab drills	8
1976	D. Byrne	classroom drills	9
1978	W. Rivers & M. S. Temperley	classroom drills	19
1978	B. Robinett	classroom drills	13
1979	W. Beile	language lab drills	8
1981	C. Candlin	classroom drills	15

Total Number of Examples Considered:			127

Appendix A: Table 3.2

List of Drill Eamples Exemplifying Operational Typologies

1.	3.4	Dacanay	Simple Substitution
2.	3.5	Dacanay	Correlative Substitution
3.	3.6	Dacanay	Moving Slot Substitution
4.	3.7	Dacanay	Transposition Transformation
5.	3.8	Dacanay	Transposition Transformation
6.	3.9	Dacanay	Transposition Transformation
7.	3.10	Dacanay	Expansion Transformation
8.	3.11	Dacanay	Expansion Transformation
9.	3.12	Dacanay	Transposition + Expansion
10.	3.13	Dacanay	Reduction Tranformation
11.	3.14	Dacanay	Integration Transformation
12.	3.15	Dacanay	Integration + Transposition
13.	3.16	Dacanay	Integration + Transposition + Reduction
14.	3.16	Dacanay	Integration + Transposition + Reduction
15.	3.20	Rivers*	Simple Substitution
16.	3.21	Rivers	Double Substitution
17.	3.22	Rivers	Correlative Substitution
18.	3.24	Rivers	Multiple Substitution
19.	3.26	Rivers	General Conversion
20.	3.27	Rivers	General Conversion
21.	3.28	Rivers	General Conversion
22.	3.29	Rivers	General Conversion
23.	3.30	Rivers	Combination Conversion
24.	3.31A	Rivers	Restatement Conversion
25.	3.31B	Rivers	Restatement Conversion
26.	3.32	Rivers	Type B Expansion
27.	3.33	Rivers	Type A Deletion
28.	3.34	Rivers	Type A1 Completion
29.	3.35	Rivers	Type A2 Completion
30.	3.36	Rivers	Type B Completion
31.	3.37	Rivers	Question-Answer Practice
32.	3.38	Rivers	Question-Answer Practice
33.	3.39	Rivers	Rejoinder Exercise
		* (Rivers & Temperley)	
34.	3.40	Robinett	Simple Substitution
35.	3.41	Robinett	Correlative Substitution
36.	3.42	Robinett	Moving Slot Substitution
37.	3.43	Robinett	Moving Slot + Correlative Substitution
38.	3.44	Robinett	Transposition Transformation

Appendix A: Table 3.2

39.	3.45	Robinett	Integration Transformation
40.	3.46	Robinett	Reduction Transformation
41.	3.47	Robinett	Response Transformation
42.	3.54	Cook	Typical Drill
43.	3.59	Cook	Plain Substitution
44.	3.60	Cook	Plain Substitution
45.	3.61	Cook	Sequence Substitution
46.	3.62	Cook	Lexical Pair Substitution
47.	3.63	Cook	Lexical Sets Substitution
48.	3.64	Cook	Lexical Meaning Substitution
49.	3.65	Cook	Pronoun Substitution
50.	3.66	Cook	Knowledge Substitution
51.	3.106	Stack	Replacement
52.	3.111	Stack	Replacement
53.	3.114	Stack	Replacement
54.	3.116	Stack	Transformation: verb tenses
55.	3.117	Stack	Transformation: verb tenses
56.	3.118	Stack	Transformation: directed increment
57.	3.120	Stack	Transformation: negation
58.	3.125	Stack	Transformation: comparisons
59.	3.127	Stack	Transformation: word order
60.	3.129	Stack	Question Formation
61.	3.130	Stack	Chain Transformation
62.	3.131	Stack	Paired Sentence: relative pronouns
63.	3.132	Stack	Paired Sentence: conditionals
64.	3.134	Stack	Paired Sentence: tense linkages
65.	3.136	Stack	Paired Sentence
66.	3.140	Stack	Fixed Increment
67.	3.142	Stack	Analogy Drill: tag questions
68.	3.143	Stack	Analogy Drill: tag questions
69.	3.144	Stack	Question Drill: you-I conversion
70.	3.145	Stack	Question Drill: narration sequence
71.	3.146	Stack	Question Drill: directed answers

Appendix A: Table 3.3

List of Drill Examples Exemplifying Pedagogical Typologies

1.	3.48	Robinett	Mechanical Drill
2.	3.49	Robinett	Meaningful Drill
3.	3.50	Robinett	Communicative Drill
4.	3.158	Robinett	Manipulative Drill
5.	3.159	Robinett	Communication Drill
6.	3.51	Paulston	Meaningless Drill -1
7.	3.52	Paulston	Meaningless Drill -2
8.	3.53	Paulston	Meaningful Drill
9.	3.55	Cook	Non-Contextualised Drill
10.	3.56	Cook	Semi-Contextualised Drill
11.	3.57	Cook	Contextualised Drill
12.	3.58	Cook	Situational Drill
13.	3.68	Byrne	Mechanical Drill
14.	3.69	Byrne	Meaningful Drill -1
15.	3.70	Byrne	Meaningful Drill -2
16.	3.71	Byrne	Special Meaningful Drill
17.	3.72	Byrne	Open Ended Responses
18.	3.73.1	Byrne	Imaginary Situations -1
19.	3.73.2	Byrne	Imaginary Situations -2
20.	3.74	Byrne	Meaningful Drill -3
21.	3.75	Byrne	Meaningful Drill -4
22.	3.76	Candlin	Get Organized: Concept
23.	3.77	Candlin	Get Organized: Narration
24.	3.78	Candlin	Get Organized: True/false
25.	3.79	Candlin	Get Organized: Connecting
26.	3.80	Candlin	Get Organized: Multiple chc
27.	3.81	Candlin	Get Organized: Recognition
28.	3.82	Candlin	Get Organized: Word Scramble
29.	3.84	Candlin	Implanting Skills: Memorizing Cognitive Language
30.	3.85	Candlin	Implanting Skills: Memorizing Affective Language
31.	3.86	Candlin	Implanting Skills: Picture Skills
32.	3.87	Candlin	Implanting Skills: Restoration-1

Appendix A: Table 3.3

33.	3.89	Candlin	Implanting Skills: Restoration: Catchwords
34.	3.90	Candlin	Implanting Skills: Restoration: Gap Text
35.	3.91	Candlin	Implanting Skills Restoration: Gap Text/New Text
36.	3.92	Candlin	Implanting Skills Restoration: Defective Dialogue
37.	3.93	Stevick	Manipulative
38.	3.94	Stevick	Meaningful
39.	3.95	Dakin	Application Drill
40.	3.96	Dakin	General Knowledge Drill
41.	3.97	Dakin	Collocation Drill
42.	3.98	Dakin	Synonymy Drill
43.	3.99	Dakin	Hyponymy Drill
44.	2.100	Dakin	Antonymy Drill
45.	3.101	Dakin	Converse Drill
46.	3.102	Dakin	Consequence Drill
47.	3.150	Beile	Non-Communicative Interchange
48.	3.151	Beile	Isolated Communicative Interchange
49.	3.152	Beile	Isolated Communicative Interchange in a Situational Context
50.	3.153	Beile	Connected Communicative Interchange
51.	3.154	Beile	Connected Communicative Interchange in a Situational Context
52.	3.155	Beile	No Contextualisation
53.	3.156	Beile	Dialog-like but Not Contextualisable
54.	3.157	Beile	Contextualisable.
55.	3.160	Stack	Meaningful Drill -1 (unpatterned)
56.	3.161	Stack	Meaningful Drill -2 (patterned)

Appendix A Table 3.4

The Variable Environment of Different Drill Presentation Media

	Oral Classroom Drill	Language Lab Drill	Written Exercise	CALL
1. Variation in the flow of the drill, based on situation.	Yes	No	No	Yes
2. Variation in individual stimuli, based on the situation.	Yes	No	No	Yes
3. Pause for reflection is possible.	No	Yes	Yes	Yes
4. Mediator can focus on a single student.	No	No	Yes	Yes
5. More than one acceptable response can be handled.	Yes	No	No	Yes
6. Immediate, feedback can be given.	Yes	Yes	Yes	Yes
7. Feedback related to the individual can be given	Yes	No	No	Yes

a. For the Language Lab column, assume the learner is working with the tape only, without a teacher monitoring the interchange.

b. For the Written Drill column, assume that the material is NOT programmed instruction and that, except in the case of point #4, the answer-key is the sole mediator.

Appendix A Table 3.5

Categorization Schemes Discussed by the Authors Consulted

Author	Operational	Pedagogical
Dacanay	XXX	
Stack	XXX	XXX
Stevick	XXX	XXX
Cook	XXX	XXX
Paulston		XXX
Dakin	XXX	XXX
Byrne		XXX
Rivers	XXX	
Robinett	XXX	XXX
Beile		XXX
Candlin		XXX

Appendix A Table 3.6

Typical Categorizations of Drills and Activities according to Traditional Operational Type:

DACANAY	(1967, p. 107)	a. substitution b. transformation c. response d. translation
STACK	(1971, p 140)	a. replacement b. transformation c. chain transformation d. paired sentence e. fixed increment f. analogy g. question drills
STEVICK	(1971, p. 394)	a. substitution b. transformation
COOK	(1972, p. 124)	a. substitution b. mutation c. repetition d. addition
DAKIN	(1973, p. 48)	a. substitution b. mutation c. transformation
RIVERS & TEMPERLEY	(1978, p. 126)	a. repetition b. substitution c. conversion d. sentence modification e. response practice f. translation
ROBINETT	(1978, pp.47-55)	a. imitation b. substitution c. transformation d. response e. translation

Appendix A Table 3.7

Correlations between traditional drill typologies

	DACANAY	STACK	STEVICK	COOK
A.				repetition
B.	substitution	replacement	substitution	substitution
C.	transformation	transformation chain transformation	transformation	mutation
D.		paired sentence fixed increment analogy		addition
E.	response	question drills		
F.	translation			

DAKIN	RIVERS	ROBINETT
	repetition	imitation
substitution	substitution	substitution
mutation transformation	conversion	transformation
	sentence modification	
	response practice	response
	translation	translation

Appendix A: Table 3.14

Typical Categorizations of Drills and Activities according to Pedagogical Gradient:

STACK	(1971, pp. 133-137)	a. meaningless b. meaningful
DAKIN	(1973, pp. 47-89)	a. meaningless b. meaningful
STEVICK	(1971, pp. 394, 401)	a. manipulative b. meaningful
BYRNE	(1976, pp. 33-38)	a. mechanical b. meaningful
ROBINETT	(1978, pp. 207-211)	a. mechanical b. meaningful c. communicative and,.. a. manipulative b. communication
PAULSTON	(1972)	a. meaningless b. meaningful c. communicative
COOK	(1972, pp. 122-124)	a. non-contextualised b. semi-contextualised c. contextualised d. situational
BEILE	(1979, pp. 115-121)	Communicativity gradient: a. non-communicative b. (communicative) Contextualisation gradient: a. no contextualisation b. dialog-like, but not contextualisable. c. contextualisable
CANDLIN	(1981, p. 5)	a. (non-communicative) b. communicative

Appendix A: **Table 3.15**

Various Degrees or Types of "Communicative Activity"

STEVICK (1971, p. 30)

a. realistic language use
b. real language use

BEILE (1979, p. 121)

a. isolated communicative
b. isolated communicative in situational context
c. connected communicative
d. connected communicative in situational context.

CANDLIN (1981, p. 5)

Exercises for ...

i. getting Skills
ii. implanting Skills
iii. developing Skills
iv. using Skills

PEIPHO (1981, p. 20)

Activities which ...

a. prepare for communication
b. develop communication
c. structure communication
d. simulate communication
e. are inherently communicative

Appendix A: The Original Drill Sample

3.4 Example: Simple Substitution

#1	Model:	I'm going home.
	Cue:	away.
	Response/	I'm going away.
#2	Model:	
	Cue:	later.
	Response:	I'm going later.

(Dacanay, 1967, p. 112)

3.5 Example: Correlative Substitution

#1	Model:	Where does Mary live?
	Cue:	José
	Response/	Where does José live?
#2	Model:	
	Cue:	the boys
	Response:	Where do the boys live?

(Dacanay, 1967, p. 113)

3.6 Example: Moving Slot Substitution

#1	Model:	The train leaves tomorrow at seven.
	Cue:	tonight
	Response/	The train leaves tonight at seven.
#2	Model:	
	Cue:	bus
	Response:	The bus leaves tonight at seven.

(Dacanay, 1967, p. 116)

3.7 Example: Transposition Transformation

Instruction: Change the statement to a question.

#1	Model:	The girl is ready.
	Response:	Is the girl ready?
#2	Model:	The boy is coming.
	Response:	Is the boy coming?

(Dacanay, 1967, p. 117)

Appendix A: The Original Drill Sample

3.8 Example: Transposition Transformation -2

Instruction: Change the position of the sentence modifiers

#1 Model: This morning he was here.
Response: He was here this morning.

#2 Model: Occasionally he comes here.
Response: He comes here occasionally.

(Dacanay, 1967, p. 117)

3.9 Example: Tranpsosition Transformation -3

#1 Model: Mother made me a dress.
Cue: for me.
Response: Mother made a dress for me.

#2 Model: My friend gave me the pen.
Cue: to me.
Response: My friend gave the pen to me.

(Dacanay, 1967, p. 118)

3.10 Example: Expansion Transformation -1

"Order of Modifiers"

#1 Model: The napkins are on the table.
Cue: two
Response/ The two napkins are on the table.

#2 Model:
Cue: paper
Response/ The two paper napkins are on the table.

#3 Model:
Cue: pretty
Response/ The two pretty paper napkins are on the table.

#4 Model:
Cue: yellow
Response: The two pretty yellow paper napkins are on the table.

(Dacanay, 1967, p. 120)

Appendix A: The Original Drill Sample

3.11 Example: Expansion Transformation -2

"Placement of Adverbs of Frequency"

#1 Model: He's on time.
Cue: always
Response: He's always on time.

#2 Model: I write to them.
Cue: always
Response: I always write to them.

#3 Model: I write to them
Cue: regularly
Response: I write to them regularly.
I regularly write to them.

#4 Model: He's leaving.
Cue: now
Response: He's now leaving.
He's leaving now.

(Dacanay, 1967, p. 124-125)

3.12 Example: Transposition & Expansion

#1 Model: He likes coffee.
Cue: not
Response: He doesn't like coffee.

#2 Model: Mary buys instant coffee.
Cue: not
Response: Mary doesn't buy instant coffee.

(Dacanay, 1967, p. 127)

3.13 Example: Reduction Transformation

Model: The children who were making a lot of noise were sent to bed.
Cue: noisy
Response: The noisy children were send to bed.

(Dacanay, 1967, p. 128)

Appendix A: The Original Drill Sample

3.14 Example: Integration Transformation

Instruction: Combine the two questions without transposing or reducing any of the elements.

Model: Who made the report to the principal?
Do you know?

Response: Do you know who made the report to the principal?

(Dacanay, 1967, p. 130)

3.15 Example: Integration Transformation with Transposition

Instruction: Combine the two questions, putting the yes/no question first.

Model: Who is he?
Do you know?

Response: Do you know who he is?

(Dacanay, 1967, p. 131)

3.16 Example: Integration Transformation with Reduction

Instruction: Combine the two sentences using "too...to"

Model: Mary is sick.
She can't study her lesssons.

Response: Mary is too sick to study her lessons.

(Dacanay, 1967, p. 130)

Appendix A: The Original Drill Sample

3.20 **Example:** **Simple Substitution**

#1 Model/Cue: Do you see my uncle over there?

Response: Yes, he's across the road.

#2 Model/Cue: Do you see my sister over there?

Response: Yes, she's across the road.

(Rivers & Temperley, 1978, p. 123)

NOTE: Referred to in the text as: 3.20(RIVERS). Similarly for all of the Rivers & Temperley examples.

3.21 **Example:** **Double Substitution Drill**

#1 Model: If I find it / I'll give it to you.

Cue: If you want it

Response/ If you want it / I'll give it to you.
#2 Model:

Cue: he'll sell it to you

Response: If you want it / he'll sell it to you.

(Rivers & Temperley, 1978, p. 127)

3.22 **Example:** **Correlative Substitution**

#1 Model: He brings his lunch.

Cue: You

Response/ You bring your lunch.
#2 Model:

Cue: John and Mary

Response: John and Mary bring their lunch.

(Rivers & Temperley, 1978, p. 128)

Appendix A: The Original Drill Sample

3.24 **Example:** Multiple Substitution

#1 Model: She brings too many pencils to school.

Cue: Peter

Response/ Peter brings too many pencils to school.
#2 Model:

Cue: money

Response/ Peter brings too much money to school.
#3 Model:

Cue: library

Response: Peter brings too much money to the library.

(Rivers & Temperley, 1978, p. 129)

3.26 **Example:** General Conversion

Instruction: Convert the following statements into questions

#1 Model: Peter has a new car.

Response: Does Peter have a new car? or
Has Peter got a new car?

#2 Model: They stop at stop signs

Response: Do they stop at stop signs?

(Rivers & Temperley, 1978, p. 131,132)

Appendix A: The Original Drill Sample

3.27 Example: General Conversion -2
("Mixed Drill")

Instruction: Change the following statements into questions

#1 Model: John and I are sitting in the classroom.

Response: Are John and I sitting in the classroom?

#2 Model: The actress lives in Canada.

Response: Does the actress live in Canada?

(Rivers & Temperley, 1978, p. 131)

3.28 Example: General Conversion -3
(Reduction Conversion)

Instruction: Replace the indicated word with the correct pronoun.

#1 Model: Janet read her mother the letter.

Cue: her mother

Response/ Janet read her the letter.
Janet read the letter to her.

#2 Model:

Cue: the letter

Response: Janet read it to her

(Rivers & Temperley, 1978, p. 133)

Appendix A: The Original Drill Sample

3.29 Example: General Conversion -3
(Expansion Conversion)

Instruction: Replace the pronoun indicated with a substantive word.

#1 Model: I gave it to her.

Cue: it

Response/ I gave a car to her.
I gave her a car.

#2 Model:

Cue: her

Response: I gave a car to Janet.

(Rivers & Temperley, 1978, p. 133)

3.30 Example: Combination Conversion

Instructions/Cue: Combine each of the following pairs of sentences into one acceptable sentence, using "that" where necessary and omitting it where possible. Make the first sentence the main clause.

#1 Model: Give me the keys.
I left you the keys.

Response: Give me the keys I left you.

#2 Model: The car is over there.
I bought the car yesterday.

Response: The car I bought yesterday is over there.

#3 Model: Don't close the door.
The door has just been painted.

Response: Don't close the door that's just been painted.

(Rivers & Temperley, 1978, p. 136)

Appendix A: The Original Drill Sample

3.31a Example: Restatement Conversion

#1	Cue:	Tell George your name is Ronald.
	Response:	George, my name is Ronald.
#2	Cue:	Ask Alice where she's going.
	Response:	Where are you going, Alice ?
#3	Cue:	Ask her to wait for you.
	Response:	Wait for me, please.

(Rivers & Temperley, 1978, p. 137)

[Wait, please.
Alice, wait !
Alice, wait up !
Just a minute, Alice.
Hold on for a second, please.
etc.]

3.31 Example: Restatement Conversion -2

Instruction: Restate the following quotations, using indirect speech.

#1 Model: She said, "I've just arrived but I'm leaving in a few minutes."

Response: She said she had just arrived but was leaving in a few minutes."

#2 Model: She asked, "Why are you looking at me like that ?"

Response: She asked me why I was looking at her like that.

(Rivers & Temperley, 1978, p. 137)

Appendix A: The Original Drill Sample

3.32 **Example: Type B Expansion**

#1	Model/Cue:	The man crosses the street.
#2	Response/ Model/Cue:	The tired old man crosses the busy street.
#3	Response/ Model/Cue:	The busy business man crosses the street twice a day.

(Rivers & Temperley, 1978, p. 140)

3.33 **Example: Type A Deletion**

Instruction: Delete the negative elements in the following sentences, making any necessary changes.

#1	Model:	They haven't any coffee.
	Response:	They have some coffee.
#2	Model:	She didn't come.
	Response:	She came.
#3	Model:	You came, didn't you ?
	Response:	You didn't come, did you ? You came, did you ?

(Rivers & Temperley, 1978, p. 140)

Appendix A: The Original Drill Sample

3.34 Example: Type A1 Completion

Instructions: Vary the final segment from future to conditional, according to the first segment

#1 Model: If I see him I'll tell him

Cue: If I saw him ...

Response/ #2 Model: If I saw him I'd tell him

Cue: If you took it ...

Response/ #3 Model: If you took it I'd tell him

Cue: If she comes ...

Response: If she comes I'll tell him

(Rivers & Temperley, 1978, p. 141)

3.35 Example: Type A2 Completion

Instructions: Complete the following statements with the appropriate occupational term

#1 Model: A person who drives a cab is a cab driver.

Cue: A person who builds houses ...

Response/ #2 Model: A person who builds houses is a builder

Cue: A person who sells meat ...

Response: A person who sells meat is a butcher

(Rivers & Temperley, 1978, p. 141)

Appendix A: The Original Drill Sample

3.36 Example: Type B Completion

Instructions: Invent a completion containing an infinitive construction for each sentence you hear

#1 Cue: She has decided ...

Response: She has decided to marry him
She has decided not to marry him
She has decided to leave the country
She has decided to take a job

#2 Cue: You needn't ...

Response: You needn't go to school today
You needn't pay me yet
You needn't tell me your secrets

(Rivers & Temperley, 1978, p. 142)

3.37 Example: Question-Answer Practice

(A Film about the adventures of a group of young people in Boston)

#1 Cue: Why didn't they come home before midnight?

Response: They didn't come home because there were fireworks over the river. It was the Fourth of July.

#2 Cue: Do you often stay out late at night?

Response: No, I go home early because I am always hungry.

(Rivers & Temperley, 1978, pp 146-147)

Appendix A: The Original Drill Sample

3.38.1 Example: Answer-Question Pracice

"In *Twenty Questions* one person (A) thinks of someone or something. By asking eliminating questions to which A may reply only yes or no, the players narrow the field of possibilities until they are able to guess the person or object in question." (Rivers & Temperley, 1978, p. 145)

3.38.2 Computer Example: Answer-Question Practice

Animal (Apple Demonstration Program)

#1

1. Computer: Think of an animal. I will try to what it is.
   Press Return to Begin

2. Computer: Does it live on a farm ?

3. Student: No.

4. Computer: Is it an elephant ?

5. Student: No.

6. Computer: I do not know this animal. What is it ?

7. Student: A rhinoceros.

8. Computer: Give me a question to distinguish a rhinoceros from an elephant.

9. Student: Does it have a horn ?

10. Computer: For a rhinoceros, is the answer Yes or No ?

11. Student: Yes.

12. Computer: Thank you.

#2

1. Computer: Think of an animal. I will try to what it is.
   Press Return to Begin

Appendix A: The Original Drill Sample

3.39 Example: Rejoinder Exercise

Instruction: Listen to the following sentences and respond with an appropriate exclamation or rejoinder.

#1 Cue: Didn't get the job. Just got there as the manager went off to lunch.

Response: Tough luck !
Too Bad !
That's too bad !
Sorry about that !

#2 Cue: See you at the bus stop.

Response: Ok.
Okay.
Sure.
Right.

Sorry.
Can't make it.

(Rivers & Temperley, 1978, p. 145)

3.40 Example: Simple Substitution

Instructions: Substitute the given word at the indicated place in the model.

#1 Model: The book is on the table.

Cue: pen

Response/ The pen is on the table.
#2 Model:

Cue: box

Response: The box is on the table.

(Robinett, 1978, p. 48)

Appendix A: The Original Drill Sample

3.41 **Example:** **Correlative Substitution**

Instructions: Substitute the given word at the indicated place in the model and make whatever changes are necessary to the verb.

#1 Model: Mary is studying.

Cue: Mary and John

Response/ Mary and John are studying.
#2 Model:

Cue: My brother

Response: My brother is studying.

(Robinett, 1978, p. 49)

3.42 **Example:** **Moving Slot Substitution (Robinett)**

Instructions: Substitute the given word at the appropriate place in the model

#1 Model: She bought a car yesterday.

Cue: He

Response/ He bought a car yesterday.
#2 Model:

Cue: house

Response: He bought a house yesterday.

(Robinett, 1978, p. 49)

Appendix A: The Original Drill Sample

3.43 Example: <u>Combined Moving Slot and Correlative Substitution</u>

Instructions: Substitute the given word at the appropriate place in the model and make whatever resultant changes are necessary.

#1 Model: She is buying a car today.

Cue: yesterday

Response/ She bought a car yesterday.
#2 Model:

Cue: They

Response: They bought a car yesterday.

(Robinett, 1978, p. 49)

3.44 Example: <u>Transposition Tranformation</u>

Instructions: a. Transform the statement into a question, then
b. answer the question.

#1a Model: She can swim.

Response/ Can she swim?
#1b Model:

Response: Yes, she can
No, she can't

#2a Model: She can play tennis.

Response/ Can she play tennis?
#2b Model:

Response: Yes, she can
No, she can't

(Robinett, 1978, p. 50)

Appendix A: The Original Drill Sample

3.45 Example: Integration Tranformation

Instructions: a. Combine the two questions into one question, then
b. answer the question.

#1a Model: Who is she?
Do you know?

Response/ Do you know who she is?
#1b Model:

Response: Yes, I do.
No, I don't

#2a Model: How far is it to Chicago?
Do you know?

Response/ Do you know how far it is to Chicago?
#2b Model:

Respohse: Yes, I do.
No, I don't.

(Robinett, 1978, p. 50)

3.46 Example: Reduction Transformation

Instruction: Paraphrase each sentence as in the example below:

She's going to have someone paint the garage.
She's going to have the garage painted.

#1 Model: She's going to have someone mow the lawn.

Response: She's going to have the lawn mowed.

#2 Model: She's going to have someone fix the sidewalk.

Response: She's going to have the sidewalk fixed.

(Robinett, 1978, p. 51)

Appendix A: The Original Drill Sample

3.47 Example: Response Drill #1

Instructions: Answer the following questions in the affirmative, using "just a few" or "just a little", as required.

#1 Model: Do you need any chairs?

Response: Yes, just a few, please.

#2 Model: Do you need any help?

Cue: Yes, just a little, please

(Robinett, 1978, p. 52)

3.48 Example: Mechanical Drill

Instructions: Substitute the following words in the sentence, and make the appropriate change in their form.

#1 Model: John is taller than Bill.

Cue: old

Response/ John is older than Bill.

#2 Model:

Cue: big

Response: John is bigger than Bill.

(Robinett, 1978, p. 208)

Appendix A: The Original Drill Sample

3.49 Example: Meaningful Drill

Instructions: Answer the following questions according to the information contained in the picture.

(Picture of several people, showing height and citing weight, name, and age)

#1 Model/Cue: Who is older, Susie or David?

Response: Susie.

#2 Model/Cue: Who is shorter, Susie or Mary?

Response: Mary.

(Robinett, 1978, p. 208)

3.50 Example: Communicative Drill

Instructions: Answer the following questions truthfully.

(Obvious visual information and students' knowledge of each other)

#1 Model/Cue: Who is older, Susie or David?

Response: Susie.

#2 Model/Cue: Who is shorter, Susie or Mary?

Response: Mary.

(Robinett, 1978, p. 209)

Appendix A: The Original Drill Sample

3.51 Example: Meaningless Drill -1

Instructions: [Replace the adjective in the model with the cue given.]

#1 Model: the thin student

Cue: tall

Response/ the tall student
#2 Model:

Cue: fat

Response: the fat student

(Paulston, 1972, p. 133)

3.52 Example: Meaningless Drill -2

Instructions: [Make the following active sentences into passive ones according to the example:

John kicked the door.
The door was kicked by John.]

#1 Model: The dog bit the man.

Response: The man was bitten by the dog.

#2 Model: The boing boinged the boing.

Response: The boing was boinged by the boing.

(Paultson, 1972, p. 135)

Appendix A: The Original Drill Sample

3.53 <u>Example:</u> <u>Meaningful Drill-1</u>
"Association/Fixed Reply Actitity"

Instruction: Ask the question suggested by the answer

#1 Model: ...for five years.

Response: How long did he study?

#2 Model: ...during March.

Response: When did he register?

#3 Model: ...until four o'clock.

Response: [Until when did he study?
Until what time did he study?]

(Paulston, 1972, p. 134)

3.54 <u>Example</u> <u>Typical Drill</u>

Instruction: [Answer the question in the negative]

#1 Model: Is Bill playing tennis tonight?

Response: No, he's not going to play.

#2 Model: Is Susan helping her mother this evening?

Response: No, she's not going to help.

#3 Model: Are Mr. and Mrs. Green paying the bill tomorrow?

Response: [No, they're not going to pay.]

(Cook, 1972, p. 122)

Appendix A: The Original Drill Sample

3.55 Example Non-Contextualised Drill

Instruction: [Substitute the cue for the appropriate word in the model]

#1 Model: John's going to Paris.

Cue: He

Response/ He's going to Paris.
#2 Model:

Cue: She

Response: She's going to Paris.

(Cook, 1972, p. 123)

3.56 Example Semi-Contextualised Drill

Instruction: [Use the name from the model in your response. in the model]

#1 Model/Cue: Fred's going to change his job.

Response: Fred? Changing his job ?
I don't believe it !

#2 Model/Cue: Jane's going to clean the car.

Response: Jane? Cleaning the car ?
I don't believe it !

(Cook, 1972, p. 123)

Appendix A: The Original Drill Sample

3.57 Example Contextualised Drill

Instruction: [Respond in the negative, using the correct pronoun.]

#1 Model/Cue: Are you coming to the party ?

Response: No, I'm not.

#2 Model/Cue: But Susan's coming, I'm sure.

Response: No, She's not.

#3 Model/Cue: Well I know Basil's going to be there.

Response: [No, He's not.]

(Cook, 1972, p. 123)

3.58 Example Situational Drill

Instruction: [Answer the question, based on the situation.]

#1 Model: (Student is cleaning blackboard)

Cue: What's he doing ?

Response: He's cleaning the blackboard.
He's working...
He's erasing the board.
etc.

(Cook, 1972, p. 123)

Appendix A: The Original Drill Sample

3.59 Example: Plain Substitution Drill -1

Instructions: Respond to the cue using the model: "I love ..."

#1 Cue: Do you like whisky ?

Response: I love whisky.

#2 Cue: Do you like tea ?

Response: I love tea. (Cook, 1972, p. 125)

3.60 Example: Plain Substitution Drill -2

Instructions: Insert the cue into the model: "I love whisky"

#1 Cue: whisky

Response: I love whisky.

#2 Cue: hate

Response: I hate whisky.

#3 Cue: He

Response: He hates whisky.

(Cook, 1972, p. 125)

3.61 Example: Sequence Substitution Drill

Instructions: Insert the second item mentioned in the cue into the model "Oh, I prefer ..."

#1 Cue: I can't decide whether I like swimming or skating best.

Response: Oh, I prefer skating.

#2 Cue: I can't decide whether I like dancing or walking best.

Response: Oh, I prefer walking.

(Cook, 1972, p. 125)

Appendix A: The Original Drill Sample

3.62 <u>Example: Lexical Pair Substitution Drill</u>

Instructions: Respond to the cue in the negative, using the model "No, he's (antonym)".

#1 Cue: Is Bill young ?

Response: No, he's old.

#2 Cue: Is John rich ?

Response: No, he's poor.

(Cook, 1972, p. 126)

3.63 <u>Example: Lexical Sets Substitution Drill</u>

Instructions: Respond to the cue using the model "Couldn't you .... on (day+1) instead?"

#1 Cue: I'm seeing him on Tuesday.

Response: Couldn't you see him on Wednesday instead ?

#2 Cue: He's meeting her on Saturday.

Response: Couldn't he meet her on Sunday instead ?

(Cook, 1972, p. 126)

3.64 <u>Example: Lexical Meaning Substitution Drill</u>

Instructions: Respond appropriately to the cue using "How annoying!" or "How nice!"

#1 Cue: It's raining!

Response: How annoying!

#2 Cue: The sun's come out!

Response: How nice!

(Cook, 1972, p. 127)

Appendix A: The Original Drill Sample

3.65 **Example:** Pronoun Substitution Drill

Instructions: Substitute a name for the given pronoun in the model "Oh yes, ... was there."

#1 Cue: I suppose he was there.

Response: Oh yes, John was there.

#2 Cue: I suppose she was there.

Response: Oh yes, Mary was there.

(Cook, 1972, p. 127)

3.66 **Example:** Knowledge Substitution Drill

Instructions: Answer the question using the grammatical pattern ".... did" or " .... was"

#1 Cue: Who wrote Hamlet ?

Response: Shakespeare did.

#2 Cue: Who was Queen Victoria's husband ?

Response: Albert was.

(Cook, 1972, p. 127)

3.68 **Example:** Mechanical Substitution Drill

Instruction: [Insert the cue into the model.]

#1 Model The dictionary you asked for has been stolen.

Cue: Use LOSE

Response/ The dictionary you asked for has been lost.
#2 Model

Cue: Use BORROW

Response: The dictionary you asked for has been borrowed.

(Byrne, 1976, p. 34)

Appendix A: The Original Drill Sample

3.69 Example: Meaningful Practice -1 (Byrne)

Visual Aids: Eight pictures: a bike, a car, a radio, a television, a guitar, a piano, a boat, and a plane.

Instructions: [Make TRUE statements in connection with the above pictures using the models:

"A .... costs more than a ...." or
"It is easier to ..... than to ....."]

#1 Response: A piano costs more than a guitar.

#2 Response: It is easier to ride a bike than to drive a car.

(Byrne, 1976, p. 37)

3.70 Example: Meaningful Drill -2 (Byrne)

Context: Picture of a room full of objects

Instructions: [Cite two objects in the pattern:

"There's a .... near the ...."

so as to make a true utterance.]

#1 Response: There's a bookcase near the door.

(Byrne, 1976, p. 37-38)

3.71 Example: Special Meaningful Drill (Byrne)

Visual Aids: Picture of room full of objects, NOT SHOWN to students.

Instructions: [Ask questions to determine what is in the picture.]

#1 Response: Is there a TV in the room ?

Feedback: Yes

#2 Response: Is there a clock in the room ?

Feedback: Yes

(Byrne, 1976, p. 38-39)

Appendix A: The Original Drill Sample

3.72 Example: Open Ended Responses (Byrne)

Instructions: Complete the following sentences

#1 Model: John was hungry, so ...

Responses: ... he had a sandwich.
... he had some biscuits.
... maybe he had some bread and cheese.
... he had some chocolat.

#2 Model: Mary was thirsty, so ...

Responses: ... maybe she had a cup of tea.
... perhaps she had a glass of water
... she had a glass of beer.

(Byrne, 1976, p. 39-40)

3.73 Example: Imaginary Situations-1

Situation: I've just bought a house. I haven't much money and I'm furnishing it very slowly.

Instructions: Ask me questions about what I have bought, using the pattern "Have you bought a .... yet?"

#1 Response: Have you bought a bed yet?

Feedback: Yes, of course I've bought a bed.

#2 Response: Have you bought a TV set yet?

Feedback: No, I haven't.

(Byrne, 1976, p. 40)

Appendix A: The Original Drill Sample

3.73 Example: Imaginary Situations-2

Situation: I hear that George wants to have a meeting tomorrow. If he rings me, I'll say I'm busy.

Instructions: Make your own excuses:

#1 Cue: [What will you say ?]

Response: If he asks me, I'll say I'm not free.

#2 Cue: [What will you say ?]

Response: If he rings me, I'll say I'm not well.

#3 Cue: [What will you say ?]

Response: If he rings me, I'll say it's too soon.

(Byrne, 1976, p. 40)

3.74 Example: Meaningful Drill-3

Context: Picture of room showing people engaged in various tasks.

Instructions: [Respond to the assertions using the pattern: "No, he/she isn't. He/she's ......"]

#1 Cue: John's reading a book.

Response: No, he isn't. He's watching television.

#2 Cue: Mary's watching television.

Response: No, she isn't. She's doing her homework.

(Byrne, 1976, p. 41)

Appendix A: The Original Drill Sample

3.75 Example: Meaningful Drill-4

Instructions: [Infer from the statement given the correct adjective to place in the model: "You must be ..."]

#1 Cue: I've been working all day.

Response: You must be tired.

#2 Cue: I haven't eaten a thing since breakfast.

Response: You must be hungry.

#3 Cue: I haven't had anything to drink either.

Response: You must be thirsty.

Complication

#4 Cue: And now I'm going to a party.

Response: You must be mad.

(Byrne, 1976, p. 42)

3.76 Example: Get Organized (Concept)

Instructions: Match the term with its description

TERM	DESCRIPTION
#1 a shoplifter	a. spots shoplifters b. pinches things from shops c. has often shoplifted d. steals from the same shop often e. tries to catch shoplifters f. is stealing things from shops
#2 a "regular"	a. --- f.

(Candlin, 1981, p. 67)

Appendix A: The Original Drill Sample

3.77 Example: Getting Organized (Narration)

Situation: Reading: "The Mystery Object Lands"

Instructions: Complete each sentence.

#1 The mystery object landed ...

a. came to rest near an electricity sub-station.
b. between 30 feet and 40 feet across.
c. on Gibbet Hill.
d. sealed the area off from the public.
e. saucer-shaped.
f. that the object was a Russian spaceship.
g. told the police about the mystery object.
h. was closed to normal traffic.
i. examined the "flying saucer".
j. acted on report from a local resident.

#2 The police ...

(Candlin, 1981, p. 68)

3.78 Example: Getting Organized (True/False)

Context: Reading: A dialogue about a flying saucer

Instructions: Decide whether the sentences below are true or false

#1 Cue: There is a thing on Russian Hill.

Response: False

#2 Cue: There is a thing on Gibbet Hill.

Response: True

(Candlin, 1981, p. 70)

Appendix A: The Original Drill Sample

3.79 **Example: Connecting Exercise**
(Getting Organized)

Situation: Twelve connected cartoon scenes with one of the two parts of the dialogue missing.

Instructions: Complete each cartoon dialogue by inserting the appropriate piece of the dialogue shown below.

Cues:

#1 "Couple of chewing gums, 'course you can have it for one."
#2 "No, sir. I just did it."
#3 "The Head smokes."
etc.

(Candlin, 1981, pp. 72-73)

3.80 **Example: Multiple Choice**
(Getting Organized)

Context: Reading: The text of an interview

Instructions: Circle the letter of the correct answer

#1 Model: The police went to see the headmaster because:

Cues:
a. they wanted to play in the team on the estate.
b. the two boys' parents did not pay a fine.
c. two of his boys were caught stealing.
d. Williams, one of his boys, stole a jersey.

(Candlin, 1981, p. 75)

Appendix A: The Original Drill Sample

3.81 **Example: Recognition Exercise (Card Game)**
(Getting Organized)

Context: Audio Aid of horror story, which contains twelve identifiable sounds

Instructions: Listen to the tape. Put the cards into the order of the sounds you hear.

Cards:

Hollow laughter	Scream
Scratching	Fire
Creaking	Whining
Tapping	Guillotine
Footsteps	Hissing
Howl	Dripping

(Candlin, 1[illegible] p. 77-78)

Appendix A: The Original Drill Sample

3.82 **Example: Word Scramble**
(Phase I: Getting Organized)

Situation: The teleprinter at Chapeltown Station isn't working properly. Some information got in which didn't belong to the message. (The learner already knows the details of the message)

Instructions: First cross out the wrong parts, then collect the others and put them in the right order. Compare your message to the original.

Cues: An entire page full of snippets:

green jacket
scar on forehead
Name:
his mother said to the police
5' 6"
etc.

(Candlin, 1981, p. 79 A-11)

3.84 **Computer Example: Memorizing "Cognitive" Language**
(Implanting Skills)

Context: Spring Cleaning Day: Pictures of individuals performing contributory activities.

Instructions: Respond to each question by stating what the person in the picture will do.

#1 Cue: (picture 1)
What about you, Richard?

Response: I'll take the drawing room curtains down.

#2 Cue: (picture 2)
What about Andrew and Harry ?

Response: They'll roll up the carpets.

(Candlin, 1981, p. 84-85)

Appendix A: The Original Drill Sample

3.85   Example: Memorizing "affective" language
(Implanting Skills)

Purpose: To learn how to deny something and stressing the denial

Context: Learner takes the role of someone accused of a crime

Instructions: Deny the following assertions using the model: " XX 's wrong if XX says that, sir, honestly"

#1 Cue: But the store detective says he watched you taking the tool.

Response: He's wrong if he says that, sir, honestly.

#2 Cue: The manager and the detective say you admitted everything.

Response: They're wrong if they say that, sir, honestly.

(Candlin, 1981, p. 86)

3.86   Example: Picture Stimulus
(Implanting Skills)

Context: Dialogue presented via stylized illustrations of what is being said, using a set of established symbols.

Instructions: Say in words what is illustrated in the pictures.

#1 Cue: (Stylized picture of a man looking at a cat)

Response: The man is looking at the cat.
The man sees the cat.
etc.

(Candlin, 1981, p. 88)

Appendix A: The Original Drill Sample

3.87 Example: Restoration exercise-Insertion

Instructions: From the list of verbs provided, find the appropriate verb to insert in each blank in the text. Use the form "has/have just ..."

(List of verbs)

The police ----. They ---- rope barriers, ---- the area, and ---- the road. Hundreds of people ----. Scientists ---- too. They ---- their instruments. The Ministry of Defence ---- that eight other craft ----. The Army ---- several units.

(Candlin, 1981, p. 90)

3.89 Example: Restoration - Catchwords (Implanting Skills)

Instructions: From the key words given below, construct the story of Mr. and Mrs. Shields and their large family.

Mrs. Dora Shields	Mr. Albert Shields, 46
5 children twins	
Brighton General Hospital	boilermaker a bit of a shock
wanted a large family trouble making ends meet	

Appendix A: The Original Drill Sample

3.90 Example: Restoration (Gap Text)
(Implanting Skills)

Context: Paragraph: I came home one night. Fell flat in the dustbins, 'cause I was drunk. And, well ... there's this bloke ... he shouts from the window...etc.

Instructions: Re-tell the story by filling in the blanks.

#1 Model: ["I came home one night. ..."]

Cue: The young man -----

Response: came home.

#2 Model: ["Fell flat in the dustbins,..."]

Cue: He fell ---

Response: in the dustbins

#3 Model: ["...'cause I was drunk. ..."]

Cue: because ---

Response: he was drunk.

(Candlin, 1981, p. 94)

3.91 Example: Restoration (Gap-text/New-text)

Context: Reading: Three dialogue scenes concerning Caroline and Roland.

Instructions: Imagine you are Caroline. Complete the following diary by filling in the blanks based on the dialogues.

CAROLINE'S DIARY

Monday, 25th July

I bought a ------ at the camp shop today and met a smashing boy. He's from ----. His name is -----. etc.

(Candlin, 1981, p. 96)

Appendix A: The Original Drill Sample

3.92 Example: Restoration (Defective Dialogue) (Implanting Skills)

Instructions: Complete the following dialogues using the word indicated.

#1 Cue:

Jack:	Do you like this town?
John:	It bores me stiff.
Jack:	--------- ? (live)
John:	For 3 months.

Response: How long have you been living here?
How long have you lived here?
For how long ...

#2	Peter:	What's wrong with your bicycle?
	Bob:	It's got a flat tyre. I can't find the hole.
	Peter:	--------- ? (look)
	Bob:	For 2 hours.

(Candlin, 1981, p. 104)

3.93.1 Example: Manipulative Drill (Stevick)

Instructions: [Re-state the cue with the verb in the past participle according to the model: "Haven't they .... yet ?"]

#1 Cue: When will they go ?

Response: Haven't they gone yet ?

#2 Cue: When will they leave here ?

Response: Haven't they left here yet ?

#3 Cue: When will they catch the bus ?

Response: Haven't they caught the bus yet ?

(Stevick, 1971, p. 400)

Appendix A: The Original Drill Sample

3.94 Example: Meaningful Drill (Stevick)

Reading: Paragraph about grocery stores

Instructions: [Re-state the cue with the verb in the past participle according to the model: "Haven't they .... yet ?"]

#1 Cue: When will they buy groceries ?

Response: Haven't they bought groceries yet ?

#2 Cue: When will they stock the counter ?

Response: Haven't they stocked the counter yet ?

#3 Cue: When will they display the food ?

Response: Haven't they displayed the food yet ?

(Stevick, 1971, p. 402)

3.95 Example: Application Drill (Meaningful Drill)

Context: Scenes of known characters involved in activities

Instructions: Describe what the character is doing in each scene.

#1 Cue: (Scene of girl sleeping)

Response: Felicity is sleeping.

#2 Cue: (Scene of boy taking bath)

Response: Anthony is having a bath.

(Dakin, 1973, p.64)

Appendix A: The Original Drill Sample

<u>3.96</u> <u>Example:</u> <u>General Knowledge Drill</u>
(Meaningful Drill)

Context: The real world.

Instructions: Make a true sentence about each of the following people using the pattern:

XX plays ......

#1: Cue: Paul McCartney

Response: Paul McCartney plays the guitar.

#2: Cue: Yehudi Menuhin

Response: Yehudi Menuhin plays the violin.

(Dakin, 1973, p.65)

<u>3.97</u> <u>Example:</u> <u>Collocation Drill</u>
(Meaningful Drill)

Instructions: [Respond to each of the statements below using either the model:

Good, I'd like to ..... it. or
Good, I like .....]

#1 Cue: This is a wonderful book.

Response: Good, I'd like to read it.
borrow

Good, I like books.
good books.
wonderful books.

#2 Cue: There's a good film at the cinema this week.

Response: Good, I'd like to see it.
go see
Good, I like films.
good films.

(Dakin, 1973, p.67)

Appendix A: The Original Drill Sample

<u>3.98</u> <u>Example:</u> <u>Synonymy Drill</u>
(Meaningful Drill)

Instructions: [Restate the following sentences using the pattern:

PRONOUN came on/by -----.]

#1 Cue: Father walked here.

Response: He came on foot.

#2 Cue: Mother flew here.

Response: She came by plane.

(Dakin, 1973, p.71)

<u>3.99</u> <u>Example:</u> <u>Hyponomy Drill</u>
(Meaningful Drill)

Instructions: [State the consequences of the situation, using the model:

There was/were so many/much GROUP NOUN that he/she/I couldn't APPROPRIATE VERB it/them all.]

#1 Cue: The old woman who lived in a shoe had a lot of sons and daughters.

Response: There were so many children that she couldn't feed them all.

#2 Cue: For dinner, she gave me a huge amount of rice and chicken.

Response: There was so much food that I couldn't eat it all.

(Dakin, 1973, p.74)

Appendix A: The Original Drill Sample

3.100 Example: Antonymy Drill
(Meaningful Drill)

Instructions: Re-state the following propositions using the model:

There's still a lot of ------.

#1 Cue: Pala isn't a rich country.

Response: There's still a lot of poverty.

#2 Cue: Pala isn't a healthy country.

Response: There's still a lot of disease.

(Dakin, 1973, p.77)

3.101 Example: Converse Drill
(Meaningful Drill)
(Open-ended)

Instructions: Respond to the following questions truthfully.

#1a Cue: Who is sitting on your left ?

Response: Mary is.

#1b Cue: So you ....

Response: So I am sitting on Mary's right.

#2a Cue: Who is sitting on your right ?

Response: Sue is.

#2b Cue: So you ...

Response: So I am sitting on Sue's left.

#3 Cue: Mary and Sue are sitting on either side of you, so you ...

Response: So I am sitting between Mary and Sue.

(Dakin, 1973, p.78)

Appendix A: The Original Drill Sample

<u>3.102</u> <u>Example:</u> <u>Consequence Drill</u>
(Meaningful Drill)

Instructions: Respond to the information using the form: It/he is now -----.

#1 Cue: The cat has been killed.

Response: It is now dead.

#2 Cue: Frankenstein's monster has escaped.

Response: He is now free.

#3 Cue: Harold has got what he wanted.

Response: He is now content.

(Dakin, 1973, p.79)

<u>3.106</u> <u>Original Example:</u> <u>Replacement</u>

Instructions: Replace the direct object by the appropriate pronoun.

#1 Model: I am buying the typwriter

Response: I am buying it.

#2 Model: [I am buying the books.]

Response: [I am buying them.]

(Stack, 1971, p. 144)

<u>3.111</u> <u>Example:</u> <u>Replacement Drill</u>

Instructions: Replace the appropriate phrase with "y"

#1 Model: Allons au cinéma.

Response: Allons-y.

#2 Model: [Allons à la gare.]

Response: [Allons-y.]

(Stack, 1971, p. 145)

Appendix A: The Original Drill Sample

3.114 Original Example: Replacement

Instructions: Replace the number by the next higher number

#1 Model: Paul has twelve records

Response: Paul has thirteen records.

#2 Model: [Sue bought three cars.]

Response: [Sue bought four cars.]

(Stack, 1971, p. 146)

3.115 Example: Replacement (Adverbs of Quantity)

Instructions: Replace the adverb of quantity with "beaucoup de"

Il a peu de livres.
beaucoup de

3.116 Example Transformation Drill

Instructions: Change the verb to the future tense:

#1 Model: He's working at the store.

Response: He'll work at the store

#2 Model: [I'm walking to work.]

Response: [I'll walk to work.]

(Stack, 1971, p. 147)

Appendix A: The Original Drill Sample

<u>3.117</u> <u>Example:</u> <u>Transformation:</u> <u>verb tenses</u>

Instructions: Change to the passé composé.

#1 Model: Alain commande un bon dîner.

Response: Alain a commandé un bon dîner hier.

#2 Model: [Roger parle à ses amis.]

Response: [Roger a parlé à ses amis hier.]

(Stack, 1971, p. 147)

<u>3.118</u> <u>Example:</u> <u>Transformation:</u> <u>directed increment</u>

#1 Model: Si Paul est fatigué...

Cue: Il se repose.

Response: Si Paul est fatigué il se reposera.

#2 Model: [Si Marie est chanceuse...]

Cue: [Elle gagne.]

Response: [Si Marie est chanceuse elle gagnera.]

(Stack, 1971, p. 148)

<u>3.120</u> <u>Example:</u> <u>Transformation:</u> <u>negation</u>

Instructions: Make the statement negative.

#1 Model: They're reading.

Response: They're not reading
[They aren't reading]
[They are not reading]

#2 Model: [He's walking.]

Response: [He's not walking.]
[He isn't walking.]
[He is not walking.]

(Stack, 1971, p. 149)

Appendix A: The Original Drill Sample

3.125 Example: Transformation: Comparisons

Instructions: [Make the following sentences comparative.]

#1 Model: Este reloj es bonito.

Response: Este reloj es más bonito.

#2 Model: [Esta casa es grande.]

Response: [Esta casa es más grande.]

Complication: buen->mejor

(Stack, 1971, p. 154)

3.127 Example: Transformation: word order

Instructions: Begin each sentence with the adverb.

#1 Model: Der Brunnen ist tief.

Response: Tief ist der Brunnen.

#2 Model: [Der Antwort war falsch.]

Response: [Falsch war der Antwort.]

(Stack, 1971, p. 155)

3.129 Example: Question Formation

Instructions: Make a question by inversion

#1 Model: Nous allons en ville.

Response: Allons-nous en ville?

#2 Model: [Vous allez à la campagne.]

Response: [Allez-vous à la campagne?]

(Stack, 1971, p. 157)

Appendix A: The Original Drill Sample

3.130 Example: Chain Transformation Drill

	Basic Utterance:	He doesn't study.
#1	Fragment Cue:	She
	Response:	She doesn't study.
#2	Fragment Cue:	They
	Response:	They don't study.
#3	Slot Change:	read
	Response:	They don't read.
	(Return to Main Point)	
#4	Fragment Cue:	He
	Response:	He doesn't read.

(Stack, 1971, p. 158)

3.131 Example: Paired Sentence: relative pronouns

Instructions: Combine the two short sentences into a single sentence using a relative pronoun.

#1	Model:	The man is a doctor.
	Cue:	I saw the man last night.
	Response:	The man ___ I saw last night is a doctor. that who

(Stack, 1971, p. 131)

Appendix A: The Original Drill Sample

3.132 Example: Paired Sentence: Conditional Sentences

Instructions: Join the following two sentences together using "if..."

#1 Model: The weather is nice. I am going downtown.

Cue: If...

Response: If the weather is nice, I will go downtown
[If the weather were nice, I would go downtown.]
[If the weather is nice, I will be going downtown.]
[If the weather were nice, I would be going downtown.]

(Stack, 1971, p. 163)

3.134 Example: Paired Sentences: tense linkages

Instructions: Combine the two sentences, putting the first verb into the imperfect and the second verb into the passé composé.

#1 Model: Alain étudie. Je rentre.

Response: Alain étudiait quand je suis rentré.

#2 Model: [Marc parle. Roger vient.]

Response: [Marc parlait quand Roger est venu.]

(Stack, 1971, p. 164)

Appendix A: The Original Drill Sample

3.136 Example: Paired Sentence Drill

Instructions: Respond to the question according to the model: "The boy speaks politely"

#1 Model: The boy is polite.

Cue: How does he speak?

Response: The boy speaks politely.

#2 Model: [The girl is careful.]

Cue: [How does she walk?]

Response: [The girl walks carefully.]

(Stack, 1971, p.165)

3.140 Example: Fixed Increment Drill

Instructions: Complete each sentence by adding "hablar español"

#1 Cue: El aprende ...

Response: El aprende a hablar español.

#2 Cue: [El recuerda]

Response: [El recuerda de hablar español.]

#3 Cue: [Le gusta ...]

Response: [Le gusta hablar español.]

(Stack, 1971, p. 169)

Appendix A: The Original Drill Sample

3.142 Example Analogy Drill - Tag Questions

Instructions: Respond in the affirmative using the model "........ too."

#1 Model: I read a lot of books.

Cue: How about Paul?

Response: Paul reads a lot of books too.

#2 Model: [They eat a lot.]

Cue: [How about Carol?]

Response: [Carol eats a lot too.]

(Stack, 1971, p. 171)

3.143 Example Analogy Drill - Tag Questions

Instructions: Apply the adjective used in the first half of each sentence to the noun suggested in the second half.

#1 Model: La pluma es negra.

Cue: Y el lápiz?

Response: El lápiz es negro.

#2 Model: [El profesor es alto]

Cue: [Y la profesora?]

Response: [La profesora es alta]

(Stack, 1971, p. 177)

Appendix A: The Original Drill Sample

3.144 Example: Question Drills: you-I conversion

Instructions: Answer the questions affirmatively, using "I".

#1 Model: Do you like music?

Response: Yes, I like music.
Yes, I do.

#2 Model: Do you attend the Sunday concerts?

Response: Yes, I attend.
Yes, I attend them.
Yes, I attend the Sunday concerts.
Yes, I do.

(Stack, 1971, p. 179)

3.145 Example: Question Drill: narration sequence

Context: (Story: Little Red Riding Hood)

Instructions: Answer the questions truthfully.

#1 Model: Did she like her red cape?

Response: Yes, she liked it.
Yes.
Yes, she did.
Yes, she liked her red cape.
etc.

#2 Model: Did she use the Cadillac?

Response: No, she didn't use it.
No, she didn't.
No.
No, she didn't use the Cadillac.
No, she didn't have a car.
etc.

(Stack, 1971, p. 180)

Appendix A: The Original Drill Sample

1.146 Example: Question Drills: directed answers

Instructions: Answer affirmatively using the cue word.

#1 Model: Does Robert study his lessons?

Cue: always.

Response: Yes, he always studies his lessons.

#2 Model: Does George work on Sundays?

Cue: sometimes

Response: Yes, he sometimes works on Sundays.

(Stack, 1971, p. 181)

3.150 Example: Non-Communicative Interchange

Instructions: Change the sentence from present to past tense

Model: He writes a lot of essays.

Cue: He wrote a lot of essays.

(Beile, 1979, p. 115)

3.151 Example: Isolated Communicative Interchanges

Instructions: Ask a yes/no question based on the statement given.

#1 Model: I've always wanted to run a garage.

Response: Have you ?

#2 Model: Mr. Sharp's talking to the directors this afternoon.

Response: Is he ?

(Beile, 1979, p. 115)

Appendix A: The Original Drill Sample

<u>3.152</u> <u>Example:</u> <u>Isolated communicative interchanges in a situational context</u>

Context: John is trying to carry on a conversation with Colin.

Instructions: But whatever he says, Colin only gives a polite reply:

#1 Model: I've always wanted to run a garage.

Response: Have you ?

#2 Model: Mr. Sharp's talking to the directors this afternoon.

Response: Is he ?

(Beile, 1979, p. 116)

<u>3.153</u> <u>Example:</u> <u>Connected communicative interchanges</u>

#1 Cue: Ask Helen whether she likes playing tennis.

Response: Do you like playing tennis, Helen ?

#2 Cue: Ask Thomas whether he enjoys listening to classical music

Response: Do you enjoy listening to classical music, Thomas ?

(Beile, 1979, p. 116)

Appendix A: The Original Drill Sample

3.154 **Example:** Connected communicative interchanges in a situational context

Context: Linda's ill. She's in bed. Mrs. Scott has got some hot tea for her.

Instructions: Take Linda's part. You don't want anything.

#1 Model: I've got some hot tea for you, Linda.

Response: I don't want any hot tea.

#2 Model: Then have some milk.

Response: I don't want any milk.

(Beile, 1979, p. 117)

3.155 **Example:** No Contextualisation (Non Communicative)

Instruction: [Describe what the person sells, using the model: The XXXr XXXes object.]

#1 Cue: baker

Response: The baker bakes bread.

#2 Cue: greengrocer

Response: The greengrocer sells fruit and vegetables.

(Beile, 1979, p.121)

3.156 **Example:** Dialog-like: Not Contextualisable

*Instruction: Answer the questions

#1 Model: What does the baker do ?

Response: The baker bakes bread.

#2 Model: What does the greengrocer do ?

Response: The greengrocer sells fruit and vegetables.

(Beile, 1979, p. 121)

Appendix A: The Original Drill Sample

3.157 Example: Contextualisable

*Instruction: Answer the question by stating that you "do" your own.

#1 Model: Where do you buy your bread ?

Response: We bake our own bread.

Model: Which greengrocer to you go to ?

Response: We grow our own fruit and vegetables.

(Beile, 1979, p. 121)

3.158 Example: Manipulative Drill

Instructions: Place the words in the correct places.

#1 Model: We have ---- milk, but we don't have ---- cream.

Cue: some/any

Response: We have some milk, but we don't have any cream.

#2 Model: We don't have ---- milk, but we have ---- cream.

Cue: some/any

Response: We don't have any milk, but we have some cream.

(Robinett, 1978, p. 210)

Appendix A: The Original Drill Sample

3.159 Example: Communication Drill

Instructions: Choose the correct word for each case

#1 Model: I'd like ---- eggs, please.

Cue: some/any

Response: I'd like some eggs, please.

#2 Model: I'm sorry, there aren't ---- left.

Cue: some/any

Response: I'm sorry, there aren't any left.

#3 Model: Do you have ---- milk or ---- cream ?

Cue: some/any

Response: Do you have some milk or some cream ?
any any

(Robinett, 1978, p. 211)

3.160 Example: Meaningful Drill-1 (unpatterned)

Context: A narration describing Robert's visit to a restaurant.

Instructions: Respond to the questions truthfully.

#1 Model: Where did Robert go ?

Response: He went to a restaurant.

#2 Model: When did he arrive ?

Response: He arrived at seven-thirty.

#3 Model: Where did he find a table ?

Response: He found a table near the window.

(Stack, 1971, p. 136)

Appendix A: The Original Drill Sample

3.161 **Example: Meaningful Drill-2** (patterned)

Instructions: Respond to the situations appropriately using the model: "You must ..."

#1 Cue: My pencil-point is broken.

Response: You must sharpen it.

#2 Cue: My hair is too long.

Response: You must get it cut.

#3 Cue: My phone is ringing.

Response: You must answer it.

(Stack, 1971, p. 137)

Appendix B: Table 3.8

Summary of Operational Classification based on Cue and Response Model

Simple Operations: Only one discrete operation to perform

(Type of Operation: Final goal of the operation is...)

I. Insertions into model

A. Insertion of cue into model: C: --> M:[]

1.	3.4	Dacanay	Simple Substitution
3.	3.6	Dacanay	Moving Slot Substitution
6.	3.9	Dacanay	Transposition Transformation
7.	3.10	Dacanay	Expansion Transformation
8.	3.11	Dacanay	Expansion Transformation
16.	3.21	Rivers	Double Substitution
34.	3.40	Robinett	Simple Substitution
36.	3.42	Robinett	Moving Slot Substitution
44.	3.60	Cook	Plain Substitution
45.	3.61	Cook	Sequence Substitution
56.	3.118	Stack	Transformation: directed increment
57.	3.120	Stack	Transformation: negation
58.	3.125	Stack	Transformation: comparisons
69.	3.144	Stack	Question Drill: you-I conversion

B. Insertion of secondary element into model E: --> M:C: []

2.	3.5	Dacanay	Correlative Substitution
28.	3.34	Rivers	Type A1 Completion
29.	3.35	Rivers	Type A2 Completion
35.	3.41	Robinett	Correlative Substitution
55.	3.117	Stack	Transformation: verb tenses
63.	3.132	Stack	Paired Sentence: Conditionals
66.	3.140	Stack	Fixed Increment
67.	3.142	Stack	Analogy Drill: tag questions
68.	3.143	Stack	Analogy Drill: tag questions

C. Insertion of element triggered by cue C:-->E:-->M:[]

15.	3.20	Rivers	Simple Substitution
21.	3.28	Rivers	General Conversion
22.	3.29	Rivers	General Conversion
40.	3.46	Robinett	Reduction Transformation
41.	3.47	Robinett	Response Transformation
46.	3.62	Cook	Lexical Pair Substitution
47.	3.63	Cook	Lexical Sets Substitution

48.	3.64	Cook	Lexical Meaning Substitution
49.	3.65	Cook	Pronoun Substitution
50.	3.66	Cook	Knowledge Substitution
51.	3.106	Stack	Replacement
52.	3.111	Stack	Replacement
54.	3.114	Stack	Replacement

D. Insertion of element triggered by deletion of cue   M:+C: --> M:-- + [] <--E:

27.	3.33	Rivers	Type A Deletion

E. Free Insertion   E: --> M:[]

26.	3.32	Rivers	Type B Expansion
30.	3.36	Rivers	Type B Completion

II. Removal of redundant elements   M:+Rd: --> M:--

10.	3.13	Dacanay	Reduction Transformation
23.	3.30	Rivers	Combination Conversion

III. Re-order of model elements   M:[1][2] --> M:[2][1]

4.	3.7	Dacanay	Transposition Transformation
5.	3.8	Dacanay	Transposition Transformation
11.	3.14	Dacanay	Integration Transformation
59.	3.127	Stack	Transformation: word order
60.	3.129	Stack	Question Formation

IV. Production of new elements, based on model C:-->R:-->(M:)

24.	3.31A	Rivers	Restatement Conversion
31.	3.37	Rivers	Question-Answer Practice
32.	3.38	Rivers	Question-Answer Practice
33.	3.39	Rivers	Rejoinder Exercise
42.	3.54	Cook	Typical Drill
70.	3.145	Stack	Question Drill: narration sequence

Complex Operations: The application of more than one simple operation required

MULTIPLE OPERATIONS: More than one discrete operation of the same type

I. Insertion + Insertion, prompted by the same cue

9.	3.12	Dacanay	Transposition & Expansion
17.	3.22	Rivers	Correlative Substitution
64.	3.134	Stack	Paired Sentence: tense linkages

Appendix B Table 3.8

II. Insertion of cue, then Insertion, prompted by cue

18. 3.24 Rivers Multiple Substitution
19. 3.26 Rivers General Conversion
25. 3.31B Rivers Restatement Conversion
37. 3.43 Robinett Moving Slot & Correlative Substitution
65. 3.136 Stack Paired Sentence

III. Re-order, then Re-order

12. 3.15 Dacanay Integration + Transformation

MIXED OPERATIONS: More than one discrete operation of different types

I. Insertion ± Removal

10. 3.13 Dacanay Reduction Tranformation
14. 3.16 Dacanay Integration Transformation with Reduction
23. 3.30 Rivers Combination Conversion

II. Insertion, then Removal, then Insertion

62. 3.131 Stack Paired Sentence: relative pronouns

III. Re-order, then production of new element

38. 3.44 Robinett Transposition Transformation
39. 3.45 Robinett Integration Transformation

IV. Production of new elements, then insertion

71. 3.146 Stack Question Drill: directed answers

CHOICE OF TEMPLATE

I. Insertion or Re-order

20. 3.27 Rivers General Conversion

II. Insertion (Type-A) or Insertion (Type-C)

61. 3.130 Stack Chain Transformation

Appendix B Table 3.9

Detailed Analysis and Categorization of Drill Examples according to Cue and Response-Model:

1. This table contains excerts from the original drills listed in Appendix A. For a complete understanding of the nature of a drill example, reference should be made to the original.

2. For a complete understanding of this table, reference should be made to section 3.7 of the text.

3. Symbols and conventions used in this table:

C: The cue (where the cue is presented as part of a cue-framework, this is noted. Only the raw cue is shown here, however)

M: The response-model

C:+M: The response-model, with the cue imbedded within it

R: The learner's response, in cases where there is no explicit response-model

Rd: Redundant elements which must be removed from the response-model

M:+Rd: The response-model, containing redundant elements.

[] An insertion-slot in the response-model.

[1] Positional slots

[xxx] An insertion-slot in the response-model, containing the original element which the learner must replace with a new element

xxx a. A cue imbedded within the response-model.
b. Redundant elements within the response-model

----complication A situation to be considered in outside of this table

--> An action of some sort. For example:
C: ---> M: refers to the action of inserting the cue into the model

Appendix B: Table 3.9

Part One: Simple Operations: Only one discrete operation to perform

(Type of Operation: Final goal of the operation is...)

I. Insertions into model

A. Insertion of cue into model: C: --> M:[]

1. 3.4 Dacanay Simple Substitution

C: away --> M: I'm going []
C: later --> M: I'm going []

3. 3.6 Dacanay Moving Slot Substitution

C: tonight --> M: The [] leaves [] at seven.
C: bus --> M: The [] leaves [] at seven.

6. 3.9 Dacanay Transposition Transformation

C: for me --> M: Mother made []'a dress [].
C: to me --> M: My friend gave [] the pen [].

7. 3.10 Dacanay Expansion Transformation

C: two --> M: The [] napkins are on the table.
C: paper --> M: The [] two [] napkins are on the table.
C: pretty --> M: The [] two [] paper [] napkins are on the table.
C: yellow --> M: The [] two [] pretty [] paper [] napkins are on the table.

8. 3.11 Dacanay Expansion Transformation

C: always --> M: He [] 's [] on time [].
C: always --> M: I [] write [] to them [].
C: regularly --> M: I [] write [] to them [].
C: now --> M: He [] 's [] leaving [].

16. 3.21 Rivers Double Substitution

C: If you want it --> M: [If I find it] [I'll give it to you].
C: he'll sell it to you --> M: [If you want it] [I'll give it to you].

Appendix B: Table 3.9

34. 3.40 Robinett Simple Substitution

C: pen --> M: The [] is on the table.
C: box --> M: The [] is on the table.

36. 3.42 Robinett Moving Slot Substitution

C: he --> M: [She] [bought] a [car] [yesterday].
C: house --> M: [He] [bought] a [car] [yesterday].
C: last week --> M: [He] [bought] a [house] [yesterday].
C: sold --> M: [He] [bought] a [house] [last week].

44. 3.60 Cook Plain Substitution

C: whisky --> M: [I] [love] [ ].
C: hate --> M: [I] [love] [whisky].
--complication
\# Correlative change to other element

C: He -->1. M: [I] [hate] [whisky].
2. M: He hate[] whisky.

45. 3.61 Cook Sequence Substitution

\# Isolation of cue from cue framework.
--complication
C: skating --> M: Oh, I prefer [].
C: walking --> M: Oh, I prefer [].

56. 3.118 Stack Transformation: directed increment

C: il se repose --> M: Si Paul est fatigué [].
C: elle gagne --> M: Si Marie est chanceuse [].

--complication
\# cue changes in form to conform to model

57. 3.120 Stack Transformation: negation.

\# Formalized Drill: cue implicit
--complication
C: not --> M: They [] reading.
C: not --> M: He [] walking.
--complication
\# Form of cue can change: not/n't

Appendix B: Table 3.9

58. 3.125 Stack Transformation: comparisons

Formalized drill: Cue is implicit

--complication

C: más M: Este reloj es [] bonito.
C: más M: Esta casa es [] grande.

69. 3.144 Stack Question Drill: you-I conversion

*Isolation of cue from cue framework

--complication

C: like music --> M: Yes, I []
C: attend the Sunday concerts --> M: Yes, I []

--complication

* Form of cue can change: ... do/attend/attend them.

B: Insertion of secondary element into model E: --> M:C: []

(Cue is inserted, triggering correlated change effected by insertion of secondary element in another position in model.)

2. 3.5 Dacanay Correlative Substitution

C:+M: Where [] Jose live?
C:+M: Where [] the boys live?

28. 3.34 Rivers Type A1 Completion

*Isolation of cue from cue framework

--complication

C:+M: If I saw him I [] tell him.
C:+M: If you took it I [] tell him.
C:+M: If she comes I [] tell him.

29. 3.35 Rivers Type A2 Completion

*Isolation of cue from cue framework

--complication

C:+M: A person who builds houses is a [].
C:+M: A person who sells meat is a []

35. 3.41 Robinett Correlative Substitution

C:+M: Mary and John [] studying.
C:+M: My brother [] studying.

Appendix B: Table 3.9

55. 3.117 Stack Transformation: verb tenses

*Partially Formalized Drill

---complication

C:+M: Alain [] un bon dîner hier.
C:+M: Roger [] à ses amis hier.

63. 3.132 Stack Paired Sentence: conditionals

C:+M: If the weather is nice, I [] go downtown.
C:+M: If the weather were nice, I [] downtown.

66. 3.140 Stack Fixed Increment

*Isolation of cue from cue framework

---complication

C:+M: El aprende [] hablar español.
C:+M: El recuerda [] hablar español.
C:+M: Le gusta [] hablar español.

67. 3.142 Stack Analogy Drill: tag questions

*Isolation of cue from cue framework

---complication

C:+M: Paul [] a lot of books too.
C:+M: Carol [] a lot too.

68. 3.143 Stack Analogy Drill: tag questions

*Isolation of cue from cue framework

---complication

C:+M: El lápiz es [].
C:+M: La profesora es [].

C. Insertion of element triggered by cue C:-->E:-->M:[]

(Cue itself is not inserted. Cue triggers choice of another element which is inserted in its place.)

15. 3.20 Rivers Simple Substitution

*Isolation of Cue from cue framework

---complication

C: uncle M: Yes, [] 's across the road.
C: sister M: Yes, []-'s across the road.

Appendix B: Table 3.9

21. 3.28 Rivers General Conversion

C:her mother M:Janet read [] the letter [].
C:the letter M:Janet read [] to her.
--complication
her mother -->"her" in one location
"to her" in another location

22. 3.29 Rivers General Conversion

C:it M:I gave [] to her.
C:her M:I gave [] a car.
--complication
a. Choice of model:
I gave [] to her/I gave her []
I gave [] a car/ I gave a car []
b. Model dependent on previous item

40. 3.46 Robinett Reduction Transformation

*Isolation of cue from cue framework
--complication
C: mow the lawn M: She's going to have []
C: fix the sidewalk M: She's going to have []

41. 3.47 Robinett Response Transformation

*Isolation of Cue from cue framework
--complication
C: chairs M: Yes, just a [], please.
C: help M: Yes, just a [], please.

46. 3.62 Cook Lexical Pair Substitution

*Isolation of Cue from cue framework
--complication
C: young M: No, he's [].
C: rich M: No, he's [].

47. 3.63 Cook Lexical Sets Substitution

*Isolation of Cue from cue framework
--complication
C: Tuesday M: Couldn't you see him on [] instead?
C: Sunday M: Couldn't you see him on [] instead?

48. 3.64 Cook Lexical Meaning Substitution

C: It's raining M: How [] !
C: The Sun's come out! M: How [] !

49. 3.65 Cook Pronoun Substitution

*Isolation of Cue from cue framework
--complication

C: he M: Oh yes, [] was there.
C: she M: Oh yes, [] was there.

50. 3.66 Cook Knowledge Substitution

C: Who wrote Hamlet? M: [] did.
C: Who was Queen Victoria's husband? M: [] was.

51. 3.106 Stack Replacement

*Isolation of Cue from cue framework
--complication

C: typewriter M: I am buying [].
C: the books M: I am buying [].

52. 3.111 Stack Replacement

*Formalized Drill
--complication

C: au cinéma M: Allons [].
C: à la gare M: Allons [].

54. 3.114 Stack Replacement

*Isolation of Cue from cue framework
--complication

C: twelve M: Paul has [] records.
C: three M: She bought [] cars.

Appendix B: Table 3.9

D. Insertion of element triggered by deletion of cue. M:+C: --> M:-- + [] <--E:

(Cue is deleted, triggering correlative change effected by insertion of secondary element in model.)

27. 3.33 Rivers Type A Deletion

Formalized drill: Cue is implicit

---complication

C: n't M: They have--- [] coffee.
C: n't M: She did-- []

E. Free Insertion E: --> M:[]

26. 3.32 Rivers Type B Expansion

M: The [] man crosses the [] street [].

30. 3.36 Rivers Type B Completion

M: She has decided []
M: You needn't []

II. Removal of redundant elements M:+Rd: --> M:--

Not Present in Simple Operations

10. 3.13 Dacanay Reduction Tranformation

1. C: --> M:[]

C: noisy
M: The [] children [] who were making a lot of noise...

2. M: The noisy children who were making a lot of noise ...
R: The noisy children ...

23. 3.30 Rivers Combination Conversion

1. C: Give me the keys + I left you the keys.
M: Give me the keys [] I left you the keys.

C: Don't close the door + The door has just been painted.
M: Don't close the door [] the door has just been painted.

Appendix B: Table 3.9

III. Re-order of model elements M:[1][2] --> M:[2][1]

4. 3.7 Dacanay Transposition Transformation

M: [The girl] [is] [ready]. --> [2] [1] [3]?
[1] [2] [3]
M: [The boy] [is] [coming].--> [2] [1] [3]?

5. 3.8 Dacanay Transposition Transformation

M: [This morning] [he was here]. --> [2] [1].
[1] [2]
M: [Occasionally] [he comes here]. --> [2] [1].

11. 3.14 Dacanay Integration Transformation

M: [Who made the report to the principal?] [Do you know?]
[1] [2]
--> [2] [1]

--complication

*other interpretations possible

59. 3.127 Stack Transformation: word order

M: [Der Brunnen] [ist] [tief]. --> [3] [2] [1].
[1] [2] [3]
M: [Der Antwort] [war] [falsch]. --> [3] [2] [1].

60. 3.129 Stack Question Formation

M: [Nous] [allons] en ville. --> [2] [1] en ville.
[1] [2]
M: [Vous] [allez] à la campagne --> [2] [1] a la ...

IV. Production of new elements, based on cue C:-->R:-->(M:)

24. 3.31A Rivers Restatement Conversion

C: Tell George your name is Ronald.
R: George, my name is Ronald.
C: Ask Alice where she's going.
R: Where are you going, Alice?
C: Ask her to wait for you.
R: Wait, please.
Alice, wait!
Alice, wait up!
etc.

Appendix B: Table 3.9

31. 3.37 Rivers Question-Answer Practice

C: Why didn't they come here before midnight?
R: They didn't come because there...
C: Do you often stay out late at night?
R: No, I go home early because I ...

33. 3.39 Rivers Rejoinder Exercise

C: Didn't get the job. Just got there as ...
R: Tough luck!
Too bad!
Sorry about that!
C: See you at the bus stop.
R: Ok.
Okay.
Sure.
Sorry, can't make it.

42. 3.54 Cook Typical Drill

Formalized Drill: Negative answers

C: Is Bill playing tennis tonight?
R: No, he's not (going to play).
No, he isn't (going to play).
Bill ...
C: Is Susan helping her mother this evening?
R: No, she's not going to help.
No, she's not helping her.
etc.

70. 3.145 Stack Question Drill: narration sequence

C: Did she like her red cape?
R: Yes.
Yes, she did.
Yes, she liked it.
Yes, she liked her red cape.

32. 3.38 Rivers Question-Answer Practice

C: Does it live on a farm?
R: No (mediated)
---complication
C: Give me a question to distinguish a rhinoceros from an elephant.
R: Does it have a horn? (unmediated)

Appendix B: Table 3.9

Part Two: Complex Operations: More than one simple operation

MULTIPLE OPERATIONS: More than one discrete operation to perform

I. Insertion + Insertion, prompted by the same cue

9. 3.12 Dacanay Transposition & Expansion

1. Insertion of secondary element into model: E:--> M:C:[]

*Formalized cue = not-->n't

C:+M: He [] n't [like] coffee.
a. b.

17. 3.22 Rivers Correlative Substitution

1. Insertion of secondary element into model: E:--> M:C:[]

C:+M: You [] [] lunch.
a. b.

64. 3.134 Stack Paired Sentence; tense linkages

*Partially formalized Cue:

C: a. [illegible] b. rentre
M: Alain [] quand je suis [].
a. b.

II. Insertion of cue, then Insertion, prompted by cue

18. 3.24 Rivers Multiple Substitution

1. Insertion of primary element into model: C:--> M:[]

C:Peter
M:[She] brings too many [pencils] to [school].

2. Insertion of secondary element into model: E:-->M:C:[]

C:+M: Peter [brings] [too many] pencils to school.
a. -----> <----------b.

Appendix B: Table 3.9

19. 3.26 Rivers General Conversion

*formalized cue: Do/Does

1. Insertion of primary element into model: C:--> M:[]

C:Do/Does
M:[] Peter has a new car.

2. Insertion of secondary element into model: E:-->M:C:[]

C:+M: Does Peter [has] a new car.
----->

25. 3.31B Rivers Restatement Conversion

*Formalized Drill

1. Insertion of cue into model: C:-->M:[]

C:she
M:She said [] [] just arrived but [] [] leaving in a few minutes
1. 1.

2. Insertion of secondary element into model: E:-->M:C:[]

C:+M: She said she [] just arrived but 0 [] leaving in a few minutes.

37. 3.43 Robinett Moving Slot & Correlative Substitution

1. Insertion of primary element into model: C:--> M:[]

C:yesterday
M:[She] is buying a car [today].

2. Insertion of secondary element into model: E:-->M:C:[]

C:+M: She [is buying] a car yesterday.
----> <----------------

65. 3.136 Stack Paired Sentence

1. Insertion of primary element into model: C:--> M:[]

C:speaks
M:The boy [i[illegible]

2. Insertion of secondary element into model: E:-->M:C1[]

C:+M: The boy speaks [polite].
----->

III. Re-order, then Re-order

12. 3.15 Dacanay Integration & Transposition

*Formalized cue

1. Re-order of model elements: M:[1][2]-->M:[1][2]

M: [1] Who is he?
[2] Do you know?

2. Re-order of model elements: M:[1][2]-->M:[1][2]

M: Do you know who [is] [he]-->Do you know [2][1]
[1] [2]

MIXED OPERATIONS

I. Insertion + Removal

10. 3.13 Dacanay Reduction Tranformation

1. Insertion of cue into model: C:-->M:[]

C: noisy
M: The [] children [] who were making a lot of noise were sent to bed.

2. Removal of redundant elements: M:+Rd:-->M:--

M:+Rd: The noisy children who were making a lot of noise were sent to bed.
M:-- The noisy children were sent to bed.

14. 3.16 Dacanay Integration Transformation with Reduction

* Formalized Drill

1. Removal of redundant elements: M:+Rd:-->M:--

M:+Rd: Mary is sick. She can't study her lessons.

M:-- Mary is -- sick -- study her lessons.

2. Insertion of cue into model: C:-->M:[]

C: to,too
M: Mary is [] sick [] study her lessons.

23. 3.30 Rivers Combination Conversion

1. Removal of redundant elements: M:+Rd:-->M:--

M:+Rd: Don't close the door the door has just been painted

M:-- Don't close the door has just been painted.

2. Insertion of cue into model: C:-->M:[]

C: that
M: Don't close the door [] has just been painted.

II. Insertion, then Removal, then Insertion

62. 3.131 Stack Paired Sentence: relative pronouns

1. Insertion of cue into model: C:-->M:[]

C: I saw the man last night.
M: The man [] is [] a [] doctor [].

2. Removal of redundant elements: M:+Rd:-->M:--

M:+Rd: The man I saw the man last night is a doctor.

M:-- The man I saw last night is a doctor.

3. Insertion of secondary element into model: E:-->M:C:[]

C: that/who
M: The man [] I saw last night is a doctor.

Appendix B: Table 3.9

III. Re-order, then production of new element

38. 3.44 Robinett Transposition Transformation

*Formalized drill

1. Re-order of model elements: M:[1][2]-->M:[1][2]

M: [She] [can] swim. --> [2][1] swim?
[1] [2]

2. Production of new element, based on model.

C: Can she swim?
R: Yes, she can.
No, she can't.

39. 3.45 Robinett Integration Transformation

*Formalized cue

1. Re-order, then Re-order

a. Re-order of model elements: M:[1][2]-->M:[1][2]

M: [1] Who is she?
[2] Do you know?

b. Re-order of model elements: M:[1][2]-->M:[1][2]

M: Do you know who [is] [she]-->Do you know [2][1]
[1] [2]

2. Production of new element, based on model.

C: Do you know who she is?
R: Yes, I do.
No, I don't.

IV. Production of new elements, then insertion

71. 3.146 Stack Question Drill: directed answers

1. Production of new elements based on model: M:-->R:

M: Does Robert study his lessons?
R: Yes, he studies his lessons.

2. Insertion of cue into model: C:-->M:[]

C: always
M: Yes, he [] studies [] his lessons [].

CHOICE OF TEMPLATE

I. Insertion or Re-order

20. 3.27 Rivers General Conversion

A. Re-order of model elements: M:[1][2]-->M:[1][2]

M: [John and I] [are] sitting in the classroom.
[1] [2] --> [2][1]

B. Insertion of cue, then Insertion, prompted by cue

1. Insertion of primary element into model: C:--> M:[]

C:Do/Does
M:[] the actress lives in Canada?

2. Insertion of secondary element into model: E:-->M:C:[]

C:+M: Does the actress [live] in Canada?
----->

II. Insertion (type A) or Insertion (type C)

61. 3.130 Stack Chain Transformation

A. Insertion of secondary element into model: E:-->M:C:[]

C:+M: She [] study.

B. Insertion of cue into model: C:-->M:[]

C:read
M:They don't []

Appendix B: Table 3.10

Insertions compared with regard to fixed or variable insertion-slot

1. This table should be interpreted with regard to section 3.7.2 of the text

2. Symbols and conventions used in this table:

* indicates the category to which the drill example has been assigned.

? indicates that, under certain conditions, the example might be categorized differently.

*/ pertains to the first operation, where there are two operations to be performed.

/* pertains to the second operation.

Fxd	Chc	SIMPLE OPERATIONS			
		I. Insertions			
		A. C: --> M:[]			
*		1.	3.4	Dacanay	Simple Substitution
	*	3.	3.6	Dacanay	Moving Slot Substitution
	*	6.	3.9	Dacanay	Transposition Transformation
	*	7.	3.10	Dacanay	Expansion Transformation
	*	8.	3.11	Dacanay	Expansion Transformation
	*	16.	3.21	Rivers	Double Substitution
*		34.	3.40	Robinett	Simple Substitution
	*	36.	3.42	Robinett	Moving Slot Substitution
	*	44.	3.60	Cook	Plain Substitution
*		45.	3.61	Cook	Sequence Substitution
*		56.	3.118	Stack	Transformation: directed increment
*		58.	3.125	Stack	Transformation: comparisons
*		57.	3.120	Stack	Transformation: negation
*		69.	3.144	Stack	Question Drill: you-I conversion

Appendix B: Table 3.10

B. E: --> M:C: []

•		2.	3.5	Dacanay	Correlative Substitution
•		28.	3.34	Rivers	Type A1 Completion
•		29.	3.35	Rivers	Type A2 Completion
•		35.	3.41	Robinett	Correlative Substitution
•		55.	3.117	Stack	Transformation: verb tenses
•		63.	3.132	Stack	Paired Sentence: conditionals
•		66.	3.140	Stack	Fixed Increment
•		67.	3.142	Stack	Analogy Drill: tag questions
•		68.	3.143	Stack	Analogy Drill: tag questions

C. C:-->E:-->M:[]

•		15.	3.20	Rivers	Simple Substitution
•	?	21.	3.28	Rivers	General Conversion
•	?	22.	3.29	Rivers	General Conversion
•		40.	3.46	Robinett	Reduction Transformation
•		41.	3.47	Robinett	Response Transformation
•		46.	3.62	Cook	Lexical Pair Substitution
•		47.	3.63	Cook	Lexical Sets Substitution
•		48.	3.64	Cook	Lexical Meaning Substitution
•		49.	3.65	Cook	Pronoun Substitution
•		50.	3.66	Cook	Knowledge Substitution
•		51.	3.106	Stack	Replacement
•		52.	3.111	Stack	Replacement
•		54.	3.114	Stack	Replacement

D. M:+C: --> M:--+ [] <--E:

•	27.	3.33	Rivers	Type A Deletion

E. E: --> M:[]

•	26.	3.32	Rivers	Type B Expansion
•	30.	3.36	Rivers	Type B Completion

MULTIPLE OPERATIONS

I. Insertion + Insertion

(based on likelyhood of someone making slot error on insertion)

•	9.	3.12	Dacanay	Transposition & Expansion
•	17.	3.22	Rivers	Correlative Substitution
•	64.	3.134	Stack	Paired Sentence: tense linkages

Appendix B: Table 3.10

II. Insertion of cue, then Insertion

/* */	18.	3.24	Rivers	Multiple Substitution
/	19.	3.26	Rivers	General Conversion
/	25.	3.31B	Rivers	Restatement Conversion
/* */	37.	3.43	Robinett	Moving Slot & Correlative Substitution
/	65.	3.136	Stack	Paired Sentence

MIXED OPERATIONS

I. Insertion + Removal

*	10.	3.13	Dacanay	Reduction Tranformation
*	14.	3.16	Dacanay	Integration Transformation with Reduction
*	23.	3.30	Rivers	Combination Conversion

II. Insertion, then Removal, then Insertion

/* */	62.	3.131	Stack	Paired Sentence: relative pronouns

IV. Production of new elements, then insertion

*	71.	3.146	Stack	Question Drill: directed answers

CHOICE OF TEMPLATE

I. Insertion or Re-order

*	20.	3.27	Rivers	General Conversion

II. Insertion (type A) or Insertion (type C)

/	61.	3.130	Stack	Chain Transformation

Appendix B: Table 3.11 Variations in Cue Presentation

1. This table should be interpreted with reference to section 3.7.3 of the text.

2. Symbols and conventions adopted in this table (Items glossed in previous tables are not repeated):

Cf: A formalized cue

P: A separate prompt, in addition to the cue

---

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
							Simple Operations
							I. Insertions into model
							A. C: --> M:[]
		*					1. 3.4 Dacanay Simple Substitution C: away,later...
		*		*			3. 3.6 Dacanay Moving Slot Substitution P: The [train] leaves [tomorrow] at seven. The [train] leaves [tonight] at seven. C: tonight, bus...
		*					6. 3.9 Dacanay Transposition Transformation C: for me, to me...
		*					7. 3.10 Dacanay Expansion Transformation C: two,paper,pretty,yellow...
		*					8. 3.11 Dacanay Expansion Transformation C: always,always,regularly,now...
		*					16. 3.21 Rivers Double Substitution C: If you want it, he'll sell it to you...
		*					34. 3.40 Robinett Simple Substitution C: pen,box...

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
			*				36. 3.42 Robinett Moving Slot Substitution C: he,house,last week,sold...
			*				44. 3.60 Cook Plain Substitution C: whisky, hate...
				*	*		45. 3.61 Cook Sequence Substitution Cf: "use second element" C: I can't decide whether I like swimming or <u>skating</u> best. C: I can't decide whether I like dancing or <u>walking</u> best.
			*				56. 3.118 Stack Transformation: directed increment C: il se repose, elle gagne...
*						*	57. 3.120 Stack Transformation: negation P: They're reading. He's walking Cf: "make negative" --> not
*							58. 3.125 Stack Transformation: comparisons Cf: "make comparative" --> mas
				*			69. 3.144 Stack Question Drill: you-I conversion C: Do you <u>like music</u>? C: Do you <u>attend the Sunday concerts</u>?
							B. E: --> M:<u>C:</u> []
		*			?	*	2. 3.5 Dacanay Correlative Substitution P: Where [does] Mary live? --> interlocked C:+M: Where [] <u>José</u> live? Where [] <u>the boys</u> live?
		*			?	*	28. 3.34 Rivers Type A1 Completion P: If I see him I [will] tell him. C:+M: If I <u>saw</u> him I [] tell him. If you <u>took</u> it I [] tell him.
		*					29. 3.35 Rivers Type A2 Completion C:+M A person who <u>builds houses</u> is a []. A person who <u>sells meat</u> is a [].

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	Drill
		*				*	35. 3.41 Robinett Correlative Substitution P: Mary [is] studying. --> interlocked C:+M: Mary and John [] studying. My brother [] studying.
		*			*	*	55. 3.117 Stack Transformation: verb tenses P: Alain [commande] un bon diner. Roger [parle] à ses amis. Cf: "Use passé composé" C:+M: Alain [] un bon diner hier. Roger [] à ses amis hier.
		*			*	?	63. 3.132 Stack Paired Sentence: conditionals Cf: "Use a form of "will" C:+M: If the weather is nice, I [] go downtown. If the weather were nice, I [] go downtown.
		*					66. 3.140 Stack Fixed Increment C:+M: El aprende [] hablar español. El recuerda [] hablar español. Le gusta [] hablar español.
		*				*	67. 3.142 Stack Analogy Drill: tag questions P: I [read] a lot of books. How about Paul? They [eat] a lot. How about Carol? C:+M: Paul [] a lot of books too. Carol [] a lot too.
		*				*	68. 3.143 Stack Analogy Drill: tag questions P: La pluma es [negra]. Y el lápiz? El profesor es [alto]. Y la profesora? C:+M: El lápiz es []. la profesora es [].

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
							C. C:-->E:-->M:[]
			*				15. 3.20 Rivers Simple Substitution C: Do you see my uncle over there? Do you see my sister over there?
			*				21. 3.28 Rivers General Conversion C: Janet read her mother the letter. Janet read the letter to her. (interlocked)
			*				22. 3.29 Rivers General Conversion C: I gave it to her. I gave her a car. (must be generated)
			*				40. 3.46 Robinett Reduction Transformation C: She's going to have someone mow the lawn.
			*				41. 3.47 Robinett Response Transformation C: Do you need any chairs? Do you need any help?
			*				46. 3.62 Cook Lexical Pair Substitution C: Is Bill young? Is John rich?
			*				47. 3.63 Cook Lexical Sets Substitution C: I'm seeing him on Tuesday. He's meeting her on Saturday.
		*					48. 3.64 Cook Lexical Meaning Substitution C: It's raining! The sun's come out!
			*				49. 3.65 Cook Pronoun Substitution C: I suppose he was there. I suppose she was there.
			*				50. 3.66 Cook Knowledge Substitution C: Who wrote Hamlet? Who was Queen Victoria's husband?

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
				*			51. 3.106 Stack Replacement C: I am buying the <u>typewriter</u>. C: I am buying the <u>books</u>.
?				*			52. 3.111 Stack Replacement Cf: "replace cue with ' y '" C: Allons <u>au cinema</u>. Allons <u>à la gare</u>.
				*			53. 3.114 Stack Replacement C: Paul has <u>twelve</u> records. Sue bought <u>three</u> cars.
							D. M:+C: --> M:--.+ [] <--E:
*						*	27. 3.33 Rivers Type A Deletion P: They haven't [any] coffee. She didn't [come]. Cf: "Delete negative elements" --> n't
							E. E: --> M:[]
	*						26. 3.32 Rivers Type B Expansion
*							30. 3.36 Rivers Type B Completion Cf: "Use infinitive construction" --> (not) to ...
							II. Removal of redundant elements M:+Rd: --> M:--
							Not Present in Simple Operations
							III. Re-order of model elements M:[1][2] --> M:[2][1]
							See Chart 3.9 Models
*					*		4. 3.7 Dacanay Transposition Transformation Cf: "Change to question" = [1][2][3] --> [2][1][3]
*							5. 3.8 Dacanay Transposition Transformation Cf: "Change position of modifiers" = [1][2] --> [2][1]

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
*							11. 3.14 Dacanay Integration Transformation Cf: "Combine the questions without transposing or reducing any of the elements" = [1][2] --> [2][1]
*							59. 3.127 Stack Transformation: word order Cf: "Begin with adverb" = [1][2][3] --> [3][2][1]
*							60. 3.129 Stack Question Formation Cf: "Make a question by inversion" = [1][2] xxx --> [2][1]
							IV. Production of new elements, based on model M:-->R:
		*		*			24. 3.31A Rivers Restatement Conversion Cf: "Follow Instructions given in Cue" C: Tell George your name is Ronald. Ask Alise where she's going. Ask her to wait for you.
		*		*			31. 3.37 Rivers Question-Answer Practice Cf: "Answer the questions truthfully" C: Why didn't they come home before midnight? Do you often stay out late at night?
		*		*			32. 3.38 Rivers Question-Answer Practice Cf: "Answer the question or provide a question as indicated" C: (generated) Does it live on a farm? Is it an elephant? What is it? Give me a question to distinguish a rhinoceros from an elephant.
		*		*			33. 3.39 Rivers Rejoinder Exercise Cf: "Respond to the cue with an appropriate rejoinder" C: Didn't get the job. Just got there as the manager went off to lunch. See you at the bus stop.

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separates: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
			*		*		42. 3.54 Cook Typical Drill Cf: "Answer the questions in the negative" C: Is Bill playing tennis tonight? Is Susan helping her mother this evening?
			*		*		70. 3.145 Stack Question Drill: narration sequence Cf: "Answer the questions truthfully" C: Did she like her red cape? Did she use the Cadillac?
							Complex Drills: MULTIPLE OPERATIONS: 1. Insertion + Insertion
		*				*	9. 3.12 Dacanay Transposition & Expansion P: He [likes] coffee. Mary [buys] instant coffee. C:+M: He [] n't [] coffee. Mary [] n't [] instant coffee.
		*				*	17. 3.22 Rivers Correlative Substitution P: He [brings] [his] lunch. You [bring] [your] lunch. (interlocked) C:+M: You [] [] lunch. John and Mary [] [] lunch.
				*	*	*	64. 3.134 Stack Paired Sentence: tense linkages Cf: "first element to imperfect, second to passé composé" = quand P: Alain [étudie]. Je [rentre]. Marc [parle]. Roger [vient]. C:+M: Alain [] quand je []. Marc [] quand je [].

Appendix b: Table D 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1. 2. 3. 4. 5. 6. 7.

II. Insertion, then Inversion

/* */ * — 18. 3.24 Rivers Multiple Substitution:

P: [She] [brings] [too many] [pencils] to [school].
[Peter] [brings] [too many] [pencils] to [school].
[Peter] [brings] [too much] [money] to [school].

C1: Peter, money, the library...

C2: Peter [] [] pencils to school.
Peter [] [] money to school.

/* */ */ — 19. 3.26 Rivers General Conversion

C1: Peter [has] a new car.
They [stop] at stop signs.

C1f: "Convert to questions" --> do/does

C2: Does Peter [] a new car.
Do they [] at stop signs.

/* */ */ — 25. 3.31B Rivers Restatement Conversion

C1: She said, "I've just arrived but I'm leaving...
She asked, "Why are you looking at her like...

C1f: "Restate using indirect speech" -> Pronoun Conversion Rules

C2: She said she [] just arrived but 0 [] leaving...
She asked me why I [] looking at [] like...

/* */ */ — 37. 3.43 Robinett Moving Slot & Correlative Substitution

P: [She] [is buying] a car [today].
[She] [bought] a car [yesterday]. (interlocked)

C1: yesterday, they...

C2: She [] a car yesterday
They [] a car yesterday

/* */ /* — 65. 3.136 Stack Paired Sentence

P: The boy is [polite].
The girl is [careful].

C1: How does he speak?
How does she walk?

C2: The boy speaks [].
The girl walks [].

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1. 2. 3. 4. 5. 6. 7.

III. Re-order, then Re-order

1.	2.	3.	4.	5.	6.	7.	
*							12. 3.15 Dacanay Integration & Transposition Cf: "Combine the questions, putting the yes/no question first"

MIXED OPERATIONS

I. Insertion + Removal

1.	2.	3.	4.	5.	6.	7.	
		*					10. 3.13 Dacanay Reduction Tranformation C: noisy,...
*/		/*					14. 3.16 Dacanay Integration Transformation with Reduction C1f: "Combine the two sentences" C2: too..to
*	*						23. 3.30 Rivers Combination Conversion c1f: "Combine the two sentences"

II. Insertion, then Removal, then Insertion

1.	2.	3.	4.	5.	6.	7.	
*	*						62. 3.131 Stack Paired Sentences: relative pronouns c1f: "Combine the two sentences"

III. Re-order, then production of new element

1.	2.	3.	4.	5.	6.	7.	
*/		/*		/*	*/		38. 3.44 Robinett Transposition Transformation P: She can swim She can play tennis C1f: "Make a question" C2f: "Answer the question" C2: Can she swim? Can she play tennis?

Appendix b: Table 3.11

1. No explicit Cue: formalized drill
2. No explicit Cue: other reason
3. Cue imbedded in model:
4. Cue separate: alone
5. Cue separate: isolation of cue from cue framework
6. Partially formalized drill (instruction needed to isolate cue)
7. Separate Prompt needed

1.	2.	3.	4.	5.	6.	7.	
*/		/*		/*	*/		39. 3.45 Robinett Integration Transformation

P: Who is she? Do you know?
How far is it to Chicago? Do you know?
C1f: "Make a question"
C2f: "Answer the question"
C2: Do you know who she is?
Do you know how far it is to Chicago?

IV. Production of new elements, then insertion

			*	*			71. 3.146 Stack Question Drill: directed answers

C1f: "Answer the question affirmatively..."
C2f: "...using the cue word."
C1: Does Robert study his lessons?
Does George work on Sundays?
C2: always,sometimes,...

CHOICE OF TEMPLATE

I. Insertion or Re-order

*							20. 3.27 Rivers General Conversion

Cf: "Change the statements into questions"
= [1][2]xxx --> [2][1]xxx
= C: do/does --> M:[]

II. Insertion (type A) or Insertion (type C)

			*	*			61. 3.130 Stack Chain Transformation

a. C: read,...
b. C:+M: She [] study.

Appendix B: Table 3.12 Variations in Response-Set

1. This table should be interpreted with reference to section 3.10.1 of the text

2. Symbols and conventions adopted in this table:

{} denotes a set

{R} Universal response-set
{r} Item-specific response-set

(UPPER CASE denotes universal material
lower case denotes item-specific material)

#{R} Inherently explicit response-set (Where there is an explicit response-set, it is assumed to be "very small". Hence, if column 4 is indicated, then column 5 is not.)

@{R} Actual response-set

[1] Multiple-choice response

{DATABASE} Use of large-scale database techniques required

aaa/bbb/ccc/.../nnn/ Possible responses

aax/aay Possible responses that differ in form only

{a}/{b} Patterned response-set

Appendix B Table 3.12

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-set: (r)
4. Inherently explicit response-set:#(r)
5. Very Small response-set
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	Simple Operations
								I. Insertions into model
								A. C: --> M:[]
?						*		1. 3.4 Dacanay Simple Substitution (R)=(DATABASE) = (words that collocate with "I'm going")
	*		*					3. 3.6 Dacanay Moving Slot Substitution #(R) = @(R) = ([1]/[2])
	*		*					6. 3.9 Dacanay Transposition Transformation #(R) = @(R) = ([1]/[2])
		*	*					7. 3.10 Dacanay Expansion Transformation #(r) = @(r) = ([1]) / ([1]/[2]) / ([1]/[2]/[3])...
	*		*					8. 3.11 Dacanay Expansion Transformation #(R) = @(R) = ([1]/[2]/[3])
	*		*					16. 3.21 Rivers Double Substitution #(R) = @(R) = ([1]/[2])
?						*		34. 3.40 Robinett Simple Substitution (R)=(DATABASE)=(set of objects that can be on tables)
	*		*					36. 3.42 Robinett Moving Slot Substitution #(R) = @(R) = ([1]/[2]/[3]/[4])
	*		*					44. 3.60 Cook Plain Substitution #(R) = @(R) = ([1]/[2]/[3])
		*	*					45. 3.61 Cook Sequence Substitution #(r) = (aaa/bbb) = (swimming/skating)

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-set: (r)
4. Inherently explicit response-set: (r)
5. Very Small response-set
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	
		*			*		*	56. 3.118 Stack Transformation: directed increment (r) = ((a)/(b)/.../(n)) (a) = right tense, faulty form (b) = wrong tense (c) = missing, faulty elements
	*				*		*	57. 3.120 Stack Transformation: negation (R) = ((A)/(B)/.../(N)) (R) = (A) = (aren't/are not/arent/arnt...) = (B) = wrong verb form = (C) = failure to negate properly
	*			*				58. 3.125 Stack Transformation: comparisons (R) = (AAA/BBB/CCC) = (más/mejor/más bien)
		*			*			69. 3.144 Stack Question Drill: you-I conversion (r) = (aaa/bbb/.../nnn) = (do/like it/like music/like it the music...)
								B. E: --> M:C: []
	*			*				2. 3.5 Dacanay Correlative Substitution (R) = (AAA/BBB) = (do/does)
	*			*				28. 3.34 Rivers Type A1 Completion (R) = (AAA/AAX/BBB/BBX) = (will/'ll/would/'d)
		*		*				29. 3.35 Rivers Type A2 Completion (r) = (aaa/bbb/ccc) = (butcher/seller/meat seller)
	*			*				35. 3.41 Robinett Correlative Substitution (R) = (AAA/BBB) = (is/are)
		*			*		*	55. 3.117 Stack Transformation: verb tenses (r) = (a) = improperly formed (b) = wrong tense (c) = wrong agreement
	*			*				63. 3.132 Stack Paired Sentence: conditionals (R) = (AAA/AAX/BBB/BBX) = (will/'ll/would/'d)

Appendix B Table 3.12

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-set: (r)
4. Inherently explicit response-set: #(r)
5. Very Small response-set
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	
	#			#				66. 3.140 Stack Fixed Increment (R) = (AAA/BBB/CCC) = (a/de/0)
		#		#				67. 3.142 Stack Analogy Drill: tag questions (r) = (aaa/bbb) = (read/reads)
		#		#				68. 3.143 Stack Analogy Drill: tag questions (r) = (aaa/bbb) = (negro/negra)
								C. C:-->E:-->M:[]
	#			#				15. 3.20 Rivers Simple Substitution (R) = (AAA/BBB/CCC) = (he/she/they)
		#		#				21. 3.28 Rivers General Conversion (r) = (aaa/bbb) = (her/to her)
	#					#	#	22. 3.29 Rivers General Conversion (R) = (DATABASE) = (nouns that could substitute for "it") = (proper nouns that could subsitute for "her") etc.
		#			#			40. 3.46 Robinett Reduction Transformation (r) = (aaa/bbb/.../nnn) = (combinations of mow/the/lawn)
	#			#				41. 3.47 Robinett Response Transforamation (R) = (AAA/BBB) = (few/little)
		#		#				46. 3.62 Cook Lexical Pair Substitution (r) = (aaa/bbb) = (young/old)
	#			#				47. 3.63 Cook Lexical Sets Substitution (R) = (AAA/BBB/...) = (Days of week)
	#			#				48. 3.64 Cook Lexical Meaning Substitution (R) = (AAA/BBB) = (annoying/nice)
	#					#		49. 3.65 Cook Pronoun Substitution (R) = (DATABASE) = (proper names marked tagged for pronoun)

Appendix B Table 3.12

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-set: (r)
4. Inherently explicit response-set:#(r)
5. Very Small response-set.
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	
*								50. 3.66 Cook Knowledge Substitution (r) = 1 = Shakespeare
	*			*				51. 3.106 Stack Replacement (R) = (AAA/BBB) = (it/them)
*								52. 3.111 Stack Replacement (R) = 1 = y
	*				*			53. 3.114 Stack Replacement (R) = (AAA/BBB/.../NNN) = (everyday counting numbers)
								D. M:+C: --> M:-- + [] <--E:
		*		*				27. 3.33 Rivers Type A Deletion (r) = (aaa/bbb) = (any/some) (come/came)
								E. E: --> M:[]
*								26. 3.32 Rivers Type B Expansion
*								30. 3.36 Rivers Type B Completion
								II. Removal of redundant elements M:+Rd: --> M:-- Not Present in Simple Operations
								III. Re-order of model elements M:[1][2] --> M:[2][1]
	*		*					4. 3.7 Dacanay Transposition Transformation #(R) = #(R) = (1-2-3 -> 6 combinations)
	*		*					5. 3.8 Dacanay Transposition Transformation #(R) = #(R) = (1-2/2-1 = 2 combinations)
	*		*					11. 3.14 Dacanay Integration Transformation #(R) = #(R) = (1-2/2-1 = 2 combinations)

Appendix B Table 3.12

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-set: (r)
4. Inherently explicit response-set:*(r)
5. Very Small response-set
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	
	*		*					59. 3.127 Stack Transformation: word order *(R) = *(R) = (6 combinations)
	*		*					60. 3.129 Stack Question Formation *(R) = *(R) = (2 combinations)
								IV. Production of new elements, based on model M:-->R:
*								24. 3.31A Rivers Restatement Conversion
*								31. 3.37 Rivers Question-Answer Practice
	*					*		32. 3.38 Rivers Question-Answer Practice (R) = (DATABASE) = (Animal names with questions & answers)
*								33. 3.39 Rivers Rejoinder Exercise
		*			*			42. 3.54 Cook Typical Drill (r) = (aaa/bbb/.../nnn) = complex template
		*			*			70. 3.145 Stack Question Drill: narration sequence (r) = (aaa/bbb/.../nnn)
								MULTIPLE OPERATIONS:
								I. Insertion + Insertion
	*/	/*		*/*				9. 3.12 Dacanay Transposition & Expansion (R1) = (AAA/BBB) = (DO/DOES) (r2) = (aaa/bbb) = (like/likes)
	/*	*/		*/*				17. 3.22 Rivers Correlative Substitution (r1) = (aaa/bbb) = (bring/brings) (R2) = (aaa/bbb/...) = (POSSESSIVE PRONOUNS)

Appendix B Table 3.12

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-sets: (r)
4. Inherently explicit response-set:#(r)
5. Very Small response-set
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	
		/			*/*		*/*	64. 3.134 Stack Paired Sentence: tense linkages (r1) & (r2) = (aaa/aax/bbb/bbx) = wrong form, wrong tense, wrong agreement
								II. Insertion, then Insertion
	*/	/*	*/	/*				18. 3.24 Rivers Multiple Substitution #(R1) = @(R1) = ([1]/[2]/[3]) (r2) = (aaa/bbb) = (bring/brings), (too many/too much)
	*/	/*		*/*				19. 3.26 Rivers General Conversion (R1) = (AAA/BBB) = (do/does) (r2) = (aaa/bbb) = (has/have)
	1/4	2/3		1/2/3/4				25. 3.31B Rivers Restatement Conversion (R1), (R4) = (AAA/BBB/...) = (Subject Pronouns) (r2) = (aaa/bbb) = (have/had) (r3) = (aaa/bbb) = (am/was)
	/		*/	/*				37. 3.43 Robinett Moving Slot & Correlative Substitution #(R1) = @(R) = ([1]/[2]) (R2) = (AAA/BBB/...) = (tense & person combinations of "is buying")
		/		*/*				65. 3.136 Stack Paired Sentence (r1) = (aaa/bbb) = (speak/speaks) (r2) = (aaa/bbb) = (polite/politely)
								III. Re-order, then Re-order
	/		*/*					12. 3.15 Dacanay Integration & Transposition #(R1) = @(R) = (1-2/2-1) = 2 combinations #(R2) = @(R) = (1-2/2-1) = 2 combinations
								MIXED OPERATIONS
								I. Insertion + Removal
	*/	/*	*/	/*				10. 3.13 Dacanay Reduction Tranformation #(R1) = @(R) = ([1]/[2]) (r2) = (aaa/bbb/...) = (words in sentence)

Appendix B Table 3.12

1. No finite response-set	5. Very Small response-set
2. Universal response-set: (R)	6. Relatively Small response-set
3. Item-specific response-set: (r)	7. Large, but still finite response-set
4. Inherently explicit response-set: #(r)	8. Patterned response-set

1.	2.	3.	4.	5.	6.	7.	8.	
	/*	*/	*/*					14. 3.16 Dacanay Integration Transformation with Reduction (r1) = (aaa/bbb/...) = (words in sentence) #(R2) = @(R) = (1-2/2-1)
	/*	*/	*/*					23. 3.30 Rivers Combination Conversion (r1) = (aaa/bbb/...) = (words in sentence) (R2) = (that/0)
								II. Insertion, then Removal, then Insertion
	1/3	2	1/2	/3				62. 3.131 Stack Paired Sentence: relative pronouns #(R1) = @(R) = ([1]/[2]/[3]) (r2) = (aaa/bbb/...) = (words in sentence) (R3) = (AAA/BBB/CCC) = (that/who/0)
								III. Re-order, then production of new element
	/		*/		/*			38. 3.44 Robinett Transposition Transformation #(R1) = @(R) = (1-2/2-1) (R2) = (AAA/BBB/.../NNN) = (Yes/No/Yes, she can;...)
	/		*/		/*			39. 3.45 Robinett Integration Transformation #(R1) = @(R) = (1-2/2-1) (R2) = (AAA/BBB/.../NNN) = (Yes, I do/No, I don't)
								IV. Production of new elements, then insertion
	/*	*/	/*	*/				71. 3.146 Stack Question Drill: directed answers (r1) = (aaa/bbb/ccc...) = (possible responses) #(R2) = @(R) = ([1]/[2])
								CHOICE OF TEMPLATE
								I. Insertion or Re-order
	1/2a/2b			1/2a/2b				20. 3.27 Rivers General Conversion (R1) = (AAA/BBB) = (template A/template B) #(R2a) = @(R2a) = (1-2/2-1) (R2b) = (AAA/BBB) = (do/does)

Appendix B Table 3.12

1. No finite response-set
2. Universal response-set: (R)
3. Item-specific response-set: (r)
4. Inherently explicit response-set:$(r)
5. Very Small response-set
6. Relatively Small response-set
7. Large, but still finite response-set
8. Patterned response-set

1. 2. 3. 4. 5. 6. 7. 8.

II. Insertion (type A) or Insertion (type C)

a/ /b /b

61. 3.130 Stack Chain Transformation
   a. (R) = i = [i]
   b. (R) = (AAA/BBB) = (doesn't/don't)

Appendix B: Table 3.13 Variations in Answer-Set

1. This table should be interpreted with reference to section 3.10.2 of the text.

2. Symbols and conventions adopted in this table:

aaa	The correct response
{a}	The set of correct responses (the answer-set)
{aa1/aa2/aa3...}	A "very small" answer-set
{aa1/aa2/.../aaN}	A small answer-set
{aa1/aa2/.../aaZ}	A large answer-set
@a	The actual correct response (in the case of multiple-choice).
{aax/aay}	Answers differ in form only

---

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1.	2.	3.	4.	5.	6.	7.	8.	
								Simple Operations
								I. Insertions into model
								A. C: --> M:[]
*								1. 3.4 Dacanay Simple Substitution aaa=away
*				*				3. 3.6 Dacanay Moving Slot Substitution @a=[2] for ... leaves [tonight] at seven.
*				*				6. 3.9 Dacanay Transposition Transformation @a=[2] for ... made a [] dress [for me].
*				*		?		7. 3.10 Dacanay Expansion Transformation @a=[2] for ... two [paper] napkins

Appendix B Table 3.13

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1.	2.	3.	4.	5.	6.	7.	8.	
	*			*				8. 3.11 Dacanay Expansion Transformation @a=[2] for he's [now] leaving. @a=[3] for he's leaving [now].
*				*				16. 3.21 Rivers Double Substitution @a=[2] for [If you want it] he'll ....
*								34. 3.40 Robinett Simple Substitution aaa=pen
*				*				36. 3.42 Robinett Moving Slot Substitution @a=[1] for [He] bought a house...
*				*				44. 3.60 Cook Plain Substitution @a=[3] for I love [whisky].
*								45. 3.61 Cook Sequence Substitution aaa=skating
*							*	56. 3.118 Stack Transformation: directed increment aaa=il se reposera partial: [il] [se] [repos] [era]
	*				*		*	57. 3.120 Stack Transformation: negation (a) = (aax/aay) = (are not/aren't) partial: [are] [not]
*								58. 3.125 Stack Transformation: comparisons aaa = mas
	*						*	69. 3.144 Stack Question Drill: you-I conversion (a) = (aa1/aa2/aa3) = (do/like it/like music) partial [word1]-> (word2)

B. E: --> M:C: []

1.	2.	3.	4.	5.	6.	7.	8.	
*								2. 3.5 Dacanay Correlative Substitution aaa = do

Appendix B Table 3.13

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1.	2.	3.	4.	5.	6.	7.	8.	
	*				*			28. 3.34 Rivers Type A1 Completion (a) = (aax/aay) = (would/'d)
*								29. 3.35 Rivers Type A2 Completion aaa = butcher
*								35. 3.41 Robinett Correlative Substitution aaa = are
*							*	55. 3.117 Stack Transformation: verb tenses aaa = a commande partial [a] [command] [e]
	*				*			63. 3.132 Stack Paired Sentence: conditionals (a) = (aax/aay) = (will/'ll)
*								66. 3.140 Stack Fixed Increment aaa = a
*								67. 3.142 Stack Analogy Drill: tag questions aaa = reads
*								68. 3.143 Stack Analogy Drill: tag questions aaa = negro
								C. C:-->E:-->M:[]
*								15. 3.20 Rivers Simple Substitution aaa = he
	*				*	*		21. 3.28 Rivers General Conversion RULE: If [1] then aaa = her [2] aaa = to her
			*			*		22. 3.29 Rivers General Conversion RULE: (a) = (nouns in DATABASE that could substitute for "it")
*							*	40. 3.46 Robinett Reduction Transformation aaa = the lawn mowed partial [the] [lawn] [mowed]

Appendix B Table 3.13

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1.	2.	3.	4.	5.	6.	7.	8.	
*								41. 3.47 Robinett Response Transformation aaa = few
*								46. 3.62 Cook Lexical Pair Substitution aaa = old
*						?		47. 3.63 Cook Lexical Sets Substitution aaa = Wednesday RULE generation easy
*								48. 3.64 Cook Lexical Meaning Substitution (R) = (AAA/BBB) = (annoying/nice)
			*			*		49. 3.65 Cook Pronoun Substitution RULE: (a) = (proper names marked tagged for "he")
*								50. 3.66 Cook Knowledge Substitution aaa = Shakespeare
*								51. 3.106 Stack Replacement aaa = it
*								52. 3.111 Stack Replacement AAA = y
*						?		53. 3.114 Stack Replacement aaa = thirteen RULE generation easy
								D. M:+C: --> M:-- + [] <--E:
*								27. 3.33 Rivers Type A Deletion aaa = any
								E. E: --> M:[]
			*					26. 3.32 Rivers Type B Expansion
							*	30. 3.36 Rivers Type B Completion partial: (not) [to] xxx

Appendix B Table 3.13

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1.	2.	3.	4.	5.	6.	7.	8.	
								II. Removal of redundant elements M:+Rd: --> M:--
								Not Present in Simple Operations
								III. Re-order of model elements M:[1][2] --> M:[2][1]
*				*				4. 3.7 Dacanay Transposition Transformation @a=2-1-3 for Is the girl ready?
*				*				5. 3.8 Dacanay Transposition Transformation @a=2-1 for He was here this morning.
*				*				11. 3.14 Dacanay Integration Transformation @a=2-1 for Do you know who made the ...
*				*				59. 3.127 Stack Transformation: word order @a=3-2-1 for Tief ist der Brunnen.
*				*				60. 3.129 Stack Question Formation @a=2-1 for Allons-nous xxx ?
								IV. Production of new elements, based on cue C:-->R:
		*					*	24. 3.31A Rivers Restatement Conversion partial: complex template
	?		*				*	31. 3.37 Rivers Question-Answer Practice partial: key word search
*						*		32. 3.38 Rivers Question-Answer Practice RULE generated, based on (DATABASE) = (Animal names with questions & answers)
		*						33. 3.39 Rivers Rejoinder Exercise
		*					*	42. 3.54 Cook Typical Drill partial: complex template
		*					*	70. 3.145 Stack Question Drill: narration sequence partial: [word1] --> (word2) --> (word3)

Appendix B Table . 3.13

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1. 2. 3. 4. 5. 6. 7. 8.

MULTIPLE OPERATIONS:

I. Insertion + Insertion

#/# — 9. 3.12 Dacanay Transposition & Expansion
1. aaa = does
2. aaa = like

#/# — 17. 3.22 Rivers Correlative Substitution
1. aaa = bring
2. aaa = your

#/# — #/# 64. 3.134 Stack Paired Sentence: tense linkages
1. aaa = étudiait
2. aaa = suis rentré
partial: [] [etudi] [ait]
[ai] [rentr] [ré]

II. Insertion, then Insertion

#/# (col. 1), #/ (col. 5) — 18. 3.24 Rivers Multiple Substitution
1. @a=[1] for [Peter] brings ...
2. aaa=brings

#/# — 19. 3.26 Rivers General Conversion
1. aaa = does/has
2. aaa = has/got

1/3/4, 2 — 25. 3.31B Rivers Restatement Conversion
1. aaa = she
2. (a) = (had/'d)
3. aaa = 0
4. aaa = was

#/# (col. 1), #/ (col. 5) — 37. 3.43 Robinett Moving Slot & Correlative Substitution
1. @a = [2] for She is buying a car [yesterday];
2. aaa= bought

Appendix B . Table 3.13

1. One correct response		5. Actual Response not true response	
2. Very small answer-set		6. Answers differ in form only	
3. Small answer-set		7. RULE generated answer necessary	
4. Large answer-set		8. Partial processing necessary or desirable	

1.	2.	3.	4.	5.	6.	7.	8.	
/								65. 3.136 Stack Paired Sentence 1. aaa=speaks 2. aaa=politely
								III. Re-order, then Re-order
/				*/*				12. 3.15 Dacanay Integration & Transposition 1. @a = 2-1 for Do you know who is he? 2. @a = 2-1 for -- --- ---- --- he is?
								MIXED OPERATIONS
								I. Insertion + Removal
/				*/*				10. 3.13 Dacanay Reduction Tranformation 1. @a =[1] for The [noisy] children ... 2. @a = who,noise for "who .... noise".
/				/*				14. 3.16 Dacanay Integration Transformation with Reduction 1. aaa = She can't 2. @a = [2][1] for ...[too] sick [to] study ...
*/	/*							23. 3.30 Rivers Combination Conversion 1. aaa= the door 2. (a)=(that/0)
								II. Insertion, then Removal, then Insertion
1/2/	3			1				62. 3.131 Stack Paired Sentence: relative pronouns 1. @a = [1] fwr The man [I saw the man last night] is... 2. aaa= the man 3. (a)= (that/whom/0)
								III. Re-order, then production of new element
*/		/*		*/			/*	38. 3.44 Robinett Transposition Transformation 1. @a = 2-1 for Can she xxx? 2. (a)=(aa1/aa2/.../aaN) =(Yes/Yes, she can/Yes, she can swim) Partial: [Yes/No] --> (word2) -- (word3)...

Appendix B Table 3.13

1. One correct response
2. Very small answer-set
3. Small answer-set
4. Large answer-set
5. Actual Response not true response
6. Answers differ in form only
7. RULE generated answer necessary
8. Partial processing necessary or desirable

1.	2.	3.	4.	5.	6.	7.	8.	
*/		*/*		*/			*/*	39. 3.45 Robinett Integration Transformation 1. @a = 2-1 for Do you know who she is? 2. (a)=(aa1/aa2/.../aaN) =(Yes/No/Yes, I do/No, I don't/Yes, she's xxx...) Partial: [Yes/No] --> (word2) --> (word3) ...
								IV. Production of new elements, then insertion
	/*	*/		*/			*/	71. 3.146 Stack Question Drill: directed answers 1. (a)=(aa1/aa2/.../aaN) partial: [Yes] - (word2) --> (word3) ,.. 2. @a=[1] for Yes) = (xx) [always] xx ...
								CHOICE OF TEMPLATE
								I. Insertion or Re-order
1/2a/2b			1/2a					20. 3.27 Rivers General Conversion 1. @a = [1] for template 2a 2. @a = 2-1 for Are John and I ... ?
			(R2b) = (AAA/BBB) (do/does)					
								II. Insertion (type A) or Insertion (type C)
a/b								61. 3.130 Stack Chain Transformation [a. @a = 1 for They don't [read]. b. aaa = doesn't

Appendix B: Table 3.16

Discrete Pedagogical Factors Illustrated in the Data

1. This table should be interpreted with reference to section 3.12 of the text

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

Y = YES, definately exhibits this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
-	-	-	-	-	-	-	-	-	3.48	Robinett	Mechanical Drill
Y	Y	Y	-	-	-	Y	-	-	3.49	Robinett	Meaningful Drill
Y	Y	Y	-	-	-	Y	-	-	3.50	Robinett	Communicative Drill
-	-	-	-	-	-	-	-	-	3.158	Robinett	Manipulative Drill
Y	Y	-	-	-	-	-	-	-	3.159	Robinett	Communication Drill
-	-	-	-	-	-	-	-	-	3.51	Paulston	Meaningless Drill -1
-	-	-	-	-	-	-	-	-	3.52	Paulston	Meaningless Drill -2
-	-	-	-	-	Y	y	-	Y	3.53	Paulston	Meaningful Drill
-	-	-	-	-	-	-	-	-	3.55	Cook	Non-Contextualised Drill
Y	-	-	-	Y	-	-	-	-	3.56	Cook	Semi-Contextualised Drill
Y	Y	-	-	y	-	-	-	-	3.57	Cook	Contextualised Drill
Y	Y	Y	-	-	Y	Y	-	-	3.58	Cook	Situational Drill

Appendix B: Table 3.16

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

Y = YES, definately exhibits this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
-	-	-	-	-	-	-	-	-	3.68	Byrne	Mechanical Drill
-	-	Y	-	-	Y	Y	-	-	3.69	Byrne	Meaningful Drill -1
-	Y	Y	-	-	Y	Y	-	-	3.70	Byrne	Meaningful Drill -2
Y	Y	-	Y	-	Y	Y	-	Y	3.71	Byrne	Special Meaningful Drill
-	-	-	-	-	Y	y	Y	Y	3.72	Byrne	Open Ended Responses
Y	Y	-	y	-	Y	-	Y	-	3.73.1	Byrne	Imaginary Situations
Y	Y	-	-	Y	Y	-	-	Y	3.73.2	Byrne	Imaginary Situations
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne	Meaningful Drill -3
Y	Y	-	-	-	Y	Y	-	-	3.75	Byrne	Meaningful Drill -4
											(All "Communicative")
-	Y	-	-	-	-	Y	-	-	3.76	Candlin	Get Organized: Concept
-	Y	Y	-	-	-	-	-	-	3.77	Candlin	Get Organized: Narration
-	Y	Y	-	-	-	-	-	-	3.78	Candlin	Get Organized: True/false
Y	Y	Y	-	-	Y	-	Y	-	3.79	Candlin	Get Organized: Connecting
-	Y	Y	-	-	-	-	-	-	3.80	Candlin	Get Organized: Multiple choice
-	Y	-	-	-	-	Y	-	-	3.81	Candlin	Get Organized: Recognition
Y	Y	Y	-	-	-	-	-	-	3.82	Candlin	Get Organized: Word Scramble

Appendix B: Table 3.16

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

Y = YES, definately exhibits this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin	Implanting Skills: Memorizing Cognitive Language
Y	Y	-	-	y	-	-	-	-	3.85	Candlin	Implanting Skills: Memorizing Affective Language
Y	Y	-	-	-	Y	Y	Y	-	3.86	Candlin	Implanting Skills: Picture Skills
-	Y	-	-	-	-	Y	-	-	3.87	Candlin	Implanting Skills: Restoration-1
-	Y	-	-	-	Y	-	-	Y	3.89	Candlin	Implanting Skills: Restoration: Catchwords
-	Y	Y	-	-	-	-	-	-	3.90	Candlin	Implanting Skills: Restoration: Gap Text
Y	Y	Y	-	-	-	-	-	-	3.91	Candlin	Implanting Skills: Restoration: Gap Text/New Text
Y	-	-	-	-	-	-	-	-	3.92	Candlin	Implanting Skills Restoration:
Y	-	-	-	-	-	-	-	-	3.93	Stevick	Manipulative
Y	Y	-	-	-	-	-	-	-	3.94	Stevick	Meaningful
											(All Meaningful)
-	-	Y	-	-	Y	Y	Y	-	3.95	Dakin	Application Drill
-	-	Y	-	-	Y	Y	Y	-	3.96	Dakin	General Knowledge Drill

Appendix B: Table 3.16

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

Y = YES, definately exhibits this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
Y	-	-	-	-	Y	Y	Y	Y	3.97	Dakin	Collocation Drill
-	-	-	-	-	-	Y	-	-	3.98	Dakin	Synonymy Drill
-	-	-	-	-	Y	Y	Y	-	3.99	Dakin	Hyponymy Drill
-	-	-	-	-	Y	Y	-	-	3.100	Dakin	Antonymy Drill
Y	Y	Y	-	-	-	Y	-	-	3.101	Dakin	Converse Drill
-	-	-	-	-	Y	Y	Y	-	3.102	Dakin	Consequence Drill
-	-	-	-	-	-	-	-	-	3.150	Beile	Non-Communicative
Y	-	-	-	-	-	-	-	-	3.151	Beile	Isolated Communicative Interchange
Y	Y	-	-	Y	-	-	-	-	3.152	Beile	Isolated Communicate Interchange in Situational Context
Y	-	-	-	-	-	-	-	-	3.153	Beile	Connected Communicative Interchange
Y	Y	-	-	Y	-	-	-	-	3.154	Beile	Connected Communicative Interchange in Situational Context
-	-	-	-	-	-	Y	-	-	3.155	Beile	No Contextualisation
-	-	-	-	-	-	Y	-	-	3.156	Beile	Dialog-like, Not Contextualisable
Y	y	-	-	-	-	Y	-	-	3.157	Beile	Contextualisable

(unpatterned)

1. Realistic Interchange 6. More than One Response Y = YES, definately exhibits

Appendix B: Table 3.16

2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
Y	Y	Y	-	-	-	-	-	-	3.160	Stack	Meaningful Drill -1
Y	-	-	-	-	Y	y	Y	-	3.161	Stack	Meaningful Drill -2 (patterned)

Appendix B Table 3.17

Detailed Analysis and Categorization of Pedagogical Sample according to Cue and Response Model:
-Including Presentation & Response Analysis Factors

1. This table should be interpreted with reference to section 3.13 of the text.

2. An explanation of the symbols and conventions adopted can be found in tables 3.9 - 3.13.

Part One: Simple Operations

I. Insertions into model

A. Insertion of cue into model: C: --> M:[]

2. 3.49 Robinett Meaningful Drill

P:	PPP	(Drawings of people on the blackboard)
C:	ccc(1)	Who is older, Susie or David?
M:	MMM	[]
C:	ccc(2)	Who is shorter, Susie or Mary?
M:	MMM	[]
R:	(R)=vs	(Susie/David)
A:	aaa	Mary

3. 3.50 Robinett Communicative Drill

P:		(Actual people in the Classroom)
C:	ccc(1)	Who is older, Susie or David?
M:	MMM	[]
C:	ccc(2)	Who is shorter, Susie or Mary?
M:	MMM	[]
R:	(R)=vs	(Susie/David)
A:	aaa	Mary

5. 3.159 Robinett Communication Drill

Cf: (CC1/CC2) Use some/any
M: mmm(1) I'd like [] eggs, please.
M: mmm(2) I'm sorry, there aren't [].

R: (R)=vs (some/any)
A: aaa(1) some
aaa(2) any

6. 3.51 Paulston Meaningless Drill

C: ccc(1) tall
M: mmm(1) the [] student.
C: ccc(2) fat
M: mmm(2) the [] student.

R: no finite (r)
A: aaa(1)=ccc(1)
aaa(2)=ccc(2)

9. 3.55 Cook Non-Contextualised Drill

C: ccc(1) He
M: MMM [] is going to Paris.
C: ccc(2) The bus
M: MMM [] is going to Paris.

R: no finite (r)
A: aaa(1)=ccc(1)
aaa(2)=ccc(2)

10. 3.56 Cook Semi-Contexualised Drill

C: ccc(1) Fred's going to change his job.
M: mmm(1) [] ? Changing his job ? I don't believe it.
C: ccc(2) Jane's going to clean the car.
M: mmm(2) [] ? Cleaning the car ? I don't believe it.

R: no finite (r)
A: aaa(1)=ccc(1)
aaa(2)=ccc(2)

37. 3.93 Stevick Manipulative Drill

C:	ccc(1)	When will they go?
M:	MMM	Haven't they [] yet?
C:	ccc(2)	When will they leave?
M:	MMM	Haven't they [] yet?
R:	(r)=vs	(go/gone/went/...)
A:	aaa	gone

38. 3.94 Stevick Meaningful Drill

T:	TTT	Paragraph about grocery stores
C:	ccc(1)	When will they buy groceries?
M:	mmm(1)	Haven't they [] groceries yet?
C:	ccc(2)	When will they stock the counter?
M:	mmm(2)	Haven't they [] the counter yet?
R:	(r)=vs	(buy/bought/boughten...)
A:	aaa	bought

47. 3.150 Beile Non Communicative Interchange

Cf:		Put into the past tense
C:	ccc	He writes a lot of essays.
M:	mmm	He [] a lot of essays.
R:	(r)=vs	(write/wrote/writed/...)
A:	aaa	wrote

51. 3.154 Beile Connected Communicative Interchange in a situational context.

T:	TTT	(Situation)
C:	ccc(1)	I've got some hot tea for you, Linda.
M:	MMM	I don't want any [].
C:	ccc(1)	Then have some milk
M:	MMM	I don't want any [].
R:	(R)=vs	(beverages & other things that could be offered)
A:	aaa(1)	hot tea
	aaa(2)	milk

B. Insertion of secondary element into model E: --> M:C1 []

22. 3.76 Candlin Get Organized: Concept

C:+M: mmm(1) A shoplifter [].
C:+M: mmm(2) A regular [].

R: (r)=large (all possible actions of characters in the domain, with all possible phrasings)
A: (a)=large (all possible ways to describe shoplifter)

23. 3.77 Candlin Get Organized: Narration

P: PPP (Reading: "The Mystery Object Lands")
C:+M: mmm(1) The mystery object landed [].
C:+M: mmm(2) The police [].

R: (r)=large (all possible actions in story, with all possible phrasings)
A: (a)=large (all possible ways to describe the landing of the mystery object)

32. 3.87 Candlin Implanting Skills: Restoration

P: PPP (Global context of paragraph)
Cf: Use the form "has/have just ..."
C:+M: mmm(1) The police [].
C:+M: mmm(2) They [] rope barriers.

R: (r)=small (short li[illegible] verbs and their possible forms)
A: aaa(1) the right verb, in the right form

35. 3.91 Candlin Implanting Skills: Gap Text/New Test

P: PPP (Scenes concerning Caroline's day)
C:+M: mmm(1) I bought a [] at the camp shop today ...

R: (R)=small (the list of substantives contained in the story)
A: aaa(1) the object shown to have been bought by Caroline in the scene

40. 3.96 Dakin General Knowledge Drill

Cf: Make a true statement
C:+M: mmm(1) Paul McCartney plays [].
C:+M: mmm(2) Yehudi Menuhin plays [].
R: (R)=large (words and expression that can collocate with "play")
A: (a)=small (elements of (r) that pertain to Paul McCartney)

52. 3.155 Beile No Contextualisation

C:+M: mmm(1) The baker bakes [].
C:+M: mmm(2) The greengrocer sells [].

R: (R)=small (The list of common objects in the grocery domain)
A: (a)=small (the elements of (r) that a baker could bake)

53. 3.156 Beile Dialog-like, but not Contextualisable

C: ccc(1) What does the baker do?
C:+M: mmm(1) The baker [] bread.
C: ccc(2) What does the greengrocer do?
C:+M: mmm(2) The greengrocer [] fruit and vegetables.

R: (R)=small (the list of verbs that can apply to the domain, in all predictable forms)
A: (a)=vs (the verbs, in the proper tense, that can describe what a baker "does" in relation to "bread")

54. 3.157 Beile Contextualisable

Cf: Answer, stating that you "do" your own
T: ttt(1) Where do you buy your bread?
C:+M: mmm(1) We [] our own bread.
T: ttt(2) Which greengrocer do you go to?
C:+M: mmm(2) We [] our own fruit and vegetables.

R: (R)=small (the list of verbs describing what can be "done" to objects in the domain)
A: (a)=vs (the elements of (R), in the proper from, which pertain to bread)

C. Insertion of element triggered by cue C:-->E:-->M:[]

1. 3.48 Robinett Mechanical Drill

C: ccc(1) old
M: MMM John is [] than Bill.
C: ccc(2) big
M: MMM John is [] than Bill.

R: (r)=vs (old/older/oldest)
A: aaa(1) older

11. 3.57 Cook Contextualised Drill

C:	ccc(1)	Are you coming the the party?
M:	MMM	No, [] not.
C:	ccc(2)	But, Susan's coming, I'm sure
M:	MMM	No, [] not.
R:	(R)=vs	(I'm/I am/you're/you are/...)
A:	(a)=vs	(I'm/I am)

13. 3.68 Byrne Mechanical Drill

P:	ppp(1)	The dictionary you asked for has been stolen.
C:	ccc(1)	lose
M:	MMM	The dictionary you asked for has been [].
P:	ppp(2)	The dictionary you asked for has been lost.
C:	ccc(2)	borrow
M:	MMM	The dictionary you asked for has been [].
R:	(r)=vs	(lose/lost/losed/...)
A:	aaa(1)	lost

16. 3.71 Byrne Meaningful Practice

Cf:		Ask questions do determine what is in the room
M:	MMM	Is there a/[] in the room?
F:		yes
R:	MMM	Is there a [] in the room?
F:		yes
R:	(R)=large	(common objects)
A:	(a)=small	(Objects liable to be found in a room and that have not yet been asked.)

21. 3.75 Byrne Meaningful Drill

C:	ccc(1)	I've been working all day.
M:	MMM	You must be [].
C:	ccc(2)	I haven't eaten a thing since breakfast.
M:	MMM	You must be [].
R:	(R)=vs	(The set of common state adjectives)
A:	(a)=vs	(those adjectives that could describe a possible state resulting from ccc(1))

24. 3.78 Candlin Get Organized: true/false

P:		(Reading about a flying saucer)
Cf:		Respond with true or false
C:	ccc(1)	There is a thing on Russian Hill.
M:	MMM	[]
C:	ccc(2)	There is a thing on Gibbet Hill.
M:	MMM	[]
R:	(R)=vs	(True/False)
A:	aaa	False

27. 3.81 Candlin Get Organized: Recognition

P:	PPP	(Horror story with sound effects)
C:	ccc(1)	(the sound of hollow laughter)
M:	MMM	The sound is [].
R:	(R)=vs	(possible descriptions of the twelve identifiable sounds on the tape)
A:	(a)=vs	(the possible descriptions of the sound in question)

34. 3.90 Candlin Implanting Skills: Gap Text

C:	ccc(1)	I came home one night...
M:	mmm(1)	The young man [].
C:	ccc(2)	...fell flat in the dustbins...
M:	mmm(2)	He fell in [].
R:	(r)=vs	(came home/come home/came/...)
A:	aaa(1)	came home

39. 3.95 Dakin Application Drill

C:	ccc(1)	(Scene of girl sleeping)
M:	mmm(1)	Felicity is [].
C:	ccc(2)	(Scene of boy taking bath)
M:	mmm(2)	Anthony is [].
R:	(r)=vs	(sleeping/asleep/sleep/sleeps...)
A:	(a)=vs	(sleeping/asleep)

41. 3.97 Dakin Collocation Drill

C: ccc(1) This is a wonderful book.
M: MMM Good, I'd like to [] it.
C: ccc(2) There's a good film at the cinema this week.
M: MMM Good, I'd like to [] it.

R: (r)=vs (read/borrow...)
A: (a)=vs (read/borrow...)

44. 3.100 Dakin Antonymy Drill

C: ccc(1) Pala isn't a rich country.
M: MMM There's still a lot of [].
C: ccc(2) Pala isn't a healthy country.
M: MMM There's still a lot of [].

R: (R)=small (List of nouns appropriate to the domain)
A: (a)=vs poverty, hunger...

55. 3.160 Stack Meaningful Drill (unpatterned)

P: (Narration about Robert's visit to a restaurant)
C: ccc(1) Where did Robert go?
M: aaa(1) He went to a [].
C: ccc(2) When did he arrive?
M: aaa(2) He arrived at [].

R: (R)=vs (The points covered in the narration)
A: aaa restaurant

56. 3.161 Stack Meaningful Drill (patterned)

C: ccc(1) My pencil-point is broken.
M: MMM You must [] it.
C: ccc(2) My hair is too long.
M: MMM You must [] it.

R: (r)=vs (sharpen/fix/to sharpen...)
(a)=vs (sharpen/fix/...)

D. Insertion of element triggered by deletion of cue. M:+C: --> [illegible] + [] (--E:

No Examples

I. Free Insertion E: --) M:[]

17. 3.72 Byrne Open Ended Responses

M: mmm John was hungry, so he had ...

R: (r)=small (List of common foods that John could eat)
A: (a)=(r)

18. 3.73 Byrne Imaginary Situations

P: PPP (Situation: furnishing house)
M: MMM Have you bought a [] yet?

R: (R)=large (The set of things that might furnish a house)
A: (a)=(R)

II. Removal of redundant elements M:+Rd: --) M:--

No Examples

III. Re-order of model elements M:[1][2] --) M:[2][1]

28. 3.82 Candlin Get Organized: Word Scramble

P: (Situation)
M: MMM [1] green jacket
[2] scar on forehead
[3] Name:
[4] his mother said to the police
etc.

R: (R)=small (The possible combinations)
A: aaa The correct order

IV. Production of new elements, based on cue C:-->R:-->(M:)

7. 3.52 Paulston Meaningless Drill

C: ccc(1) The dog bit the man.
C: ccc(2) The boing boinged the boing.

R: (R)=small (Predictable errors in syntax and morphology)
A: aaa The man was bitten by the dog.

8. 3.53 Paulston Meaningful Drill

Cf: Ask the question suggested by the answer
C: ccc(1) ...for five years
R: How long did he study?
C: ccc(2) ...during March
R: When did he register?

R: No finite response set
A: No finite holistic answer set, but fixed form for partial processing: How long ... ?

12. 3.58 Cook Situational Drill

P: (Student Cleaning Blackboard)
C: What's he doing?
R: He's cleaning the blackboard.
He's erasing the board.
He's working.

R: (r)=small (relatively finite number of ways of describing action)
(a)=vs (grammatically correct elements of (r))

25. 3.79 Candlin Set Organized: Connecting Exercise

P: (Cartoon Scene for each item, with part of dialogue missing)
Cf: Complete each dialogue
C: (Cartoon Scene)
R: Couple of chewing-gums, 'course you can have it for one.
R: No, sir. I just did it.

R: No finite Response Set, unless multiple choice
A: No finite Answer Set, unless multiple choice

26. 3.80 Candlin Set Organized: Multiple Choice

P: (Reading: the text of an interview)
Cf: Complete each sentence truthfully
C: The police went to see the headmaster because...
R: two of his boys were caught stealing.

R: (R) Holistically: no finite response set
Key words: (R)=vs (Set of key items in story)
A: (a)=vs (key item≠ possible variations in wording)

29. 3.84 Candlin Implanting Skills: Memorizing Cognitive Language

T: (Context: Spring Cleaning Day. Pictures of individuals performing contributory activities)
Cf: Respond to each question by stating what the person in the picture will do
C: ccc (Picture of Richard working)
ccc What about you, Richard?
R: I'll take the drawing-room curtains down.
C: What about Andrew and Harry?
R: They'll roll up the carpets.

R: (r)=small (Possible ways to describe action, including errors)
A: (a)=vs (Possible ways to descrive action)

31. 3.86 Candlin Implanting Skills: Picture Stimulus

C: (Dialogue presented via stylized symbols representing what is being said)
R: The man is looking at the cat.
The man sees the cat.
etc.

R: (R)=small (Possible interpretations of picture + possible phrasings + possible errors)
A: (a)=vs (Possible interpretations of picture + possible phrasings)

33. 3.89 Candlin Implanting Skills: Restoration

P: (List of key phrases)
Cf: Construct the story using the phrases in the list
R: Mrs Dora Shields and ...

R: No finite Response Set
A: No finite Answer Set

36. 3.92 Candlin Implanting Skills: Restoration

P: Do you like this town?
It bores me stiff.
C: For three months (live)
R: How long have you lived here?

R: (R)=small (How long + you + live +this town: possible combinations)
A: (a)=vs (How long have you lived here?
been living ...
in this town)

48. 3.151 Beile Isolated Communicative Interchange

Cf: Ask a yes/no question based on the statement
C: I've always wanted to run a garage.
R: Have you?
C: Mr. Sharp's talking to the directors this afternoon.
R: Is he?

R: (R)=vs (Fairly restricted forms)
A: aaa Have you?

49. 3.152 Beile Isolated Communicative Interchange in a situational context.

T: (Situation)
Cf: Give only a polite reply
C: I've always wanted to run a garage.
R: Have you?
Really?

R: (R)=small (Manageable list of forms)
A: (a)=small

50. 3.153 Beile Connected Communicative Interchange

C: Ask Helen whether she likes to play tennis.
R: Do you like to play tennis, Helen?
Do you like playing tennis, Helen?
Do you like the game of tennis?

R: (R)=small if partial analysis
A: (a)=vs If partial analysis

Part Two: Complex Operations: More than one simple operation

MULTIPLE OPERATIONS: More than one discrete operation to perform

4. 3.158 Robinett Manipulative Drill

Operations: Insertions of cues/Choice of Slots

C: 1. some 2. any
M: We have [] milk, but we don't have [] honey.
a. b.

15. 3.70 Byrne Meaningful Drill

P: PPP (Picture of a room full of Objects)
Cf: Choose objects so as to make true statements
M: MMM There's a [] near the [].
a. b.
R: (R)=vs (the objects in the picture)
A: (a1),(a2) (those elements of (R) that are near each other)

19. 3.73 Byrne Imaginary Situations

Cf: Make your own excuses
T: What will you say?
M: If he [] me, I'll say [].
a. b.
R: No finite response set
A: (a1)=small (verbs like asks, rings, etc., that will fit semantically)
(a2)=no finite answer set

20. 3.74 Byrne Meaningful Drill

P: PPP (Picture of room, showing people engaged in various tasks)
C: ccc(1) John's reading a book
M: MMM No, [] isn't. []'s [].
a. a. b.

R: (R1)=vs (he/she/it)
(R2)=vs (those actions listed in the picture)
A: aa1 he
(a2)=vs (watching television/watching TV/...)

30. 3.85 Candlin Implanting Skills: Memorizing Affective Language

P: ppp X is/are wrong if x say/says that, sir, honestly.
C: ccc(1) But the store detective says he watched you taking the tool.
M: MMM [][] wrong if [] [] that, sir, honestly.
a.b. a. c.

R: (R1)=vs (he/she/they)
(R2)=vs (is/'s/are/'re)
(R3)=vs (say/says)
A: aa1 he
(a2) (is/'s)
aa3 says

42. 3.98 Dakin Synonymy Drill

P: PPP He/she came on/by ----
C: ccc(1) Father walked here.
M: MMM [] came [] []
a. b. c.

R: (R1)=vs (He/she)
(R2)=vs (on/by)
(R3)=vs (foot/plane/boat/train...)
A: aa1 He
aa2 on
AA3 foot

43. 3.99 Dakin Hyponymy Drill

C: ccc(1) The old woman who lived in a shoe had a lot of sons and daughters.
M: MMM There [] so [] [] that [] couldn't [] [] all.
C: ccc(2) For dinner, she gave me a huge amount of rice and chicken.
M: MMM There [] so [] [] that [] couldn't [] [] all.
a. b. c. d. e. f.

R: (R) a. (was/were)
b. (many/much)
c. (group nouns)
d. (he/she/I/...)
e. (verbs that collocate with group nouns in c.)
f. (it/them)

A: (a)=vs

45. 3.101 Dakin Converse Drill

P: (Classroom reality)
C: Who is sitting on your left?
M: [] is
C+M: So you are sitting on Mary's [].

1. (R)=vs (People in the class)
aaa Mary
2. (R)=vs (left/right)
aaa left

46. 3.102 Dakin Consequence Drill

C: ccc(1) The cat has been killed.
M: MMM [] is now [].
a. b.
C: ccc(2) Frankenstein's monster has escaped.
M: MMM [] is now [].

R: (R1)=vs (it/he/she)
(r)=large (dead/killed/died/with God/in heaven...)
A: aa1 it
(a)=small (correct elements of (r)

MIXED OPERATIONS

No Examples

CHOICE OF TEMPLATE

14. 3.69 Byrne Meaningful Practice

P: (Eight Pictures of Objects)
Cf: Make a true statement in connection with the objects in the pictures.
M: A [] costs more than a [] or
a. b.
It is easier to [] a [] than to [] a [].
c. a. d. b.

R: (R1/R2)=vs (The set of objects in the picture)
(R3/R4)=small (The set of verbs that collocate with objects in the picture)
A: (a1/a2)=vs (Elements of (R) that make a true statement)
(a3/a4)=vs (The verbs that collocate with (a1/a2))

Appendix B Table 3.17b

Summary of Categorization of Pedagogical Sample according to Cue and Response-Model:

Part One: Simple Operations

I. Insertions into model

A. Insertion of cue into model:

2.	3.49	Robinett	Meaningful Drill
3.	3.50	Robinett	Communicative Drill
5.	3.159	Robinett	Communication Drill
6.	3.51	Paulston	Meaningless Drill
9.	3.55	Cook	Non-Contextualised
10.	3.56	Cook	Semi-Contexualised
37.	3.93	Stevick	Manipulative Drill
38.	3.94	Stevick	Meaningful Drill
47.	3.150	Beile	Non Communicative Interchange
51	3.154	Beile	Connected Communicative Interchange. situational context.

B. Insertion of secondary element into model

22.	3.76	Candlin	Get Organized: Concept
23.	3.77	Candlin	Get Organized: Narration
32.	3.87	Candlin	Implanting Skills: Restoration
35.	3.91	Candlin	Implanting Skills: Gap Text/New Text
40.	3.96	Dakin	General Knowledge
52.	3.155	Beile	No Contextualisation
53.	3.156	Beile	Dialog-like, but not Contextualisable
54.	3.157	Beile	Contextualisable

C. Insertion of element triggered by cue

1.	3.48	Robinett	Mechanical Drill
11.	3.57	Cook	Contextualised Drill
13.	3.68	Byrne	Mechanical Drill
16.	3.71	Byrne	Meaningful Practice
21.	3.75	Byrne	Meaningful Drill

24.	3.78	Candlin	Get Organized: true/false
27.	3.81	Candlin	Get Organized: Recognition
34.	3.90	Candlin	Implanting Skills: Gap Text
39.	3.95	Dakin	Application Drill
41.	3.97	Dakin	Collocation Drill
44.	3.100	Dakin	Antonymy Drill
55.	3.160	Stack	Meaningful Drill (unpatterned)
56.	3.160	Stack	Meaningful Drill (patterned)

D. <u>Insertion of element triggered by deletion of cue.</u>

No Examples

E. <u>Free Insertion</u>

17.	3.72	Byrne	Open Ended Responses
18.	3.73	Byrne	Imaginary Situations

II. <u>Removal of redundant elements</u>

No Examples

III. <u>Re-order of model elements</u>

28.	3.82	Candlin	Get Organized: Word Scramble

IV. <u>Production of new elements, based on cue</u>

7.	3.52	Paulston	Meaningless Drill
8.	3.53	Paulston	Meaningful Drill
12.	3.58	Cook	Situational Drill
25.	3.79	Candlin	Get Organized: Connecting Exercise
26.	3.80	Candlin	Get Organized: Multiple Choice
29.	3.84	Candlin	Get Organized: Word Scramble
31.	3.86	Candlin	Implanting Skills: Picture Stimulus
33.	3.89	Candlin	Implanting Skills: Restoration
36.	3.92	Candlin	Implanting Skills: Restoration

Appendix B   Table 3.17b

48.	3.151	Beile	Isolated Communicative Interchange
49.	3.152	Beile	Isolated Communicative Interchange in a situational context.
50.	3.153	Beile	Connected Communicative Interchange

Part Two: Complex Operations: More than one simple operation

MULTIPLE OPERATIONS:

4.	3.158	Robinett	Manipulative Drill
15.	3.70	Byrne	Meaningful Drill
19.	3.73	Byrne	Imaginary Situations
20.	3.74	Byrne	Meaningful Drill
30.	3.85	Candlin	Implanting Skills: Memorizing Affective Language
42.	3.98	Dakin	Synonymy Drill
43.	3.99	Dakin	Hyponymy Drill
45.	3.101	Dakin	Converse Drill
46.	3.102	Dakin	Consequence Drill

<u>MIXED OPERATIONS</u>

No Examples

<u>CHOICE OF TEMPLATE</u>

14.	3.69	Byrne	Meaningful Practice

Appendix B   Table   3.17c

Summary of Presentation & Answer-Analysis Factors in the Pedagogical Sample
(-Less activites characterized as "meaningless").

1. This table summarizes table 3.17

2. The symbols and conventi[illegible]ped in this table are the same as in tables 3.9 - 3.13:

P:	Prompt
C:	Cue
M:	Response-model
R:	Response-set
A:	Answer-set

PPP	represents text of prompt
TTT	represents text which does not figure operationally in the activity
ccc	represents text of cue.
CCF	represents formalized cue
MMM	represents universal response-model
mmm	represents item-specific response-model

{R}	stands for universal response-set
{r}	stands for item-specific response-set

aaa	stands for a single correct response
{a}	stands for an answer-set of more than one element

Sets are characterized as:

=vs	"very small"
=small	
=large	
=inf.	"infinite"

Appendix B Table 3.17c

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					Simple Operations			
					A. Insertion of cue into model:			
PPP	ccc	MMM	(R)=vs	aaa	2.	3.49	Robinett	Meaningful Drill
PPP	ccc	MMM	(R)=vs	aaa	3.	3.50	Robinett	Communicative Drill
	CCF	mmm	(R)=vs	aaa	5.	3.159	Robinett	Communication Drill
	ccc	mmm	(r)=0	aaa	10.	3.56	Cook	Semi-Contexualised Drill
TTT	ccc	mmm	(r)=vs	aaa	38.	3.94	Stevick	Meaningful Drill
TTT	ccc	MMM	(R)=vs	aaa	51.	3.154	Beile	Connected Communicative...
					B. Insertion of secondary element into model			
		mmm	(r)=large	(a)=large	22.	3.76	Candlin	Get Organized: Concept
PPP		mmm	(r)=large	(a)=large	23.	3.77	Candlin	Get Organized: Narration
PPP	CCF	mmm	(r)=small	aaa	32.	3.87	Candlin	Implanting Skills: Restoration
PPP		mmm	(R)=small	aaa	35.	3.91	Candlin	Implanting Skills: Gap Text/New Test
	CCF	mmm	(R)=large	(a)=small	40.	3.96	Dakin	General Knowledge Drill
ttt	CCF	mmm	(R)=small	(a)=vs	54.	3.157	Beile	Contextualisable
					C. Insertion of element triggered by cue			
	ccc	MMM	(R)=vs	(a)=vs	11.	3.57	Cook	Contextualised Drill
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice
	ccc	MMM	(R)=vs	(a)=vs	21.	3.75	Byrne	Meaningful Drill
PPP	CCF	MMM	(R)=vs	aaa	24.	3.78	Candlin	Get Organized: true/false
PPP	ccc	MMM	(R)=vs	(a)=vs	27.	3.81	Candlin	Get Organized: Recognition
	ccc	mmm	(r)=vs	aaa	34.	3.90	Candlin	Implanting Skills: Gap Text
	ccc	mmm	(r)=vs	(a)=vs	39.	3.95	Dakin	Application Drill
	ccc	MMM	(r)=vs	(a)=vs	41.	3.97	Dakin	Collocation Drill
	ccc	MMM	(R)=small	(a)=vs	44.	3.100	Dakin	Antonymy Drill
PPP	ccc	mmm	(R)=vs	aaa	55.	3.160	Stack	Meaningful Drill (unpatterned)
	ccc	MMM	(r)=vs	(a)=vs	56.	3.161	Stack	Meaningful Drill (patterned)
					E. Free Insertion			
		mmm	(r)=small	(a)=small	17.	3.72	Byrne	Open Ended Responses
PPP		MMM	(R)=large	(a)=large	18.	3.73.1	Byrne	Imaginary Situations
					III. Re-order of model elements			
PPP		MMM	(R)=small	aaa	28.	3.82	Candlin	Get Organized: Word Scramble

Appendix B Table 3.17c

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					IV. Production of new elements, based on cue			
	CCF		(r)=inf.	(a)=inf.	8.	3.53	Paulston	Meaningful Drill
	ccc							
PPP	ccc		(r)=small	(a)=vs	12.	3.58	Cook	Situational Drill
PPP	CCF				25.	3.79	Candlin	Get Organized: Connecting Exercise
PPP	CCF		(R)=inf.	(a)=vs	26.	3.80	Candlin	Get Organized: Multiple Choice
TTT	ccc(picture)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...
	ccc(text)							
	ccc(picture)		(R)=small	(a)=vs	31.	3.86	Candlin	Implanting Skills: Picture Stimulus
PPP	CCF		(R)=inf.	(a)=inf.	33.	3.89	Candlin	Implanting Skills: Restoration
PPP	ccc		(R)=small	(a)=vs	36.	3.92	Candlin	Implanting Skills: Restoration
	CCF		(R)=vs	aaa	48.	3.151	Beile	Isolated Communicative Interchange
	ccc							
TTT	CCF		(R)=small	(a)=small	49.	3.152	Beile	Isolated Communicative...
	ccc							
	ccc		(R)=small	(a)=vs	50.	3.153	Beile	Connected Communicative Interchange
					Complex Operations Multiple Operations			
PPP	CCF	MMM	(R)=vs	(a)=vs	15.	3.70	Byrne	Meaningful Drill
TTT	CCF	MMM	(R)=inf.	(al)=inf.	19.	3.73.2	Byrne	Imaginary Situations
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill
ppp	ccc	MMM	(R)=vs	(a)=vs	30.	3.85	Candlin	Implanting Skills: Memorizing Affective Language
ppp	ccc	MMM	(R)=vs	aaa	42.	3.98	Dakin	Synonomy Drill
	ccc	MMM	(R)=small	(a)=vs	43.	3.99	Dakin	Hyponymy Drill
	ccc	MMM	(R)=vs	aaa	45.	3.101	Dakin	Converse Drill
	ccc	MMM	(r)=large	(a)=small	46.	3.102	Dakin	Consequence Drill
					Choice of Template			
PPP	CCF	MMM	(R)=small	(a)=vs	14.	3.69	Byrne	Meaningful Practice

Appendix B: Table 3.18

All "Meaningful+" Drills Exhibiting only a Single Pedagogical Factor

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

Y = YES, definately exhibits this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
Y	-	-	-	-	-	-	-	-	3.92	Candlin	Implanting Skills: Restoration Defective Dialogue
-	-	[illegible]	[illegible]	-	-	Y	-	-	3.98	Dakin	Synonymy Drill
Y	-	-	[illegible]	[illegible]	-	-	-	-	3.151	Beile	Isolated Communicative Interchange
Y	-	-	-	-	-	-	-	-	3.153	Beile	Connected Communicative Interchange

Appendix B: Table 3.19

The Pedagogical Factors Exhibited by the "Meaningless" Drills in the Pedagogical Sample

1. This table should be interpreted with reference to section 3.12 of the text

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

Y = YES, definately exhibits this feature
y = yes, exhibits this feature to some minor degree

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	Author	Characterisation
-	-	-	-	-	-	-	-	-	3.48	Robinett	Mechanical Drill
-	-	-	-	-	-	-	-	-	3.51	Paulston	Meaningless Drill -1
-	-	-	-	-	-	-	-	-	3.52	Paulston	Meaningless Drill -2
-	-	-	-	-	-	-	-	-	3.55	Cook	Non-Contextualised Drill
-	-	-	-	-	-	-	-	-	3.68	Byrne	Mechanical Drill
Y	-	-	-	-	-	-	-	-	3.93	Stevick	Manipulative
-	-	-	-	-	-	-	-	-	3.150	Beile	Non-Communicative Interchange
-	-	-	-	-	-	Y	-	-	3.155	Beile	No Contextualisation
-	-	-	-	-	-	Y	-	-	3.156	Beile	Dialog-like, Not Contextualisable

Appendix B: Table 3.20

Pedagogical Factors Exhibited by Question-Answer Exchanges

1. This table should be interpreted with reference to section 3.12 of the text

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
Y	Y	Y	-	-	-	Y	-	-	3.49	Robinett
Y	Y	Y	-	-	-	Y	-	-	3.50	Robinett
Y	Y	-	-	y	-	-	-	-	3.57	Cook
Y	Y	Y	-	-	Y	Y	-	-	3.58	Cook
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin
Y	Y	Y	-	-	-	Y	-	-	3.101	Dakin
-	-	-	-	-	-	Y	-	-	3.156	Beile
Y	y	-	-	-	-	Y	-	-	3.157	Beile
Y	Y	Y	-	-	-	-	-	-	3.160	Stack

Note:

1. Beile provides 3.156 as an example of a question-answer interaction that is NOT realistic.

Appendix B   Table   3.20b

Operational Criteria of Drills Exhibiting a Question - Answer Interchange

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					Simple Operations			
					A. Insertion of cue into model:			
PPP	ccc	MMM	(R)=vs	aaa	2.	3.49	Robinett	Meaningful Drill
PPP	ccc	MMM	(R)=vs	aaa	3.	3.50	Robinett	Communicative Drill
					B. Insertion of secondary element into model			
ttt	CCF	aaa	(R)=small	(a)=vs	54.	3.157	Beile	Contextualisable
					C. Insertion of element triggered by cue			
	ccc	MMM	(R)=vs	(a)=vs	11.	3.57	Cook	Contextualised Drill
PPP	ccc	aaa	(R)=vs	aaa	55.	3.160	Stack	Meaningful Drill (unpatterned)
					IV. Production of new elements, based on cue			
PPP	ccc		(r)=small	(a)=vs	12.	3.58	Cook	Situational Drill
TTT	ccc(picture) ccc(text)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...
					Complex Operations			
					Multiple Operations			
	ccc	MMM	(R)=vs	aaa	45.	3.101	Dakin	Converse Drill

Appendix B: Table 3.21

Examples that Exhibit Realistic Interchange

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
Y	Y	Y	-	-	-	Y	-	-	3.49	Robinett
Y	Y	Y	-	-	-	Y	-	-	3.50	Robinett
Y	Y	-	-	-	-	-	-	-	3.159	Robinett
Y	-	-	-	Y	-	-	-	-	3.56	Cook
Y	Y	-	-	y	-	-	-	-	3.57	Cook
Y	Y	Y	-	-	Y	Y	-	-	3.58	Byrne
Y	Y	-	Y	-	Y	Y	-	Y	3.71	Byrne
Y	Y	-	y	-	Y	-	Y	-	3.73.1	Byrne
Y	Y	-	-	Y	Y	-	-	Y	3.73.2	Byrne
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne
Y	Y	-	-	-	Y	Y	-	-	3.75	Byrne
Y	Y	Y	-	-	Y	-	Y	-	3.79	Candlin
Y	Y	Y	-	-	-	-	-	-	3.82	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin
Y	Y	-	-	y	-	-	-	-	3.85	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.86	Candlin
Y	Y	Y	-	-	-	-	-	-	3.91	Candlin
Y	-	-	-	-	-	-	-	-	3.92	Candlin
Y	-	-	-	-	-	-	-	-	3.93	Stevick
Y	Y	-	-	-	-	-	-	-	3.94	Stevick
Y	-	-	-	-	Y	Y	Y	Y	3.97	Dakin
Y	Y	Y	-	-	-	Y	-	-	3.101	Dakin
Y	-	-	-	-	-	-	-	-	3.151	Beile
Y	Y	-	-	Y	-	-	-	-	3.152	Beile
Y	-	-	-	-	-	-	-	-	3.153	Beile
Y	Y	-	-	Y	-	-	-	-	3.154	Beile
Y	y	-	-	-	-	Y	-	-	3.157	Beile
Y	Y	Y	-	-	-	-	-	-	3.160	Stack
Y	-	-	-	-	Y	y	Y	-	3.161	Stack

Appendix B Table 3.21b

Operational Criteria of Drills Exhibiting a Realistic Interchange

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					Simple Operations			
					A. Insertion of cue into model:			
PPP	ccc	MMM	(R)=vs	aaa	2.	3.49	Robinett	Meaningful Drill
PPP	ccc	MMM	(R)=vs	aaa	3.	3.50	Robinett	Communicative Drill
	CCF	mmm	(R)=vs	aaa	5.	3.159	Robinett	Communication Drill
	ccc	mmm	(r)=0	aaa	10.	3.56	Cook	Semi-Contexualised Drill
	ccc	MMM	(r)=vs	aaa	37.	3.93	Stevick	Manipulative Drill
TTT	ccc	MMM	(r)=vs	aaa	38.	3.94	Stevick	Meaningful Drill
					B. Insertion of secondary element into model			
PPP		mmm	(R)=small	aaa	35.	3.91	Candlin	Implanting Skills: Gap Text/New Test
ttt	CCF	mmm	(R)=small	(a)=vs	54.	3.157	Beile	Contextualisable
					C. Insertion of element triggered by cue			
	ccc	MMM	(R)=vs	(a)=vs	11.	3.57	Cook	Contextualised Drill
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice
PPP		MMM	(R)=large	(a)=large	18.	3.73.1	Byrne	Imaginary Situations
	ccc	MMM	(R)=vs	(a)=vs	21.	3.75	Byrne	Meaningful Drill
	ccc	MMM	(r)=vs	(a)=vs	41.	3.97	Dakin	Collocation Drill
PPP	ccc	mmm	(R)=vs	aaa	55.	3.160	Stack	Meaningful Drill (unpatterned)
	ccc	MMM	(r)=vs	(a)=vs	56.	3.161	Stack	Meaningful Drill (patterned)
					III. Re-order of model elements			
PPP		MMM	(R)=small	aaa	28.	3.82	Candlin	Get Organized: Word Scramble

Appendix B Table 3.21b

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					IV. Production of new elements, based on cue :			
PPP	ccc		(r)=small	(a)=vs	12.	3.58	Cook	Situational Drill
PPP	CCF				25.	3.79	Candlin	Get Organized: Connecting Exercise
TTT	ccc(picture)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...
	ccc(text)							
	ccc(picture)		(R)=small	(a)=vs	31.	3.86	Candlin	Implanting Skills: Picture Stimulus
PPP	ccc		(R)=small	(a)=vs	36.	3.92	Candlin	Implanting Skills: Restoration
	CCF		(R)=vs	aaa	48.	3.151	Beile	Isolated Communicative Interchange
	ccc							
TTT	CCF		(R)=small	(a)=small	49.	3.152	Beile	Isolated Communicative...
	ccc							
	ccc		(R)=small	(a)=vs	50.	3.153	Beile	Connected Communicative Interchange
					Complex Operations Multiple Operations			
TTT	CCF	MMM	(R)=inf.	(al)=inf.	19.	3.73.2	Byrne	Imaginary Situations
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill
ppp	ccc	MMM	(R)=vs	(a)=vs	30.	3.85	Candlin	Implanting Skills:
	ccc	MMM	(R)=vs	aaa	45.	3.101	Dakin	Converse Drill

Appendix B: Table 3.22

Examples Which Exhibit Extended Context

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
Where context has no effect on correct response										
Y	Y	-	-	-	-	-	-	-	3.159	Robinett
Y	Y	-	-	Y	-	-	-	-	3.57	Cook
Y	Y	-	-	Y	Y	-	-	Y	3.73.2	Byrne
Y	Y	-	-	-	Y	Y	-	-	3.75	Byrne
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin
Y	Y	-	-	Y	-	-	-	-	3.85	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.86	Candlin
-	Y	Y	-	-	-	-	-	-	3.90	Candlin
Y	Y	-	-	-	-	-	-	-	3.94	Stevick
Y	Y	-	-	Y	-	-	-	-	3.152	Beile
Y	Y	-	-	Y	-	-	-	-	3.154	Beile
Y	Y	-	-	-	-	Y	-	-	3.157	Beile
Where context does have an effect on correct response										
Y	Y	Y	-	-	-	Y	-	-	3.50	Robinett
Y	Y	Y	-	-	Y	Y	-	-	3.58	Cook
-	Y	Y	-	-	Y	Y	-	-	3.70	Byrne
Y	Y	-	Y	-	Y	Y	-	Y	3.71	Byrne
Y	Y	-	Y	-	Y	-	Y	-	3.73.1	Byrne
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne
-	Y	-	-	-	-	Y	-	-	3.76	Candlin
-	Y	Y	-	-	-	-	-	-	3.77	Candlin
-	Y	Y	-	-	-	-	-	-	3.78	Candlin
Y	Y	Y	-	-	Y	-	Y	-	3.79	Candlin
-	Y	Y	-	-	-	-	-	-	3.80	Candlin
-	Y	-	-	-	-	Y	-	-	3.81	Candlin
Y	Y	Y	-	-	-	-	-	-	3.82	Candlin

Appendix B: Table 3.22

-	Y	-	-	-	-	Y	-	-	3.87	Candlin
-	Y	-	-	-	Y	-	-	Y	3.89	Candlin
Y	Y	Y	-	-	-	-	-	-	3.91	Candlin
Y	Y	Y	-	-	-	Y	-	-	3.101	Dakin
Y	Y	Y	-	-	-	-	-	-	3.160	Stack

Appendix B: Table 3.22b

Operational Criteria of Drills Exhibiting Extended Context

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions included in this table is found in table 3.17c.

Presentation			Answer Analysis		
P:	C:	M:	R:	A:	Categories

I. Where context has no affect on response:

P:	C:	M:	R:	A:	Categories			
					Simple Operations			
					A. Insertion of cue into model:			
	CCF	mmm	(R)=vs	aaa	5.	3.159	Robinett	Communication Drill
							Context:	Simulated dialogue about breakfast
TTT	ccc	mmm	(r)=vs	aaa	38.	3.94	Stevick	Meaningful Drill
							Context:	Grocery store topics
TTT	ccc	MMM	(R)=vs	aaa	51.	3.154	Beile	Connected Communicative...
							Context:	Visiting a sick friend
					B. Insertion of secondary element into model			
ttt	CCF	mmm	(R)=small	(a)=vs	54.	3.157	Beile	Contextualisable
							Context:	Conversation about buying food
					C. Insertion of element triggered by cue			
	ccc	MMM	(R)=vs	(a)=vs	11.	3.57	Cook	Contextualised Drill
							Context:	Simulated dialogue about party
	ccc	MMM	(R)=vs	(a)=vs	21.	3.75	Byrne	Meaningful Drill
							Context:	A hard day
	ccc	mmm	(r)=vs	aaa	34.	3.90	Candlin	Implanting Skills: Gap Text
							Context:	Paragraph about drunk coming

Appendix B: Table 3.22b

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
								home
					IV. Production of new elements, based on cue			
TTT	ccc(picture) ccc(text)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...
							Context:	Spring cleaning day
	ccc(picture)		(R)=small	(a)=vs	31.	3.86	Candlin	Implanting Skills: Picture Stimulus
							Context:	Simulated dialogue about cat
TTT	CCF ccc		(R)=small	(a)=small	49.	3.152	Beile	Isolated Communicative... Interchange
							Context:	Difficult conversation with Colin
					Complex Operations Multiple Operations			
ppp	ccc	MMM	(R)=vs	(a)=vs	30.	3.85	Candlin	Implanting Skills: Memorizing Affective Language
							Context:	Someone is accused of a crime
TTT	CCF	MMM	(R)=inf.	(a)=inf.	19.	3.73.2	Byrne	Imaginary Situations
							Context:	George ringing to set up meeting

II. Where context has an affect on the response

					Simple Operations			
					A. Insertion of cue into model:			
PPP	ccc	MMM	(R)=vs	aaa	3.	3.50	Robinett	Communicative Drill
							Context:	The classroom situation
					B. Insertion of secondary element into model			
		mmm	(r)=large	(a)=large	22.	3.76	Candlin	Get Organized: Concept
							Context:	The domain of shoplifting
PPP		mmm	(r)=large	(a)=large	23.	3.77	Candlin	Get Organized: Narration
							Context:	"The Mystery Object Lands"
PPP	CCF	mmm	(r)=small	aaa	32.	3.87	Candlin	Implanting Skills: Restoration
							Context:	Paragraph
PPP		mmm	(R)=small	aaa	35.	3.91	Candlin	Implanting Skills: Gap Text/New Test
							Context:	Caroline's Diary

Appendix B: Table 3.22b

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					C. Insertion of element triggered by cue			
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice
							Context:	Objects in a room
PPP	CCF	MMM	(R)=vs	aaa	24.	3.78	Candlin	Get Organized: true/false
							Context:	A dialogue about flying saucers
PPP	ccc	MMM	(R)=vs	(a)=vs	27.	3.81	Candlin	Get Organized: Recognition
							Context:	Horror story, with sound effects
PPP	ccc	mmm	(R)=vs	aaa	55.	3.160	Stack	Meaningful Drill (unpatterned)
							Context:	Robert's visit to a restaurant

Appendix B: Table 3.23

Examples Which Exhibit Truth Value or Information Gap

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
+(Q-stimulus/A-response)										
Y	Y	Y	–	–	–	Y	–	–	3.49	Robinett
Y	Y	Y	–	–	–	Y	–	–	3.50	Robinett
–	Y	Y	–	–	–	–	–	–	3.78	Candlin
Y	Y	Y	–	–	Y	Y	–	–	3.58	Cook
Y	Y	Y	–	–	–	Y	–	–	3.101	Dakin
Y	Y	Y	–	–	–	–	–	–	3.160	Stack
+(Q-response/A-feedback)										
Y	Y	–	Y	–	Y	Y	–	Y	3.71	Byrne
Y	Y	–	y	–	Y	–	Y	–	3.73.1	Byrne
(No question)										
–	–	Y	–	–	Y	Y	–	–	3.69	Byrne
–	Y	Y	–	–	Y	Y	–	–	3.70	Byrne
Y	Y	Y	–	Y	Y	Y	Y	–	3.74	Byrne
–	–	Y	–	–	Y	Y	Y	–	3.95	Dakin
–	–	Y	–	–	Y	Y	Y	–	3.96	Dakin
–	Y	Y	–	–	–	–	–	–	3.77	Candlin
Y	Y	Y	–	–	Y	–	Y	–	3.79	Candlin
–	Y	Y	–	–	–	–	–	–	3.80	Candlin
Y	Y	Y	–	–	–	–	–	–	3.82	Candlin
–	Y	Y	–	–	–	–	–	–	3.90	Candlin
Y	Y	Y	–	–	–	–	–	–	3.91	Candlin

Notes:

1. Example 3.78 (Candlin) approximates a Q-A interchange and so is included in this table.

Appendix B   Table 3.23b

Operational Criteria of Drills Exhibiting Truth Value or Information Gap

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis					
P:	C:	M:	R:	A:	Categories			
					Simple Operations			
					A. Insertion of cue into model:			
PPP	ccc	MMM	(R)=vs	aaa	2.	3.49	Robinett	Meaningful Drill
PPP	ccc	MMM	(R)=vs	aaa	3.	3.50	Robinett	Communicative Drill
					B. Insertion of secondary element into model			
PPP		mmm	(r)=large	(a)=large	23.	3.77	Candlin	Get Organized: Narration
PPP		mmm	(R)=small	aaa	35.	3.91	Candlin	Implanting Skills: Gap Text/New Test
	CCF	mmm	(R)=large	(a)=small	40.	3.96	Dakin	General Knowledge Drill
					C. Insertion of element triggered by cue			
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice
PPP		MMM	(R)=large	(a)=large	18.	3.73.1	Byrne	Imaginary Situations
PPP	CCF	MMM	(R)=vs	aaa	24.	3.78	Candlin	Get Organized: true/false
	ccc	mmm	(r)=vs	aaa	34.	3.90	Candlin	Implanting Skills: Gap Text
	ccc	mmm	(r)=vs	(a)=vs	39.	3.95	Dakin	Application Drill
PPP	ccc	mmm	(R)=vs	aad	55.	3.160	Stack	Meaningful Drill (unpatterned)
					III. Re-order of model elements			
PPP		MMM	(R)=small	aaa	28.	3.82	Candlin	Get Organized: Word Scramble
					IV. Production of new elements, based on cue			
PPP	ccc		(r)=small	(a)=vs	12.	3.58	Cook	Situational Drill
PPP	CCF				25.	3.79	Candlin	Get Organized: Connecting Exercise
PPP	CCF		(R)=inf.	(a)=vs	26.	3.80	Candlin	Get Organized: Multiple Choice
					Complex Operations			
					Multiple Operations			

Appendix B Table 3.23b

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
PPP	CCF	MMM	(R)=vs	(a)=vs	15.	3.70	Byrne	Meaningful Drill
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill
	ccc	MMM	(R)=vs	aaa	45.	3.101	Dakin	Converse Drill
					Choice of Template			
PPP	CCF	MMM	(R)=small	(a)=vs	14.	3.69	Byrne	Meaningful Practice

Appendix B: Table 3.24

Examples that Exhibit Function Practice

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
Y	Y	-	-	Y	-	-	-	-	3.152	Beile
Y	Y	-	-	Y	-	-	-	-	3.154	Beile
Y	-	-	-	Y	-	-	-	-	3.56	Cook
Y	Y	-	-	y	-	-	-	-	3.57	Cook
Y	Y	-	-	Y	Y	-	-	Y	3.73.2	Byrne
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne
Y	Y	-	-	y	-	-	-	-	3.85	Candlin

Appendix B Table 3.24b

Operational Criteria of Drills Exhibiting Function Practice

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17a.

Presentation			Answer Analysis					
P:	C:	M:	R:	A:	Categories			
					Simple Operations			
					A. Insertion of cue into model:			
	ccc	mmm	(r)=0	aaa	10.	3.56	Cook	Semi-Contexualised Drill
TTT	ccc	MMM	(R)=vs	aaa	51.	3.154	Beile	Connected Communicative...
					C. Insertion of element triggered by cue			
	ccc	MMM	(R)=vs	(a)=vs	11.	3.57	Cook	Contextualised Drill
					IV. Production of new elements, based on cue			
TTT	CCF ccc		(R)=small	(a)=small	49.	3.152	Beile	Isolated Communicative...
					Complex Operations			
					Multiple Operations			
TTT	CCF	MMM	(R)=inf.	(a1)=inf.	19.	3.73.2	Byrne	Imaginary Situations
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill
ppp	ccc	MMM	(R)=vs	(a)=vs	30.	3.85	Candlin	Implanting Skills: Memorizing Affective Language

Appendix B: Table 3.25

Examples that Exhibit More than One Response

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
-	-	-	-	-	Y	y	-	Y	3.53	Paulston
Y	Y	Y	-	-	Y	Y	-	-	3.58	Cook
-	-	Y	-	-	Y	Y	-	-	3.69	Byrne
-	Y	Y	-	-	Y	Y	-	-	3.70	Byrne
Y	Y	-	Y	-	Y	Y	-	Y	3.71	Byrne
-	-	-	-	-	Y	y	Y	Y	3.72	Byrne
Y	Y	-	y	-	Y	-	Y	-	3.73.1	Byrne
Y	Y	-	-	Y	Y	-	-	Y	3.73.2	Byrne
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne
Y	Y	-	-	-	Y	Y	-	-	3.75	Byrne
Y	Y	Y	-	-	Y	-	Y	-	3.79	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.86	Candlin
-	Y	-	-	-	Y	-	-	Y	3.89	Candlin
-	-	Y	-	-	Y	Y	Y	-	3.95	Dakin
-	Y	-	-	Y	Y	Y	-		3.96	Dakin
Y	-	-	-	-	Y	Y	Y	Y	3.97	Dakin
-	-	-	-	-	Y	Y	Y	-	3.99	Dakin
-	-	-	-	-	Y	Y	-	-	3.100	Dakin
-	-	-	-	-	Y	Y	Y	-	3.102	Dakin
Y	-	-	-	-	Y	y	Y	-	3.161	Stack

Appendix B Table 3.25b

Operational Criteria of Drills Exhibiting More than One Response

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis						
P:	C:	M:	R:	A:	Categories				
					B. Insertion of secondary element into model				
		mmm	(r)=large	(a)=large	22.	3.76	Candlin	Get Organized: Concept	
PPP		mmm	(r)=large	(a)=large	23.	3.77	Candlin	Get Organized: Narration	
	CCF	mmm	(R)=large	(a)=small	40.	3.96	Dakin	General Knowledge Drill	
ttt	CCF	mmm	(R)=small	(a)=vs	54.	3.157	Beile	Contextualisable	
					C. Insertion of element triggered by cue				
	ccc	MMM	(R)=vs	(a)=vs	11.	3.57	Cook	Contextualised Drill	
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice	
	ccc	MMM	(R)=vs	(a)=vs	21.	3.75	Byrne	Meaningful Drill	
PPP	ccc	MMM	(R)=vs	(a)=vs	27.	3.81	Candlin	Get Organized: Recognition	
	ccc	mmm	(r)=vs	(a)=vs	39.	3.95	Dakin	Application Drill	
	ccc	MMM	(r)=vs	(a)=vs	41.	3.97	Dakin	Collocation Drill	
	ccc	MMM	(R)=small	(a)=vs	44.	3.100	Dakin	Antonymy Drill	
	ccc	MMM	(r)=vs	(a)=vs	56.	3.161	Stack	Meaningful Drill (patterned)	
					E. Free Insertion				
		mmm	(r)=small	(a)=small	17.	3.72	Byrne	Open Ended Responses	
PPP		MMM	(R)=large	(a)=large	18.	3.73.1	Byrne	Imaginary Situations	
					IV. Production of new elements, based on cue				
	CCF		(r)=inf.	(a)=inf.	8.	3.53	Paulston	Meaningful Drill	
	ccc								
PPP	ccc		(r)=small	(a)=vs	12.	3.58	Cook	Situational Drill	
PPP	CCF				25.	3.79	Candlin	Get Organized: Connecting Exercise	
PPP	CCF		(R)=inf.	(a)=vs	26.	3.80	Candlin	Get Organized: Multiple Choice	
TTT	ccc(picture)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...	
	ccc(text)								
	ccc(picture)		(R)=small	(a)=vs	31.	3.86	Candlin	Implanting Skills: Picture Stimulus	

Appendix B Table 3.25b

Presentation			Answer Analysis					
P:	C:	M:	R:	A:	Categories			
PPP	CCF		(R)=inf.	(a)=inf.	33.	3.89	Candlin	Implanting Skills: Restoration
PPP	ccc		(R)=small	(a)=vs	36.	3.92	Candlin	Implanting Skills: Restoration
TTT	CCF		(R)=small	(a)=small	49.	3.152	Beile	Isolated Communicative...
	ccc							
	ccc		(R)=small	(a)=vs	50.	3.153	Beile	Connected Communicative Interchange
					Complex Operations			
					Multiple Operations			
PPP	CCF	MMM	(R)=vs	(a)=vs	15.	3.70	Byrne	Meaningful Drill
TTT	CCF	MMM	(R)=inf.	(al)=inf.	19.	3.73.2	Byrne	Imaginary Situations
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill
ppp	ccc	MMM	(R)=vs	(a)=vs	30.	3.85	Candlin	Implanting Skills: Memorizing Affective Language
	ccc	MMM	(R)=small	(a)=vs	43.	3.99	Dakin	Hyponymy Drill
	ccc	MMM	(r)=large	(a)=small	46.	3.102	Dakin	Consequence Drill
					Choice of Template			
PPP	CCF	MMM	(R)=small	(a)=vs	14.	3.69	Byrne	Meaningful Practice

Appendix B: Table 3.26

Examples that Exhibit the Need for Lexical Skill

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
Y	Y	Y	-	-	-	Y	-	-	3.49	Robinett
Y	Y	Y	-	-	-	Y	-	-	3.50	Robinett
-	-	-	-	-	Y	y	-	Y	3.53	Paulston
Y	Y	Y	-	-	Y	Y	-	-	3.58	Cook
-	-	Y	-	-	Y	Y	-	-	3.69	Byrne
-	Y	Y	-	-	Y	Y	-	-	3.70	Byrne
Y	Y	-	Y	-	Y	Y	-	Y	3.71	Byrne
-	-	-	-	-	Y	y	Y	Y	3.72	Byrne
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne
Y	Y	-	-	-	Y	Y	-	-	3.75	Byrne
-	Y	-	-	-	-	Y	-	-	3.76	Candlin
-	Y	-	-	-	-	Y	-	-	3.81	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.86	Candlin
-	Y	-	-	-	-	Y	-	-	3.87	Candlin
-	-	Y	-	-	Y	Y	Y	-	3.95	Dakin
-	-	Y	-	-	Y	Y	Y	-	3.96	Dakin
Y	-	-	-	-	Y	Y	Y	Y	3.97	Dakin
-	-	-	-	-	-	Y	-	-	3.98	Dakin
-	-	-	-	-	Y	Y	Y	-	3.99	Dakin
-	-	-	-	-	Y	Y	-	-	3.100	Dakin
Y	Y	Y	-	-	-	Y	-	-	3.101	Dakin
-	-	-	-	-	Y	Y	Y	-	3.102	Dakin
-	-	-	-	-	-	Y	-	-	3.155	Beile
-	-	-	-	-	-	Y	-	-	3.156	Beile
Y	y	-	-	-	-	Y	-	-	3.157	Beile
Y	-	-	-	-	Y	y	Y	-	3.161	Stack

Appendix B Table 3.26b

Operational Criteria of Drills Exhibiting the Need for Lexical Skill

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis					
P:	C:	M:	R:	A:	Categories			
					Simple Operations			
					A. Insertion of cue into model:			
PPP	ccc	MMM	(R)=vs	aaa	2.	3.49	Robinett	Meaningful Drill
PPP	ccc	MMM	(R)=vs	aaa	3.	3.50	Robinett	Communicative Drill
					B. Insertion of secondary element into model			
		mmm	(r)=large	(a)=large	22.	3.76	Candlin	Get Organized: Concept
PPP	CCF	mmm	(r)=small	aaa	32.	3.87	Candlin	Implanting Skills: Restoration
	CCF	mmm	(R)=large	(a)=small	40.	3.96	Dakin	General Knowledge Drill
ttt	CCF	mmm	(R)=small	(a)=vs	54.	3.157	Beile	Contextualisable
					C. Insertion of element triggered by cue			
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice
	ccc	MMM	(R)=vs	(a)=vs	21.	3.75	Byrne	Meaningful Drill
PPP	ccc	MMM	(R)=vs	(a)=vs	27.	3.81	Candlin	Get Organized: Recognition
	ccc	mmm	(r)=vs	(a)=vs	39.	3.95	Dakin	Application Drill
	ccc	MMM	(r)=vs	(a)=vs	41.	3.97	Dakin	Collocation Drill
	ccc	MMM	(R)=small	(a)=vs	44.	3.100	Dakin	Antonymy Drill
	ccc	MMM	(r)=vs	(a)=vs	56.	3.161	Stack	Meaningful Drill (patterned)
					E. Free Insertion			
		mmm	(r)=small	(a)=small	17.	3.72	Byrne	Open Ended Responses

Presentation			Answer Analysis						
P:	C:	M:	R:	A:	Categories				
					IV. Production of new elements, based on cue				
	CCF		(r)=inf.	(a)=inf.	8.	3.53	Paulston	Meaningful Drill	
	ccc								
PPP	ccc		(r)=small	(a)=vs	12.	3.58	Cook	Situational Drill	
TTT	ccc (picture)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...	
	ccc (text)								
	ccc (picture)		(R)=small	(a)=vs	31.	3.86	Candlin	Implanting Skills: Picture Stimulus Interchange	
					Complex Operations				
					Multiple Operations				
PPP	CCF	MMM	(R)=vs	(a)=vs	15.	3.70	Byrne	Meaningful Drill	
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill	
ppp	ccc	MMM	(R)=vs	aaa	42.	3.98	Dakin	Synonomy Drill	
	ccc	MMM	(R)=small	(a)=vs	43.	3.99	Dakin	Hyponymy Drill	
	ccc	MMM	(R)=vs	aaa	45.	3.101	Dakin	Converse Drill	
	ccc	MMM	(r)=large	(a)=small	46.	3.102	Dakin	Consequence Drill	
					Choice of Template				
PPP	CCF	MMM	(R)=small	(a)=vs	14.	3.69	Byrne	Meaningful Practice	

Appendix B: Table 3.27

Examples that Provide the Possibility of Creative Input, Within Constraints

1. This table should be interpreted with reference to section 3.12 of the text.

1.	Realistic Interchange	6.	More than One Response
2.	Extented Context	7.	Lexical Understanding Required
3.	Truth Value	8.	New Elements, within Constraints
4.	Information Gap	9.	New Elements, Free
5.	Function Practice		

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
-	-	-	-	-	Y	Y	Y	Y	3.72	Byrne
Y	Y	Y	-	Y	Y	Y	Y	-	3.74	Byrne
Y	Y	Y	-	-	Y	-	Y	-	3.79	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.84	Candlin
Y	Y	-	-	-	Y	Y	Y	-	3.86	Candlin
-	-	Y	-	-	Y	Y	Y	-	3.95	Dakin
-	-	Y	-	-	Y	Y	Y	-	3.96	Dakin
Y	-	-	-	-	Y	Y	Y	Y	3.97	Dakin
-	-	-	-	-	Y	Y	Y	-	3.99	Dakin
-	-	-	-	-	Y	Y	Y	-	3.102	Dakin
Y	-	-	-	-	Y	Y	Y	-	3.161	Stack

Appendix B   Table   3.27b

Operational Criteria of Drills Providing for Creative Input, Within Constraints

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis		Categories			
P:	C:	M:	R:	A:				
					Simple Operations			
					B. Insertion of secondary element into model			
	CCF	mmm	(R)=large	(a)=small	40.	3.96	Dakin	General Knowledge Drill
					C. Insertion of element triggered by cue			
	ccc	mmm	(r)=vs	(a)=vs	39.	3.95	Dakin	Application Drill
	ccc	MMM	(r)=vs	(a)=vs	41.	3.97	Dakin	Collocation Drill
	ccc	MMM	(r)=vs	(a)=vs	56.	3.161	Stack	Meaningful Drill (patterned)
					E. Free Insertion			
		mmm	(r)=small	(a)=small	17.	3.72	Byrne	Open Ended Responses
					IV. Production of new elements, based on cue			
PPP	CCF				25.	3.79	Candlin	Get Organized: Connecting Exercise
TTT	ccc(picture)		(R)=small	(a)=vs	29.	3.84	Candlin	Implanting Skills:...
	ccc(text)							
	ccc(picture)		(R)=small	(a)=vs	31.	3.86	Candlin	Implanting Skills: Picture Stimulus
					Complex Operations			
					Multiple Operations			
PPP	ccc	MMM	(R)=vs	(a)=vs	20.	3.74	Byrne	Meaningful Drill
	ccc	MMM	(R)=small	(a)=vs	43.	3.99	Dakin	Hyponymy Drill
	ccc	MMM	(r)=large	(a)=small	46.	3.102	Dakin	Consequence Drill

Appendix B: Table 3.20

Examples that Provide the Possibility of Creative Input, Without Constraints

1. This table should be interpreted with reference to section 3.12 of the text.

---

1. Realistic Interchange
2. Extended Context
3. Truth Value
4. Information Gap
5. Function Practice
6. More than One Response
7. Lexical Understanding Required
8. New Elements, within Constraints
9. New Elements, Free

1.	2.	3.	4.	5.	6.	7.	8.	9.	Reference	
–	–	–	–	–	Y	Y	–	Y	3.53	Paulston
Y	Y	–	Y	–	Y	Y	–	Y	3.71	Byrne
–	–	–	–	–	Y	Y	Y	Y	3.72	Byrne
Y	Y	–	–	Y	Y	–	–	Y	3.73.2	Byrne
–	Y	–	–	–	Y	–	–	Y	3.89	Candlin

Appendix B Table 3.28b

Operational Criteria of Drills Allowing Creative Input, Without Constraints

1. This table should be interpreted with reference to section 3.13 of the text.

2. The key to the symbols and conventions used in this table is found in table 3.17c.

Presentation			Answer Analysis					
P:	C:	M:	R:	A:	Categories			
					Simple Operations			
					C. Insertion of element triggered by cue			
	CCF	MMM	(R)=large	(a)=small	16.	3.71	Byrne	Meaningful Practice
					E. Free Insertion			
		mmm	(r)=small	(a)=small	17.	3.72	Byrne	Open Ended Responses
					IV. Production of new elements, based on cue			
	CCF		(r)=inf.	(a)=inf.	8.	3.53	Paulston	Meaningful Drill
PPP	CCC CCF		(R)=inf.	(a)=inf.	33.	3.89	Candlin	Implanting Skills: Restoration Interchange
					Complex Operations Multiple Operations			
TTT	CCF	MMM	(R)=inf.	(a)=inf.	19.	3.73.2	Byrne	Imaginary Situations