

## Chapter 5

### CELL in a Language Learning Environment

#### Framework for the Architecture

##### 5.1 Introduction

In order to design and construct a Computer-Enhanced Language Learning environment, it is necessary to be clear on exactly what is meant by each aspect of this construct and how these are realised in practical terms in a software application. A wide range of approaches to instructional design has been implemented, and it is informative in a new design to examine some of these projects, to determine to what extent they fulfil principles of learner-centred design model as defined here. This examination needs to include an overview of good practice in the positioning and operation of navigation features, visual screen presentation, the nature and timing of the presentation of, or access to, help and feedback, and views on the role of the learner in the CELL process.

This last point is more critical in the design of a CELL environment than in other teaching and learning environment, because of the additional factor of technology. In *MMInteraktif*, the name given to the software package developed as part of this work, the role of the learner is critical. This is because, as elaborated in the previous chapter, in order for learners to navigate successfully through a self-access CELL package, they need to have the skills and strategies to make informed decisions about their learning paths through the material, and the package also needs to cater for learners making these decisions.

The major issue then becomes, as introduced in Chapter 1, whether the technology package is technology-driven, pedagogy-driven, or learner-driven. In a technology-driven package, for example, the deciding factors in instructional design are the features and capabilities of the technology, while a pedagogy-driven package is based on principles of teaching practice which have been found to be effective. A learner-driven design, however, incorporates the best aspects of pedagogy while keeping the learners' needs and learner control as the focus. Nevertheless, computers have not always been viewed favourably by teachers and learners. In this, computers share the fate of other technological aids and resources that have been introduced to classrooms in the past.

While the use of computers in the classroom, or for language learning, is still relatively new, the use of technology has a role that is fairly well-established, if not always well-defined or well-received. As Davies and Higgins (1985) noted several years ago, for technology to be accepted and used by teachers, it is necessary for teachers to feel that it is fulfilling a useful role:

The lack of enthusiasm amongst language teachers for the new technology may also be due to the disappointing results achieved with technical aids so far. The language laboratory does not have a good track record, and many teachers are sceptical about new gadgets like the videocassette recorder. This is a pity, as technical aids can be of immense value in the classroom, providing of course that the teacher knows how to use them properly and how to integrate them into a teaching programme.

(Davies & Higgins, 1985:1 - 2)

While on the topic of learner involvement, Higgins and Johns maintain that:

[...] it can be argued that packages, with their assumptions about the order in which tasks are to be undertaken and the time which should be spent, are taking decisions that should be made by the teacher and the student, particularly the student.

(Higgins & Johns, 1984:86)

These two quotes seem to epitomise some of the problems perceived with the role of computers in language learning, and some of the points to be borne in mind when designing any CELL activities. The key point that Higgins and Johns make is that *students* should have the major part of control over the timing and direction of their learning activities. By substituting ‘computer’ for ‘videocassette recorder’ in the second quote, the reminder is clearly that technology in learning is useful only insofar as it is handled competently by teachers and integrated into the teaching and learning program as a whole.

In light of the conclusions reached in the previous chapters, language learning, whether mediated by computers, teachers, or other people, needs to be learner-centred, culturally embedded, and goal-directed, and the instructional design needs to reflect this. This chapter will begin with an elaboration on the role of learner-centred methodology in CELL first introduced in Chapter 1, leading to the presentation of a model developed by this author for the integration of CELL in a learner-centred language program, or *language learning environment*. This will include a discussion of some models for the incorporation of computers into language learning.

Subsequently, a theoretical overview of approaches to instructional design will be presented, using the methodological framework for CALL courseware developed by Hubbard (1992). This will include an analysis of the best features of other instructional design approaches, and an identification of their benefits, leading to a framework for the

architecture of the *MMInteraktif* system to follow. At each stage in the presentation of this overview, illustrations and exemplars will be provided from the listening and viewing software package called *MMInteraktif*, designed and developed as part of this current work, together with a discussion of the learner management of the program by means of ‘transparent’ (Gordon, S., 1994) instructional design features. A detailed description of the architecture of the *MMInteraktif* package designed within the model developed in this and previous chapters will be provided in the following chapter.

## **5.2 Learner-centred methodology and CELL**

Learner-centred language learning incorporates some elements of both humanistic methodology and cognitive learning theories. The involvement of the whole person in the language learning process is a critical element of humanism (Stevick, 1990), as discussed in Chapter 1. The hypothesis adopted from cognitive learning theories is that learners learn language through progressive modification of their interlanguages brought about through a process of hypothesis testing, confirmation/disconfirmation and subsequent modification (McLaughlin, 1987: 94 - 5; 145 - 7).

As part of this process, learners take an active role, cognitively interpreting new experience in terms of previous experience and models they have built up. This combination of the theories of both humanism and cognitivism, together with the research findings from strategy studies discussed in the previous chapter, lead to a new view of the interaction among these elements. This can be stated as follows: given the opportunities to use language and learning strategies in the second language, and some explanation and education in their appropriate application, learners can then develop these strategies through exposure to, and experience in, the second language. In a

learner-centred approach, such development can take place through a series of steps in which teachers play a progressively diminishing role (Nakhoul, 1993) as the involvement and investment of the learner progressively increase. This process, sometimes known as ‘scaffolding’ (Bruner, 1983; Applebee & Langer, 1983; Palincsar, 1986; Donato, 1994), forms the basis of our understanding of the operation of the Zone of Proximal Development (ZPD) discussed in the previous chapter (section 4.8.3) and later in section 5.4.3.3.2. The process of scaffolding is also reflected in the integration of Vygotskian principles with cognitivism in *MMInteraktif*, shifting the conception of initial learner state from a static cognitivist model to the dynamic one of socioculturalism.

There are dangers, of course, in expecting students to take control of their own learning without adequate preparation and practice. Some of these include the learner resistance mentioned by Nakhoul and Candy in the previous chapter, as well as confusion, frustration, and a sense of failure caused by inadequate preparation. As Robinson so succinctly puts it:

Clearly, *being put in charge* and *being in control* are not synonymous. In CALL, *student control* of the process is often akin to being put in charge of one’s learning without having developed the competencies to be in control (i.e., to succeed at the language task). Such premature and unguided *student control* should be avoided until the learner has developed the language competencies to succeed.

(Robinson, 1991:158)

Learners can best develop the necessary skills and processes to be able to take competent control of their own learning in a learner-centred language learning environment – where all activities are inter-related in having as their main aim the

development of informed learner autonomy. In such an environment, through constant practice, learners become more autonomous and self-directing in their attitudes and approaches to their own learning, enabling teachers to devote their time and attention to further enhancement of the materials available to learners as resources, and to counselling and advising students when the need arises. This, then, is the role envisaged for *MMInteraktif*, the software package described in this chapter.

### **5.2.1 Approaches to CALL design**

In the development of a learner-centred language learning environment, the framework is formed from the methodological principles as discussed above, while the medium is the resources. These resources include people such as teachers, other staff and friends, print resources, and non-print resources such as cassette, video (both linear and digital), laserdisc, CD-ROMs and computer equipment. Various writers have provided lists and categories of the characteristics of computers (Wyatt, 1988: see Figure 5.1 below) and the advantages of computers over other media (Weible, 1988), all of which include the individualisation of instruction. As discussed in earlier chapters, individualisation is a central element of humanism.

Weible, for example, mentions three unique qualities of CALL as an instructional medium: its facility for providing structured interaction; its process-oriented nature; and the possibility for providing automated individualised instruction (1988: 74). From a matrix of possible CALL applications, Pusack and Otto (1984) distilled four categories of approach: (1) practice and diagnosis, (2) tutorial, (3) simulation and problem-solving, and (4) utility, where programs of the last of these categories ‘place a very high priority on the ability of students to make decisions in using the computer for language learning’

(Pusack, 1987: 16). It is the development of the necessary decision-making, or metacognitive, strategies and skills, as discussed in the previous chapter, that forms the basis of automated individualised instruction using a learner-centred approach.

Figure 5.1 **Relational Classification of CALL Approaches**

(Wyatt, 1988: 89 - 90)

Approach	Characteristics
A. INSTRUCTIONAL e.g. tutorial, drill and practice, holistic practice, many “games”	<ul style="list-style-type: none"> <li>• Students are responders, not initiators, despite their high level of activity</li> <li>• Detailed set of high- and low-level learning objectives</li> <li>• Predetermined learning path(s)</li> <li>• The computer instructs the student; students learn <i>from</i> the computer</li> </ul>
B. COLLABORATIVE e.g. modeling, discovery, simulation, adventure reading, annotation, some “games”	<ul style="list-style-type: none"> <li>• Students are initiators, take more responsibility for their learning</li> <li>• May only be possible to specify learning objectives in high-level terms</li> <li>• No predetermined learning paths</li> <li>• Elements of discovery learning; students learn with the computer</li> </ul>
C. FACILITATIVE	<ul style="list-style-type: none"> <li>• Students are initiators, entirely responsible for their learning</li> <li>• Learning objectives and paths not specified or embodied in computer program</li> <li>• Students use computer as <i>tool</i> to reduce “inauthentic labor” [time-consuming or repetitive work not related to the activity of the current task]</li> </ul>

Authors in the area of CALL, however, have differed in their opinions of the extent to which computers can encourage collaborative learning and provide flexible learning paths (Weible, 1988). In this latter concern, Weible is echoing the view of Higgins and Johns (1984) as expressed in the introduction to this chapter: that software packages are often designed by people other than teachers, and certainly not by learners. As technology advances further, however, fewer of these differences will be voiced, both because technology is already outstripping its critics by undergoing rapid transformations in response to criticisms, and because more teachers are designing their own software tailored to their learners' needs. The level of learner involvement in, for example, the form of interactivity that was not possible in 1988, is now fairly freely available in the form of interactive digitised audio (*IDA* – Lian, 1985, 1987) and interactive video. Indeed, Lian's list of five categories for the role of computers in language learning – as a teacher and/or manager of learning, as a resource, as a tool, as an instrument for communication, and as a manager of users (in the sense of a local computer, rather than the more traditional sense of programmed learning) – provides an early insight into some of the possibilities that have now been made technically possible.

### **5.2.2 The interaction between CELL and the learning environment**

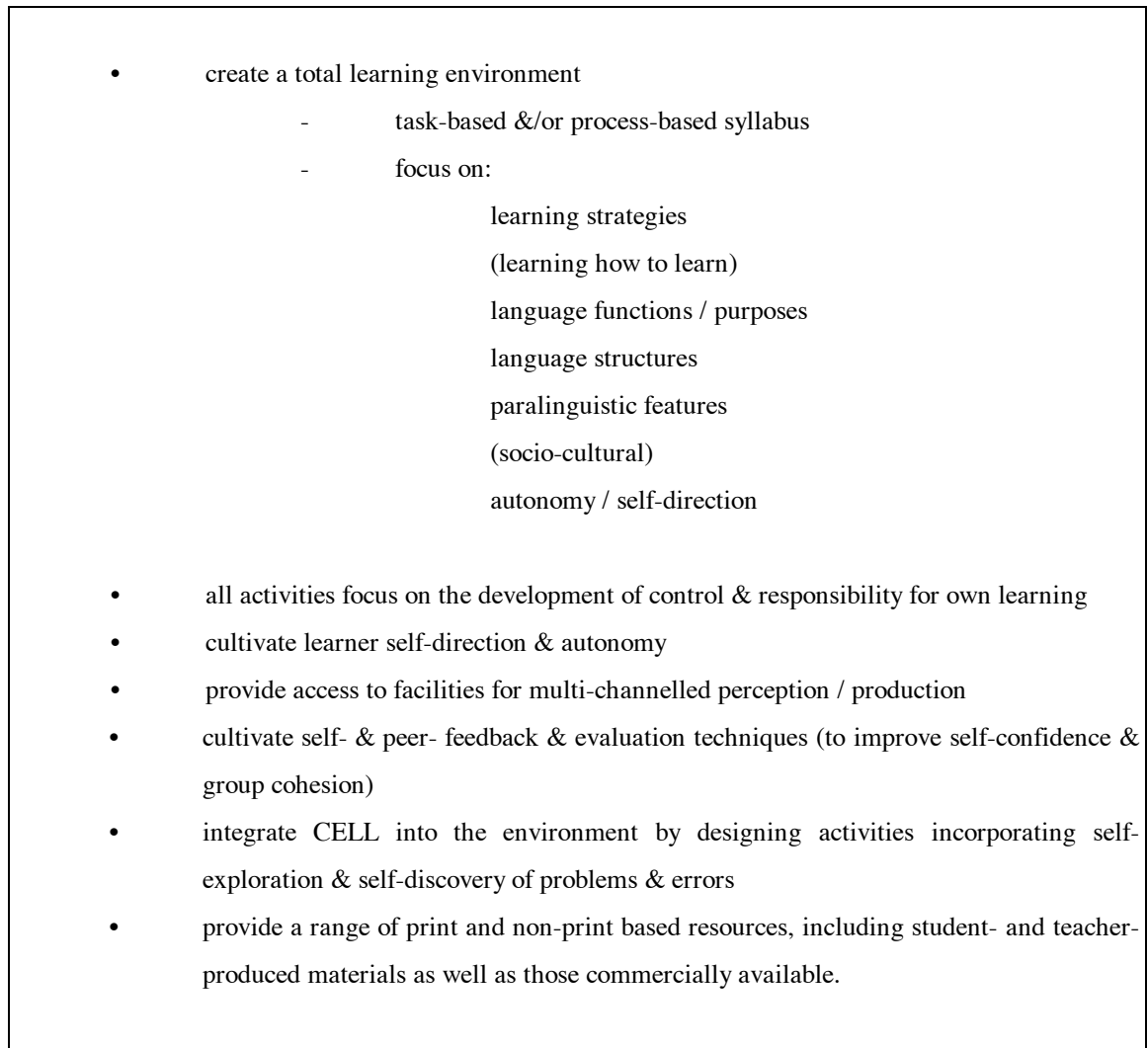
The guiding principle in the integration of technology into language learning should be usability within a framework of sound pedagogy. As mentioned in previous chapters, decisions to include *any* teaching resources or tools, including technology, need to be based upon principles derived from our best understandings of the nature of the teaching and learning process. As Pusack reminds us: 'Good CALL software has one essential property: it incorporates our best knowledge about how today's learners learn languages' (1987: 38). Increasingly, computers are being seen as tools in the language



teaching/learning process – tools for teachers to use in their teaching, and equally, tools for learners to use when they have something specific they wish to work on by themselves, at their own rate. This brings us back to the questions of how technology can be integrated into a language learning programme, and how courseware should be designed in order to best suit the place technology has in the programme. The various forms of computerised technology should be seen as resources in the learning process, in much the same way as books, but with the added interactive or cooperative learning dimension (see Figure 5.2 below).

As mentioned earlier, the instructional design of software can be technology-driven, teacher-driven, or learner-driven. The development of an integrated CELL learning environment also requires consideration of these three participants in the learning process. As shown in the findings of the survey discussed in section 1.6.3 of Chapter 1, teachers need training, not only in the design and use of CELL materials, but also in appropriate use of technology within a learner-centred framework. As outlined in the previous chapter, for learners, awareness-raising and practice in the use of learning strategies need to be provided to develop the skills and processes applicable to on-going learning, culminating in learners being able to take control of their own learning. Finally, the role of the technology should be as a resource in an existing methodological framework. As such, it is necessary that the development of the same skills, processes and strategies as in other parts of the learning environment be incorporated into software and courseware, while exploiting the technology in the most appropriate way.

Figure 5.2 A model of a CELL learning environment



### 5.2.3 Learning strategies in CELL

With regard to the incorporation of awareness-raising in the use of learning strategies, Weible (1987: 76) warns of the dangers in programmatic CALL of inculcating in learners the use of strategies which are less useful or desirable for on-going learning, such as in drilling activities. In order for successful learning to occur as part of the acculturation into classroom or learning communities it is common, and usually necessary, for learners to become familiar with the methodology realised through the teaching approach or learning materials to which they are exposed. In the sociocultural approach to language

learning, this phenomenon is recognised as having positive and productive potential, as mentioned in the previous chapter. Thus, in a well-designed CELL package based on this approach, learners should be able to become skilled in applying useful and productive learning strategies which foster and enable the development of higher level thinking processes, as well as the acquisition of new features of the target language.

Unfortunately, most software packages so far seem to have focussed on the ‘aided’ or ‘assisted’ aspects of CALL, with insufficient emphasis on the *enhancement* of learning as represented by the acronym CELL. The following sections of this chapter will examine this distinction through a review of some of the most prominent software packages for language learning and an analysis of the instructional design principles on which they are based.

### **5.3 Some approaches to instructional design: CELL as humanised CALL**

As discussed in Chapter 1, and revisited in Chapters 3 and 4, humanism in language learning requires the consideration of the whole learner, not just learners’ linguistic needs, or teachers’ perceptions of what these might be. In order for a humanistic CELL package to enhance learning, therefore, it is necessary, in the design of such a package, to consider learners’ learning styles and strategies. Included in these must also be a consideration of the possibility that learners may sometimes desire to be teacher- or software-directed, rather than self-directed. This aspect of self-directed learning has been identified in numerous studies (Candy, 1987; Nakhoul, 1993) and is emerging as a critical aspect of feedback from students using *MMInteraktif*. Learner control of navigation through the package, and of decisions on paths through the materials, is an essential part of this consideration.

In his discussion of the nature of CALL software, Lian (1991) introduces the question of control as being critical to the development of CALL. He maintains that learners ‘can, and indeed, must be given control over the ways in which they can select and interact with learning materials of all kinds, including computer-based materials’ (1991: 3). He identifies three conditions essential for learners to be able to gain control over their environment: the development of learner autonomy, the provision of numerous resources, and the provision for learners of the ‘means for controlling their learning’ (Lian, 1991: 3). As proposed in the previous chapter, the issue of learner autonomy is linked to raising learners’ awareness of the importance of taking control over one’s own learning, as well as an understanding of the strategies that can be employed to do this.

### **5.3.1 Aspects of the nature of computer-learner interactions**

From a programming rather than pedagogic perspective, Weyer (1988: 91), stresses the importance of taking learner intentions and needs into consideration in order to take computer-based learning from ‘Information in Isolation’ to something that more closely resembles the ‘Real World’. To this end, he advocates the creation of ‘a taxonomy of learner needs, styles, and problem-solving approaches’ (1988: 95-6), as being necessary to the development of systems that ‘help the learner use information more intelligently’. In this way, such systems can become more user-oriented than information-oriented. Some parallels can be drawn here with the sociocultural perspective, as Weyer attempts to conceptualise the impact of shared experiences and the personalisation of information on the restructuring of meaning within hypertext environments. Essentially he is interpreting the construction of meaning in a sociocultural manner in recognising the mediation taking place between users and the ‘hypertext’ environment. A detailed

explanation of what is meant by this term, and other related ones such as ‘hypermedia’ and ‘interactivity’, has been provided in Chapter 1, section 1.3.2.

In a theoretical examination of the interpersonal meanings realised through human-computer interactions, van Leeuwen and Shaner (1990) introduce the issue of power relationships, maintaining that ‘language in interaction can never be interpersonally neutral’. While van Leeuwen and Shaner use Halliday’s (1985) four basic speech functions – demanding information, demanding goods and services, offering information, and offering goods and services – to analyse and classify the textually-carried messages of computers and users, in doing so they have also produced some interesting conclusions for socioculturalists. Starting from the observation that ‘wherever there is language, there is not just a more or less efficient exchange of information, there is also an exchange of interpersonal meanings – interpersonal meanings which help construct some kind of social relationship between the participants of the interaction, and some kind of context [...]’ (van Leeuwen & Shaner, 1990) – they proceed to point out the contradictions and inconsistencies of register used by computers with their human interactors.

Thus, for example, at one point in an interactive session the computer uses formal written language to make a request of the learner (‘Please select the drives in which the files are found’), while at another point an informal register, more appropriate to spoken language, is used to give encouragement (‘Good Helen! Add to your response now ...’), and even formal apologies are found (‘Sorry I do not have that information’). Van Leeuwen and Shaner use this data to predict the emergence of a new, and more dialogic, written register to characterise the language exchanged in human-computer interactions.

They also suggest that it is the emergence of this language-based confusion over power relationships (master-slave, client-server, or some other metaphor), that has hastened the move from command language interfaces to more graphical ones. Other reasons for this move will be discussed in section 5.4.3.5 on interface features. Another result of this confusion is reflected in the other major trend in interface design – speech recognition and voice activation, and their incorporation into virtual reality environments.

### **5.3.2 A psycholinguistic approach**

In her discussion of the role of CALL in communicative language learning, Garrett (1987) advocates the use of computers to assist in awareness-raising among learners of the psycholinguistic processing aspects of grammar in language learning. Garrett bases her argument on several premises. The first of these is the shift away from language teaching, focussed on the transmission of a body of knowledge, to language learning, or learning to speak the language, which recognises language learning as the acquisition of a complex skill. As a corollary to this recognition, teachers can no longer expect to be able to teach a body of knowledge, but rather ‘assist the learning by providing learners with structured opportunities for practice’ (Garrett, 1987: 172). While such structure is necessary and useful, as will be illustrated in the details of the progress of flow of control through *MMInteraktif*, it is possible to allocate to learners more control over the nature of this structure, and the order of presentation. In this way, more of the burden of choice in direction is placed upon individual learners, at the same time as making provision for different learning styles. Faced with a presentation manner that is not in keeping with their learning styles, some learners may, for example, have difficulty in choosing a path through the materials which provides optimal learning.

Garrett's second premise is that computers can provide individualised instruction through analysis of learner responses and tailored or immediate feedback on these, in a manner that is not humanly possible for a teacher with a class full of learners. The third premise is that consciousness-raising of both processing itself and of major grammatical concepts can be achieved through computer tutorials, for example by exposing learners to a range of literal translations of the same content from other languages. Garrett's final premise is that information gained from learner practice on major grammatical topics and specific grammar points can contribute to our understanding of second language processes and foreign language education, as well as answering more basic questions on the effectiveness of certain kinds of CALL activities for certain kinds of learners, including all their individual differences. This, in fact, is an important area of investigation for the field of CALL research, as many studies have been undertaken on the effectiveness of CALL activities of various different types, without sufficient examination of the effects of the actual type of activity, or the individual differences or preferences of participating learners, independent of the technology-based environment (Chapelle, 1994). This issue will be raised again in the final chapter.

Given the premises above, Garrett claims that a psycholinguistic approach to CALL design and implementation such as she proposes can subsequently lead to the production of CALL packages that are better informed psycholinguistically, and help fill the gaps in our knowledge of the connections between the teaching of grammar, and learners being able to use the language grammatically. However, while this point is well taken, Garrett's focus on *grammatical* processing may be restrictive, since it does not accommodate other aspects of processing involved in language, such as the semantic and non-verbal.

Nevertheless, Garrett's approach does recognise the major role of the learner and learners' mental processing in mediated second language learning.

### **5.3.3 An interlanguage approach**

From a rather different perspective, Doughty (1991) advocates the use of interactive software accompanying videodiscs to promote the negotiation of meaning from a second language acquisition (interlanguage) perspective. In common with Garrett, she recognises the advantages of computers for the collection and analysis of data on learners' interactions with software packages for improving our understanding of how learners learn a second language. Where Garrett focuses on grammar, however, Doughty examines learners' negotiation of meaning from the perspective of comprehension using two different models of second language acquisition: the Negotiated Interaction Model (Hatch, 1978; Long, 1985; Pica and Doughty, 1985; Doughty and Pica, 1986), and the Cognitive-Processing Model (Rumelhart, 1977; McLaughlin, 1987; Carrell, 1988; Bransford *et al.*, 1989; O'Malley and Chamot, 1990).

These two models, constructed as they are to explain different aspects of second language acquisition, can be regarded as being complementary, rather than competing with each other. Similarly, the sociocultural model elaborated in the previous chapter is also not incompatible with these two models, though it interprets the same phenomena from another perspective again. While the Cognitive-Processing Model deals primarily with the perception, interpretation, storage and retrieval processes involved in language learning, incorporating Schema Theory (Carrell, 1988; Bransford *et al.*, 1989), the Negotiated Interaction Model focuses on the impact of the interactions of interlocutors on learners' language production, and therefore internalised interlanguage systems.



However, as discussed in the previous chapter, both these models embrace mechanistic metaphors of input and output, and for the purposes of this work, therefore, lack explanatory power in the area of shared construction of meaning from a sociocultural perspective.

According to Doughty, the Cognitive-Processing Model, for example, is useful in explaining how learners internalise formal structures, and the Negotiated Interaction Model is useful for the information it provides on the role of interaction in learners' language production. Nevertheless, Doughty recognises that learners need more guidance than these models provide, and that such guidance should take the form of awareness-raising of the range of strategies that can be employed, and how these can help them utilise more context in more creative ways (Doughty, 1991: 10). Awareness-raising thus comprises one aspect of the enhancement of learning embodied in a CELL environment. In an interactive videodisc environment, Doughty also designates possible roles for the three major features of negotiated interaction: requesting clarification and confirming understanding as being controlled by the learner, and checking for comprehension as being better placed in the control of the computer (Doughty, 1991: 12).

In this regard, Doughty urges software designers to 'balance contextualization with the need to push learners to understand difficult target language material that is not readily comprehensible' (1991: 7), to counteract further instances of the findings of Mohan (1987) that less interaction among learners occurred with computer-mediated tasks than with tasks not incorporating the use of technology. She goes on to suggest that 'some routine negotiation moves (several varieties of each) could be [...] created using the authoring system and a variable key element, and such resources should be available to

the learner via function keys' (1991: 8). One way of implementing these suggestions will be illustrated in the *MMInteraktif* software package described later in this chapter.

#### **5.3.4 Towards the incorporation of humanistic elements**

In response to what he perceives as criticisms levelled at CALL as being based on behaviourist principles, Stevens (1992) defines humanism in CALL as meaning that 'courseware lends itself [...] more to what students want it to be than what a particular program designer may have originally intended it to be' (1992: 11). While his definition is acceptable as far as it goes, given our current understanding of learning strategies and individual differences as overviewed in the last two chapters, it is now possible to design more humanistic features into software packages, changing the role of software from one of *assisting* to that of *enhancing*. Thus, rather than guessing what students might want a piece of software to do, it is possible to design CELL software that responds to what students identify as their needs, and to provide students with a framework around which to build their learning paths.

Stevens further elaborates on the role of humanism in CALL to encompass three positive contributions: the provision of environments conducive to learning, the move from teaching-centredness to learning- or learner-centredness, and a reemphasis on the worth of individual learners, and learners as individuals. He sees the obvious product of these influences as being the focus, in modern software design, on the development and fostering of learner autonomy as exemplified in exploratory CALL such as multimedia applications, and problem-solving software such as adventure games. As an earlier study has shown (Stevens, 1984), 'learning is enhanced when choice and control are in the hands of the learner' (Stevens, 1992: 23). The CELL software described in this chapter

demonstrates modes whereby enhancement is achieved by putting both this choice and the control in the hands of the learner, and simultaneously helping inform the learner about the rationale behind the choices made.

### **5.3.5 Towards the incorporation of cognitive elements**

In his conception of the role of computers in language learning, Mohan (1992) provides three models: the computer as language teacher (including as a resource), the computer as a stimulus for talk (Mohan's least preferred model - where the focus is on learner-learner interaction, and the computer is mostly peripheral), and the computer as context for cognitive language development. It is the last of these models to which the software in this chapter most closely corresponds. Mohan draws this model from the work of Cummins (1984) on bilingualism and its effects on, and relation to, general cognitive and academic development in English. The two critical features of Cummins' bipartite model of language proficiency in carrying out communicative tasks are the level of cognitive demand, and the level of contextual support. In this model, the most difficult tasks for second language learners are those which are both highly cognitively-demanding, and context-reduced, or disembedded, to refer back to Skehan's term discussed in the last two chapters. Because of the possibilities they make available to the learners for making choices in response to questions and problems, Mohan is convinced that computers are better in the role of providing context for cognitive language development, than as a stimulator of conversational fluency.

It is this model that Mohan sees as being the most productive by making a 'series of cognitive tasks available to the user'. However, Mohan qualifies this statement by adding that 'how demanding and difficult these tasks are will depend on the match between the

program and the user, among other things' (1992: 123). Mohan does not elaborate further on how this might be achieved or realised. In *MMInteraktif*, the taxonomy of listening and viewing tasks detailed in Chapter 2 has been constructed to illustrate one practical method of grading tasks for cognitive demand. By using this taxonomy as the organising structure for access to comprehension tasks, learners can make informed decisions on whether to pursue more tasks of a similar kind, but with different content material; whether to try other tasks at the same level of cognitive demand, but of a different kind and with different content material; or whether to go on to another range of tasks at a higher or lower level of cognitive demand.

The issue of the match between the program and the user is resolved by incorporating into the design principles of awareness-raising in learning strategies, particularly those strategies shown in the literature to be important for language learning management and the development of learner autonomy. In keeping with the issues raised by Candy (1987) and Nakhoul (1993) as discussed in the previous chapter, provision is also made for learners to follow lessons in connected sequences if they prefer a more teacher-directed mode of instruction or path through the materials. These lesson sequences are presented in the order they would be by a classroom teacher, following principles of communicative methodology and communicative task design (Nunan, 1989; Nunan, 1993; Pica *et al.* 1993). The implementation of these aspects of the software design, and the mechanisms for allowing the learner to maintain control, will be illustrated in the next chapter.

Taking as their starting point the principle that 'listening comprehension, in and of itself, is a necessary skill in which to attain proficiency when learning a foreign language', Bright *et al.* (1991: 18) proceed to describe and implement an instructional design model

for an interactive videodisc package based on current understanding of second language acquisition (Pederson, 1987) and what they see as Weible's (1987) "'plea" for creativity' (Bright *et al.*, 1991: 17). Bright *et al.* use three basic principles distilled from the instructional design theory of Gagné and Briggs (1979) and the instructional transaction theory of Merrill *et al.* (1989) to design Spanish language lesson sequences for repurposed laserdiscs. These principles are that instruction should support the internal processes of the learner, that sets of knowledge and skills should be integrated to reflect the holistic nature of authentic language, and that communication is inherently interactive. On the basis of these principles, they have produced lesson sequences in four stages: the Overview, Viewing the Video, a Mastery Check, and the final phase, Linkage to the Classroom.

In their Overview phase, learners are presented with a sequence of menus comprising Objective, Preview, and Cues, which are designed to focus and inform learners, and help them activate prior contextual knowledge. In the Viewing phase, learners can view the video segment, though they can only access other video controls through a menu bar activated by stopping the video. In the Mastery Check phase, learners must achieve at least 80% on a series of multiple choice, true/false, and cloze questions to proceed to the next phase; otherwise they must repeat the previous viewing phase. The final phase, Linkage to Classroom, is actually carried out in the classroom where the teacher uses 'a number of established question-and-answer techniques in order to ascertain that the students are thoroughly familiar with the characters involved in the segment and with the general story line' (Bright *et al.*, 1991: 22). In spite of the designers' best efforts, it is not clear from the description of the instructional design provided how this design follows

the principles it claims to espouse, much less modern approaches to teaching and learning as described in this and earlier chapters.

While the first phase, the Overview, does seem to fulfil the stated purposes of setting the scene for learners by helping them activate prior knowledge, and this aim is well-grounded in Schema Theory, the subsequent phases seem to lose direction. The restrictions put on learners at the Viewing phase, for example, though in keeping with the 4th and 5th events of Gagné and Briggs – ‘Present the stimulus material’, and ‘Provide learning guidance’ (Bright *et al.*, 1991: 19) – are overly restrictive for an interactive computer-enhanced environment, especially one that is expected to assist learners to develop communicative competence. Similarly, the Mastery Check phase still seems embedded in the older ‘drill-and-kill’ pedagogic approach grounded in behaviourist psychology, that other authors in the area (Weible, 1987; Garrett, 1987) have warned us does little to enhance learning except of the most basic rote kind. Finally, the activities described as forming the linkage to the classroom in the final phase seem to follow similarly behaviourist audio-lingual techniques, with no mention of the learner’s role, and no definition of either the kinds of outcomes anticipated, or the processes being developed.

### **5.3.6 Towards the incorporation of learner elements**

In contrast, in her approach to instructional design Meskill (1991) seems much more attuned to modern language pedagogy, and has produced an informative synthesis of computer-based instructional design and early learner-centred pedagogy. In particular,

her Phase I: The Front End, incorporates an investigation of the nature of the medium (interactive videodisc) and the needs and constraints of both learner and institution, and a formulation of institutional and instructional goals. When she comes to the second phase, The Instruction, however, though she does recognise the importance of specifying learning processes and outcomes, Meskill restricts herself to the Functional-Notional approach to language learning and, on the basis of this, specifies inventories of target language outcomes (following Munby, 1978), enabling strategies, and linguistic content, all of which are ‘chunked and organized into a hierarchical series of learning events’ (Meskill, 1991: 49).

In her third phase, Instructional Strategies, Meskill includes three different ‘human-system configurations: single user, pair work and group work’ (Meskill, 1991: 54), and this is a major strength of her design approach. These combinations of learners and machines are specified, together with rationales for such combinations, purposes for them, and the nature of expected interactions. However, even this aspect of her instructional design is strongly teacher-centred and restrictive, in that learners can only explore or experiment within the parameters she has set, based on conditions and effects. Meskill has subsequently investigated more extensively (1992) the kinds of oral interactions that occur among learners working with computers in various combinations, and concluded that in order for spontaneous and creative language to occur, software needs to be specifically designed to encourage and stimulate this. From these findings, it seems that for software to *enhance* learning, it is critical that it be used for the purposes for which it is designed. This again implies that the CELL software be designed for uses that are as flexible as possible, in order to minimise the danger of teachers and learners trying to put it to uses for which it is not designed. It is therefore essential that a clear

rationale and philosophy is provided with CELL software, so that teachers and learners using the software realise its roles and limitations.

### **5.3.7 Identifying features of CALL design approaches**

In his examination of design implications for an evaluation of CALL on the basis of various language teaching approaches, Hubbard (1987) lists three major categories of approach. While these categories are not meant to represent any particular theory or model of second language learning, some are identifiable (see Figure 5.3 below). The three categories are: the behaviourist, explicit learning, and acquisition approaches. For Hubbard, behaviourist approaches to language learning ‘are based on the principle that a response, linguistic or otherwise, is learner behavior resulting from associating that response with a given stimulus. Through positive reinforcement for correct behavior and negative reinforcement for incorrect behavior, these responses become overlearned until they are automatic’ (Hubbard, 1987: 231). Explicit learning approaches (Long, 1983) include the cognitive approaches of Ellis (1985b) and Bialystok (1978) discussed in Chapters 1 and 4. In Hubbard’s opinion, these derive from the earlier grammar translation methods, and information-processing approaches (McLaughlin et al., 1983), as mentioned earlier. For Hubbard, acquisition approaches (Krashen, 1981) include certain, though by no means all, communicatively oriented approaches, such as the Natural Approach (Terrell, 1977; Krashen & Terrell, 1983), and the Comprehension Approach (Winitz, 1981) discussed in chapters 1, 2, and 4.

With reference to CALL, several authors (Ariew, 1984; Baker, 1984; Dalgish, 1984; Underwood, 1984) feel that too much software has been designed along behaviourist principles, to the detriment of deeper and more creative approaches to design. As



Hubbard comments, ‘drill-and-practice software, especially the type whose exercises can be processed in a superficial way, is less likely to lead to the learning and retention of explicit rules and vocabulary than is software that requires a deeper level of cognitive processing’ (1987: 234). On the topic of explicit learning approaches, Hubbard includes the issue of learner control, citing Higgins (1983) as advocating student control in preference to computer controlled exercises. Studies by Stevens (1984) and Hubbard *et al.* (1986) have both found that student control enhances learning, though the latter study also emphasises the need for appropriate training in the use of the software for effective learning to take place. The features that Hubbard assigns to each of these approaches as they impinge on effective CALL software are found in Figure 5.3 below.

Hubbard also discusses learner strategy orientation in relation to the effectiveness of CALL software design, as a feature of *learning* rather than *teaching* approaches. Some of the strategy types Hubbard mentions as being ‘particularly well-suited to being introduced and practiced on the computer’ (1987: 239) are top-down strategies such as skimming, predicting, and inferring meaning; ‘production strategies such as writing dialogs, brainstorming, list making [...]’; and reader orientation strategies. In his discussion of this category, Hubbard actually lists many of the features that we have identified as distinguishing CELL from CALL software design. The *MMInteraktif* package described in this chapter, indeed, incorporates all of the top-down strategies mentioned here by Hubbard, as well as several others, and listener, rather than reader, orientation strategies in the Lesson Sequences. As the focus of the package is on listening and viewing, the tasks in, and structure of, the package are designed to develop perception, interaction and negotiation strategies, rather than those of production.

Figure 5.3 **Approaches to CALL Design** (tabulated from Hubbard, 1987: 230 - 237)

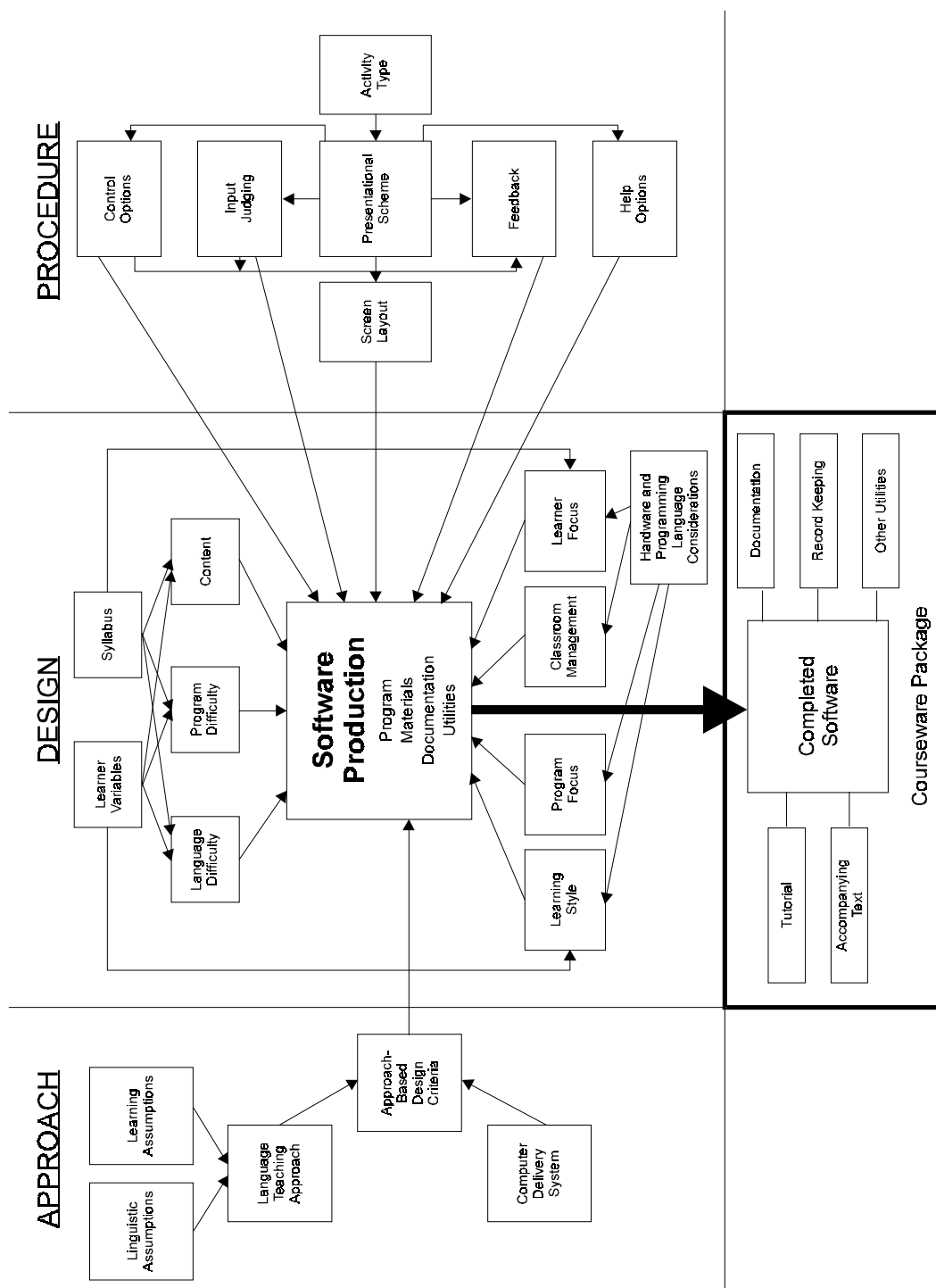
Behaviourist CALL	Explicit Learning CALL	Acquisition Oriented CALL	Learner Strategy Oriented CALL
<ol style="list-style-type: none"> <li>1. presents vocabulary and structure appropriate to learner's level</li> <li>2. maintains learner's attention to task</li> <li>3. does not accept errors as correct answer</li> <li>4. requires learner to input correct answer before proceeding</li> <li>5. provides learner with positive feedback for correct answers</li> <li>6. provides sufficient material for mastery and overlearning to occur</li> <li>7. reinforces patterns and vocabulary presented in a lesson</li> <li>8. presents grammar rules or patterns inductively with no attempt at teaching explicit formulations of them</li> </ol>	<ol style="list-style-type: none"> <li>1. introduces or reviews grammar rules and word meanings in an understandable, learnable, and reasonably accurate form</li> <li>2. provides effective practice so that (a) novel target language input can be readily understood, and (b) learner's understanding of rules leads to the production of grammatically acceptable spoken or written target-like discourse in novel situations</li> <li>3. gives meaningful rather than mechanical practice</li> <li>4. gives practice contextualized in a coherent discourse larger than a single sentence</li> <li>5. provides hints of various types to help lead students to acceptable answers</li> <li>6. accepts alternative correct answers within the given context</li> <li>7. Provides the student with explanation of correct answers</li> <li>8. anticipates incorrect or inappropriate answers and explains why such answers are incorrect or inappropriate</li> <li>9. maintains student's interest throughout the exercise</li> <li>10. allows an appropriate degree of student control</li> </ol>	<ol style="list-style-type: none"> <li>1. provides meaningful and communicative interaction between the learner and the computer</li> <li>2. provides comprehensible input at a level just beyond that currently acquired by the learner</li> <li>3. promotes a positive self-image in the learner</li> <li>4. motivates the learner to use the software</li> <li>5. motivates the learner to use the language</li> <li>6. provides a challenge but does not produce frustration or anxiety</li> <li>7. does not include overt error correction</li> <li>8. allows the learner the opportunity to produce comprehensible output</li> <li>9. acts effectively as a catalyst to promote learner-learner interaction in the target language</li> </ol>	<ol style="list-style-type: none"> <li>1. introduces the learner to strategies that are useful and immediately usable</li> <li>2. introduces the learner to strategies appropriate to the learner's level</li> <li>3. explains the value of the strategies</li> <li>4. provides meaningful practice in the use of the strategies</li> <li>5. presents practice material in such a way that the task is more easily or successfully accomplished if the appropriate strategy or strategies are used</li> <li>6. provides, when possible, a variety of strategies (or of techniques for utilizing a given strategy) for a given type of task suited to a range of learning styles</li> <li>7. provides feedback on which strategies might have worked best for given tasks after the learner has attempted them.</li> </ol>

As Hubbard (1987) comments, his classifications of CALL packages by orientation does not mean that all packages fit neatly into a particular category, but rather these classifications show some of the basic principles behind the instructional design of CALL courseware packages. The CELL package described in this chapter incorporates many of the features of both Explicit Learning CALL and Learner Strategy Oriented CALL, as well as some features of Acquisition Oriented CALL, and even one from Behaviourist CALL: providing positive feedback. The major distinguishing features of *MMInteraktif* as a CELL package are, however, the much greater emphasis on learner control over navigation through the materials, together with information for the learner on how this can be developed, and the use of authentic – rather than teacher-created or teacher-structured – materials as the content. This level of eclecticism is necessary to promote the learning of new, more helpful, strategies through the learning of new language. In other words, rather than just teaching about strategies, strategy development is implemented as an integral part of language learning.

#### 5.4 Towards a methodological framework for CELL instructional design

In a comprehensive examination of factors contributing to the construction of a methodological framework for CALL, Hubbard (1992) attempts to describe the complex inter-relationships between the classification schemes and lists of components of authors such as Phillips (1985) and Wyatt (1987), and the depth of analysis in media-specific methodologies such as that of Weible (1987). This methodological framework, which Hubbard conceives of as a network divided into three components – **Approach** (cf. Richards and Rodgers, 1982), **Design** (cf. Phillips, 1985), and **Procedure** (see Figure 5.4 below) – will be used to explicate the instructional design features of *MMInteraktif*, with some reference to other methodologists in specific areas of design.

Figure 5.4. Hubbard's Development module  
(Hubbard, 1992:46)



### 5.4.1 Approach

The **Approach** section of Hubbard's framework includes consideration of the following aspects which, with reference to this project, have been discussed elsewhere: **Linguistic Assumptions**, **Learning Assumptions**, **Language Teaching Approach**, and the **Computer Delivery System**, all leading into approach-based design criteria. Thus, for the design of *MMInteraktif*, the linguistic assumptions made are that the grammar of language is governed by rules and patterns, that language knowledge comprises a combination of structure and function, which is in turn mediated by interaction, and that all units of language from the morpheme to the text and context are fundamental to its use and acquisition (after Halliday, 1985).

Assumptions made about learning, as discussed in the previous chapter, are that learning occurs as a result of interaction and participation in learning communities, and that for language, this is achieved through a process of conscious and unconscious modelling, hypothesis-formation and confirmation/disconfirmation through interaction, and appropriation of recurring models. As discussed in detail in previous chapters, by adopting these models, and through interaction with interlocutors expert in the community, novices are able to internalise these models. This process is, in turn, reflected in the *learning* rather than *teaching* approach taken in the instructional design of *MMInteraktif*. Designed as a humanistic-cognitive, learner-centred package that fosters learner self-direction and ultimately autonomy, *MMInteraktif* is a task-based instructional medium, which is holistic in its focus on language. The package is thus meant to be a tool in the hands of learners to simultaneously explore and learn both language and language learning strategies.

With reference to the computer delivery system, Hubbard reiterates the point made earlier by Weible (1987), that instructional designers would do well to bear in mind the strengths of computers as an instruction delivery medium, reminding us, for example, that computers are not best used as conversation partners.

#### **5.4.2 Design**

In his **Design** section, Hubbard includes (following Phillips, 1985) **Learner Variables, Syllabus Orientation, Language Difficulty, Program Difficulty, Content, Learning Style, Program Focus, Classroom Management, Learner Focus, and Hardware and Programming Language** considerations.

##### **5.4.2.1 Syllabus orientation**

While Hubbard regards Syllabus Orientation and Learner Variables as the most important components, for the purposes of *MMInteraktif*, Syllabus Orientation is of only minor consideration, as the flexibility and comprehensiveness of the package allow it to be incorporated into a variety of different syllabus types. In addition, as this software package is designed to be used as a language resource, rather than as courseware, which is the focus of Hubbard's discussion, more flexibility is possible in design with regard to syllabus orientation. Thus, for example, while many different task types are included in *MMInteraktif*, ranging from discriminating among different discourse markers to reading meaning from non-verbal signals, no specific order of presentation is predetermined by the designer and imposed on learners. It is therefore at the Approach level, in this case, learner-centred and task-based, that syllabus considerations impinge on design, rather than at Hubbard's Design level.

#### 5.4.2.2 Learner variables, language difficulty, and content

Hubbard's **Learner Variables** component includes the individual differences discussed in detail in Chapter 3, and the learning strategies examined in Chapter 4 of this work, while the **Language Difficulty** component is dealt with here through the incorporation of learner self-selection on the basis of task types classified according to the taxonomy discussed in Chapter 2. As the **Content** component of this software package is authentic text taken from feature films or television, the only decisions the designer need make with regard to Language Difficulty or Content relate to ensuring the inclusion of a representative sample of texts in terms of register, genre, context, situation, interaction, and the gender balance of participants. The role of these decisions, a necessary part of instructional design for any medium, is represented in the diagram (Figure 5.6 in section 5.4.2.4) illustrating the process of learner-centred task design for authentic materials.

#### 5.4.2.3 Learning style

Hubbard's Learning Style component (1992: 55) is taken directly from Phillips' (1985) framework, which is in turn based on the classification scheme of Kemmis *et al.* (1977) for learning tasks rather than learner variables. Five general types of learning activities are proposed for CALL in this scheme: recognition, recall, comprehension, experiential learning, and constructive understanding. There is an obvious correspondence here with the taxonomy of tasks in terms of cognitive demand proposed in Chapter 2, and incorporated into *MMInteraktif* in the Taxonomy Layer: recognition and recall are clearly at the Knowledge level, comprehension at the level of the same name, and experiential learning at the Application and Analysis levels, while constructive understanding corresponds roughly to the level of Synthesis. The Evaluation Level of the

*MMInteraktif* Taxonomy, or the task types in this level of classification, however, do not appear in any of the other schemes or frameworks referred to.

#### **5.4.2.4 Program focus**

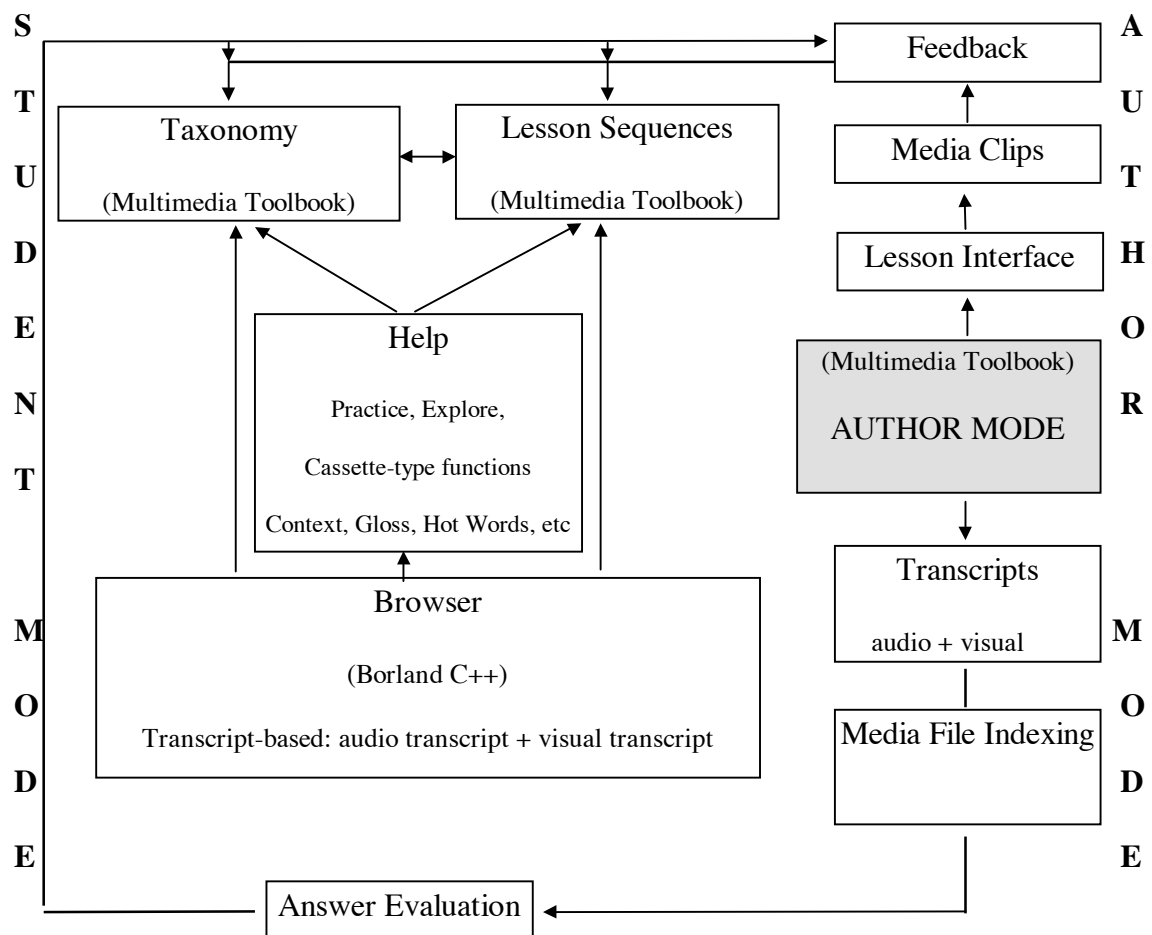
For *MMInteraktif*, the Program Focus, or linguistic objectives of the activities, are determined by a combination of the text type on which the activities are based, and the level on the listening and viewing comprehension taxonomy where the activities are located. Hubbard describes the categories within this component, such as discourse/text, lexis, morphology, as being in need of further elaboration by developers. In *MMInteraktif*, this elaboration is achieved by providing a three-layered approach to the listening and viewing materials: Browser, Taxonomy, and Lesson Sequences; and by allowing learners the freedom to choose which of these layers they wish to work within at any point (see Figure 5.5). In the Browser layer, the implementation of which will be described in greater detail later, learners can browse through the audiovisual materials in linear or non-linear fashion. Word gloss, phrase information, intonation practice, visual classification searches, and record and playback facilities are all available in this layer, as well as cassette controls for linear playback. Learners can, therefore, focus on any level of textual feature, from lexical or grapho-phonological to discourse and paralinguistic.

In the Taxonomy layer, learners can access tasks from the taxonomy, based on the audiovisual materials, including information on the levels of difficulty inherent in the taxonomy, and individual tasks. Help in this layer includes access to the transcript, grammar notes, feedback on both correct and incorrect answers, and access to the Browser layer and all its features. Within the Taxonomy Layer, learners have the freedom to choose which tasks they work on, at which level of the taxonomy, and in



whatever order. The Lesson Sequences layer, on the other hand, structures the available tasks for learners in an order similar to that in which they might be presented in class by a teacher. Thus, for example, a previewing task is presented first, followed by a task involving visual playback only, on which a predicting task is based. This is followed by a more in-depth comprehension task, and finally, a task with greater linguistic or paralinguistic focus.

Figure 5.5 **Organisational Structure of *MMInteraktif***



However, even within this structured layer, the learner-centred approach is realised through the continued facility for learners to go forward or backward as they wish, whether or not they have completed a lesson in the sequence. Each Lesson Sequence module begins with an overview screen outlining the number, order, and objective of each lesson in the sequence. Each lesson screen also informs learners of the number of lessons in that sequence, and the order of that particular lesson in the sequence.

With regard to design of the tasks themselves, Dickinson (1987) makes the point that self-instructional materials, like any other teaching and learning materials, should be interesting, varied and clear. According to Dickinson (1987: 80), specific design features for self-instructional materials include:

- a clear statement of objectives (in terms of learners' levels and needs);
- meaningful language input (comprehensible and with appropriate help and support e.g. transcripts, dictionaries etc.);
- exercise materials and activities (sufficient to achieve the stated objectives);
- flexibility of materials (to accommodate or resolve conflict between meeting learners' individual styles and following appropriate learning paths);
- learning instructions (intelligible and informative, including worked examples);
- language learning advice (information on learning how to learn and learning strategies);

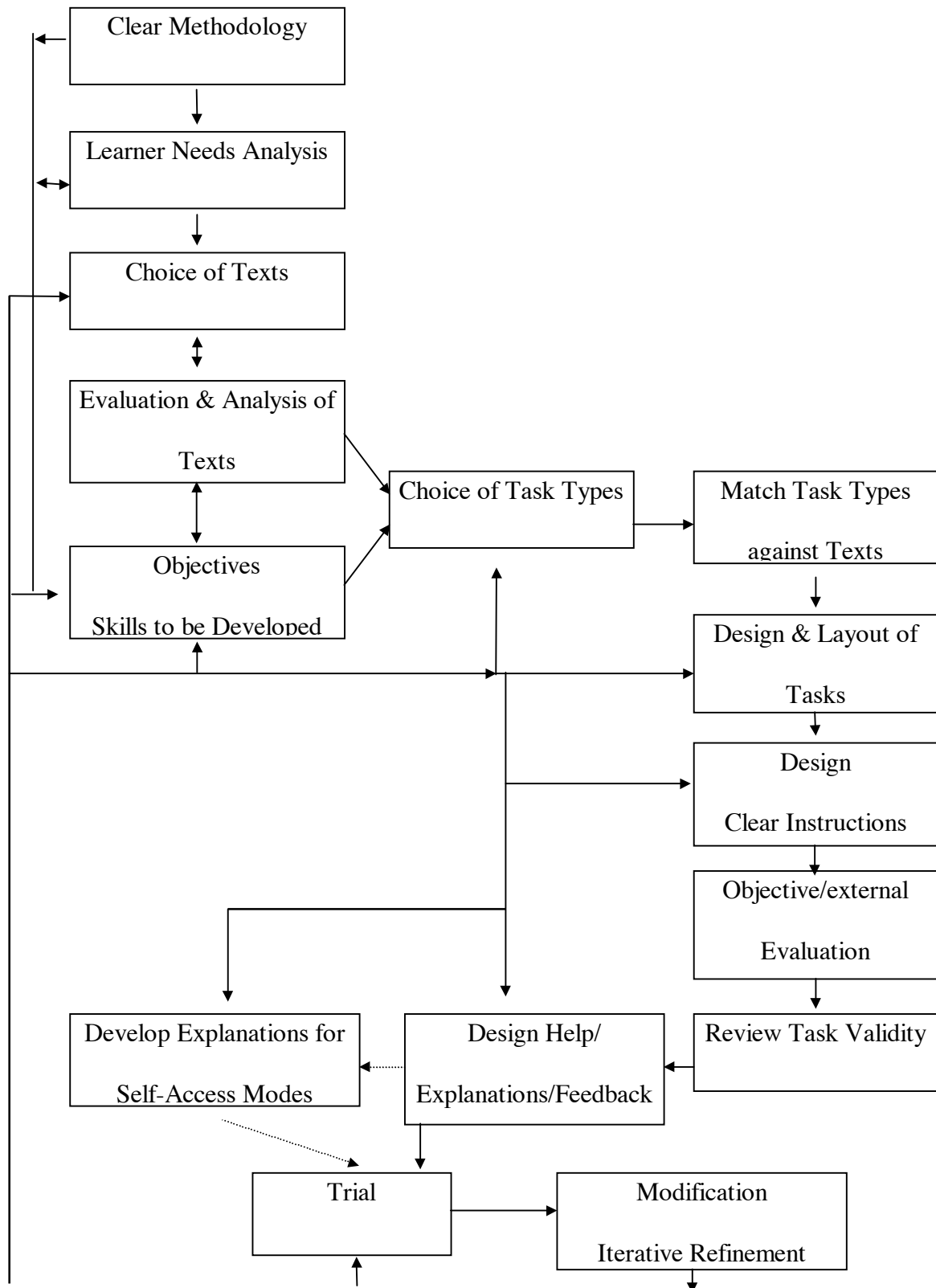
- feedback and tests (including feedback on how and where learners are wrong, and even how the correct answer is reached, and for tests, that they test what is taught);
- advice about record keeping (of what has been done and to what level - to check progress and as psychological aid);
- reference materials (e.g. grammar books, dictionaries, information on functions, notions and discourse, information on pronunciation);
- indexing (or contents list);
- motivational factors (attractiveness, professional appearance, accessibility, reasonably sized learning units, objectives explained, motivational messages based on affective and metacognitive strategies);
- advice about progression (where the learner has been, where to go next, and information on metacognitive strategies for planning learning).

These factors, having been found applicable in non-computer-based self instructional materials, also make eminent sense for self-instructional materials in a computer-enhanced environment, such as the software package being discussed here. In addition, and also on the topic of the learning task, it is necessary to consider exactly how the

tasks are designed, or what procedures are followed in the design and construction of tasks.

The diagram below (Figure 5.6) brings together earlier work by this author (Hoven, 1988), with the addition of some points made by both Hubbard and Dickinson. This whole process begins with the conceptualisation of clear methodology and hinges on the linkage between this methodology and learners' identified needs. Other points stressed in this process are the selection and matching of texts with tasks, and the necessity of iteration. Modifications at various stages of the process require checking through the design stage by stage to take account of the effects and implications of the modifications.

Figure 5.6 **The process of learner-centred task design based on authentic materials**  
(adapted from Hoven, 1988)



#### **5.4.2.5 Classroom management, program difficulty, hardware and programming language**

Hubbard's **Classroom Management** component, as based on Phillips' system, refers to the grouping of students using computers. In the case of the current version of *MMInteraktif*, as discussed earlier, learner participation is focused on the single user in a self-access situation. However, the nature of the instructional design and the interface does allow provision for continuing research and development in the direction of collaborative, network-based learning contexts. The **Program Difficulty** component and **Hardware and Programming Language** considerations will be dealt with in detail in a later section on *MMInteraktif* interface design.

#### **5.4.3 Procedure**

Hubbard's **Procedure** section (1992: 58) includes the following components of programming decisions: **Activity Type**, **Presentation Scheme**, **Input Judging**, **Feedback**, **Control Options**, and **Screen Layout**. As Hubbard himself comments (1992: 59) and indicates in Figure 5.4 earlier, the central component of the Procedure section is the Presentation Scheme, as it incorporates all the other components except activity type. The navigational features of CELL instructional software design covered in this section are intrinsic to the concept of a learner-centred model for CELL. These features are critical, not merely at the level of providing the means whereby learners can make choices about their learning paths, but, more importantly, because of the provision of opportunities for learners to use learning styles with which they feel comfortable. Ease of navigation is also essential for learners to be able to see, and make use of, the full range of options available at a number of levels, including media and task complexity as discussed earlier in Chapters 3 and 4.

#### 5.4.3.1 Activity type

Phillips (1985) recognised that in CALL in the past, the choices in Activity Type were few, namely: game, quiz, text reconstruction, simulation, problem-solving, and exploratory activities. However, because of the speed with which computer technology and software has developed, these choices have expanded to include more flexible modes of input and networking, allowing greater interaction between learners and the computer, as well as between learners at the computer. Oxford (1994a) also comments on the importance of making available to learners in an ICALL environment (which is equally applicable in a CELL environment) a variety of interaction types, rather than just a Student-(Intelligent) Tutor. By this she means using the full range of computer-student and student-computer-student interaction combinations currently available to enhance learning in communicative ways. These include the activity types mentioned above by Phillips, as well as others such as information-gap activities, story-creation, and role-plays.

Nevertheless, in the context of the listening and viewing software package design being discussed here, activity types will be restricted to those involving only one learner and one computer at a time, as is typical in a self-access situation aimed at developing listening and viewing skills. Although other, very productive and beneficial, activities, as described by Phillips and Oxford, are possible in networked classrooms, and in paired and other groupings around a computer, *MMInteraktif* is designed for single-user, individual work. Also, in keeping with both the principles of language learning involving authentic materials described in Chapter 1, and the task types incorporated in the taxonomy, as well as the learning strategies discussed in the previous chapter, the range of games,

quizzes, and simulations will be limited by the choice of texts for inclusion as content in such a package. This author maintains that, even with this provision, a sufficient range of activity types can be made available to learners, incorporating a variety of stimulus and content material, and exploiting several different mediums, for the learning styles discussed in Chapter 3 to be catered for. The record-keeping facilities of *MMInteraktif*, discussed in detail in the next chapter, will also provide a rich source of data for continuing research into the effectiveness and ease of use of different activity types.

Although significant advances have been made in the areas of natural language parsing, neural networks, speech recognition, and intelligent tutoring systems (Bull *et al.* 1993; Nagata, 1993; Yuan, Kunst & Borchardt, 1994; Oxford, 1994a; Bull, 1995; Hagen, 1995), the facility to handle random learner input is still relatively limited (Bull & Smith, 1995; Nagata, 1995), and certainly still inadequate for a learner-centred, learner-controlled package. For this reason, this author decided to limit learner input to the computer to voice recording for the purposes of self-comparison and subsequent review by a teacher, some text input in the form of single word or short phrase, drag-and-drop activity, and choosing among given options. This present limitation, however, does not preclude the incorporation of the architecture of the package into more ambitious models at a later date, when software engineering technology has advanced further. The decision was also based on the principle that it is better to focus on good activity types which have been successfully used in self-access situations, and consider how these can be realised in a CELL environment. It was with this decision in mind that the taxonomy of listening and viewing comprehension tasks was developed.



Decisions about determining at which level a specific task should be placed are based on the level of cognitive processing needed to complete the task itself, considered separately from other demands that completing the task may put on the learner. The less production required of the learner to complete a task, the more cognitive processing capacity is free to devote to responding to the demands of the task itself. This is also related to the issue of ease of use or user-friendliness of the software. If software is difficult to use, negotiation through the software can also take away cognitive processing capacity from the actual task. England (1989: 36), for example, cites 'the difficulty of concentrating on content and operating procedures' as one of the reasons for non-use of available software in educational institutions, while Nickerson (1981) mentions the lack of a simple command language as a factor causing frustration among users.

In addition, Sweller (1988) has found in the fields of mathematics and science that conventional problem-solving activity is not effective in schemata acquisition. He claims that the overlap in cognitive processing between problem-solving and the acquisition of schema is so small that these two kinds of processing actually compete for available capacity. Because of this competition for processing capacity, when schema acquisition is attempted through the usual kinds of problem-solving activities, cognitive load is increased to the extent that learning is impeded in both areas. As a result of further investigations, Sweller *et al.* (1990) suggest that this interference of problem-solving activity can be overcome by providing learners with more worked examples in which disparate sources of information have been previously integrated. In a range of experiments to test this theory, Tarmizi and Sweller (1988) and Ward and Sweller (1990) find strong evidence that, in the examples provided, learners' attention needs to be directed appropriately for cognitive load to be reduced, and that 'worked examples were

effective only when attention to disparate sources of information was reduced' (Sweller *et al.*, 1990).

Ganster, Hurrell, and Thomas add that:

task environments that present impoverished information or which present information rapidly have a greater likelihood of imposing [cognitive] load than do tasks that present in a clear and reasonably paced manner. Tasks that require the worker to attend to such informational input make the perception and interpretation of information more difficult, and thus impose more cognitive load.

(Ganster, *et al.*, 1987: 233)

It is therefore important to find a balance between developing in learners the ability to process cognitively demanding tasks, while at the same time minimising the system-related operations required in the actual task demands, and still providing enough information for learners to complete the task successfully.

This principle forms the basis for the reasoning behind the activities used by Bloom *et al.* (1956) to test students' processing at various levels of his taxonomy. Bloom *et al.* were able to use multiple choice activity types for testing students at all levels on the taxonomy: by being able to successfully comprehend the question, process the distracters, and choose the correct answer from among these, students demonstrated that they were capable of cognitive processing at that level. The structure of the actual tasks, however, remained consistently of the multiple choice type, with certain clear, generic instructions consistent across all tasks.

#### **5.4.3.2 Presentational scheme**

The term ‘presentational scheme’ refers to the manner in which the various tasks and activities in the software package are realised or represented by the computer interface. As programming and authoring languages increase in power and complexity, a wider range of techniques is possible for presenting the existing activities, as well as for creating new ones. Thus, for example, where in the past a multiple choice activity could only be represented textually, with learners using arrow keys to select their choice of answer, it can now be represented graphically, using click-and-select or drag-and-drop techniques with a mouse. Alternatively, learners can play several movies and choose among them by means of a mouse click to indicate which one corresponds to a given audio statement. Hubbard regards as one of the most important decisions ‘what the communicative modality of both the computer’s output and the learner’s input will be’ (1992: 59). As in the example above, the computer’s output might be graphical, aural, audiovisual, or textual, while the learner’s input might take the form of anything from a touch on the screen to a mouse click or entering text using the keyboard.

As mentioned earlier in the section on instructional design approaches, there is still considerable confusion among instructional designers and developers about the nature of the relationship between computers and users (van Leeuwen & Shaner, 1990). The increasing number of programs of different kinds, ranging from intelligent tutor to factory robotics, has compounded rather than simplified this confusion in the last few years. However, it is still valid to conjecture, as do van Leeuwen & Shaner mentioned earlier, that the need to simplify the conceptualisation of this relationship has been an additional influencing factor in the progressive move from text-based to graphical interfaces.

Within the structure of *MMInteraktif* as illustrated in Figure 5.5 earlier, a number of design decisions relating to the presentational scheme needed to be made. The range of these is shown in Figure 5.6 below. Having considered these aspects of the design, it was then necessary to review the guidelines found in the literature for implementing these decisions.

Figure 5.7 **Design decisions to be made** (Source – original)

- nature of learner control (which all contribute towards nature and timing of presentation of help & feedback):
  - choice of level to work at?
  - choice of advancing regardless of answer?
  - choice of skipping certain steps?
- level of interactivity (sources of computer input e.g. CD-ROM, digital video or audio, laserdisc) vs learner input vs computer-displayed answer options;
- system help (which key/button/menu?) vs task help (display the question again) vs global help (dictionary/transcript/translation);
- how and when to indicate correct vs incorrect response given;
- what additional feedback, if any, is given for learner responses;
- how to handle errors in free input by learners (spelling/typing/inflectional/derivational etc. - Hubbard, 1992: 60);
- nature of record keeping (based on pedagogical approach & CALL orientation):
  - scores kept?
  - correct, incorrect, and retried answers recorded?
  - other keyboard and mouse activity recorded?
  - timing on tasks and number of repeated actions recorded?
  - diagnostic/remedial information given?
  - learner only vs teacher only vs accessible to both.

On the topic of presentational scheme, Gordon (1994: 112) makes the point that it is difficult to research the effect of variables such as comprehension and retention by learners using a CALL program until ease of use has been achieved, because ‘learners won’t like systems as well if they aren’t easy to use, and they will also show poorer learning to the extent that cognitive resources must be devoted to using the interface’. This, together with her observation that most CALL users are novices who spend only a short part of their learning using a particular CALL package, give rise to her first goal of interface design: ‘Create a system that is easy for the learner to use’ (Gordon, 1994: 106). For Gordon, ease of use means that ‘at any given point, users should understand what is being presented, what they are required to do or have the option of doing, and how to accomplish their current goal’ (Gordon, 1994: 106).

In her guidelines for incorporating interaction into instructional design, Gordon stresses the importance of ‘transparency’. By this she means ‘at least two things should be apparent to the user: the general nature of the program (what it is used for), and how to interact with the program, (1994: 108). The importance of transparency as a feature of user-oriented systems is also stressed by Ulich (1987: 33) along with consistency (across screens), tolerance (of user errors), compatibility, support, flexibility/user-definability, and participation (of the user in the design of the dialogue systems). In addition to these general principles, Gordon (1994: 108) lists the following specific guidelines:

- Present an overview at the beginning (how the system works and how to interact with it);
- Provide an advance organiser (e.g. a menu, table of contents, map etc.);

- Make use of learners' previously existing knowledge (e.g. of icons and system help);
- A graphical interface is generally better than command language;
- Be consistent in the use of design elements from one screen to the next;

and (for hypertext or hypermedia systems)

- Make the underlying organisational structure hierarchical.

This last point of Gordon's above refers to the need to structure with care the links between the various elements of hypermedia, such that a hierarchical structure is formed.

With regard to the use of an advance organiser and a graphical interface for novices, Williges (1987) found that visual augmentation to the user interface in the form of a graphical representation of the hierarchical structure of the system improved the performance of users with both high and low spatial ability. The addition of this graphical representation is one example of an attempt to cater for individual differences, and had actually been introduced to assist users with low spatial ability who were using a number of inefficient navigation strategies based on command language or text-based commands.

In the listening and viewing comprehension package described in this work, Gordon's guidelines above have been followed. An overview is given at the beginning of *MMInteraktif*, showing learners the structure of the package, and how the links are made between different layers. Each of the Lesson Sequence modules begins with a menu detailing each of the lessons in the module and showing their sequence of presentation. All icons for browser, lesson, and system navigation follow Gordon's guidelines in that

there is a textual description displayed as well as the graphical icon. While the interface for the Taxonomy and Lesson Sequence modules is predominantly graphical, the built-in menu system of the Microsoft Windows environment is also used in the Browser layer, to allow for the large number of features and operations that are available there. Consistency is maintained across the screens and across the Taxonomy and Lesson Sequence Layers, and wherever possible, conventions are used that are common to the Windows environment to enhance predictability, and other benefits of using a familiar, standardised interface.

#### **5.4.3.3 Input judging, feedback, and Help options**

Input Judging, Feedback, and Help Options are all inter-related in terms of the design decisions that have to be made, and their degree of ‘transparency’ and accessibility is related to how they are represented on the screen. In relation to the actual content of the help and feedback systems, Rüschoff comments:

When designing hypertext and multi-media resources it is extremely important to think carefully about what buttons we provide the learners with and what kind of help and information can be accessed through them. Above all, help must lead to strategy building and be cognition oriented rather than just geared towards task solving.

(Rüschoff, 1993: 12-13)

As mentioned above, in keeping with the learner-centred principles along which *MMInteraktif* is designed, even in the structured Lesson Sequences layer learners are not constrained by the system to a lock-step approach. Thus, while they are provided with information on the suggested order of the lessons, they are given the facility to proceed without working through a lesson, or to go back to a previous lesson, or, indeed, to exit

to another layer, or out of the package. *MMInteraktif* does, however, record all steps taken by the learner through the materials.

In *MMInteraktif*, help is provided in a number of ways, depending on the learner's level of self-direction, and need. Lesson-specific help is given as part of answer feedback, while on-screen icons at both the Taxonomy and Lesson Sequence Layers provide contextual help by allowing learners to play the utterances surrounding the one(s) on which the task is based, allowing learners access to grammar reference notes relating to a specific task, or providing more free-form help in the form of access to the Browser, complete with transcript, cassette controls, and a reference grammar. Media control panels are always available for learners to play the audio or visual media as they require, at any layer. The actual mechanisms through which this is implemented are illustrated in the next chapter.

The Browser is designed as the most comprehensive help system for those learners who are self-directed enough to make optimal use of it. By incorporating the Browser as a level of help in the other layers, the package enables less self-directed learners to become familiar with its features in a more structured manner. Structure is also provided for the incorporation of learning strategies by means of a graphical information screen illustrating the relationship of the various categories of strategies to each other, with links to information on each of the categories, and tasks which are representative of those categories.



#### 5.4.3.3.1 Input judging and answer evaluation

Historically, input judging has been approached from a variety of perspectives, as discussed in Chapter 1, from the ‘one-answer is correct’ approach, to various forms of semi-intelligent answer mark-up (e.g. DASHER: Pusack, 1983), to the more elaborate knowledge-based approach of expert systems and the natural language parsing (Nagata, 1995) and diagnostic feedback of intelligent tutors (Bull & Smith, 1995). Burston (1990) describes a compromise approach to input judging between the high (and therefore expensive) computer-processing power required for natural language parsing, and the pragmatic needs of teachers and learners in educational institutions with stringent budgets. He classifies semi-intelligent answer mark-up on a sliding scale, on the basis of the degree to which string matching is exact, and the degree to which teacher-authors can predict possible student wrong answers. Burston’s categories are listed in Figure 5.8 below.

Through teacher-author incorporation of feedback messages appropriate to each level of the mark-up routines, Burston claims a level of semi-intelligence for the flexible response handling as described. In this he reflects the concerns of Nys (1989) as discussed in Chapter 1 that, certainly with the current levels of ICALL development, the intelligence of CALL should reside with the uses to which humans put it, rather than with the computer systems. The Help and Feedback options of *MMInteraktif* attempt to do just this – to incorporate in a manner that is informative and easily accessible to learners the overviews, reference material, practice opportunities, as well as task-specific feedback that will enable them to optimise their learning.

Figure 5.8 Categories of Answer Mark-Up

(adapted from Burston, 1990)

- Non-Differential Variation (where students' exact keyboard inputs are matched character-by-character to an ideal template)
- Relaxed Pattern Matching (which eliminates typing, punctuation, and case errors, and can also include various 'wildcard' combinations to allow for other relatively trivial errors such as misspellings)
- Automatic Pattern Mark-up (as in programs such as PLATO – Hart, 1995; and DASHER – Pusack, 1983; which indicate visually by means of blanks or highlights which characters or sections of a student's input do not match the 'correct' template)
- Flexible Response Handling:
  - Right Enough/Wrong Enough (where the focus of the exercise determines to what level and in what order the top-down mark-up occurs - e.g. in cases where capitalisation and spacing are wrongly used, but not regarded as important, and an answer is otherwise correct, it is Right Enough)
  - Fuzzy Right/Fuzzy Wrong (a less stringent category than the previous one, where the full range of answer mark-up routines above is brought into action).

Peng (1993) advocates a similar progression to that of Burston, using a combination of answer mark-up and error-contingent feedback for Asian language CALL. She describes the CALIS (Computer Assisted Language Instructional System) software as providing this combination of answer mark-up routines, based on an 'edit distance' algorithm (Nesbit, 1990), and error contingent feedback, based on a language-specific error detection program. Burston would therefore classify the CALIS software as being 'semi-intelligent' as defined above.

Joy, Lian & Russell (1983), on the other hand, stress that, depending on the point at which a question is asked of learners, answers may be evaluated differently. Thus, for

example, at an early point all answers may be accepted with no evaluation, allowing learners to build on their knowledge and understandings through further interactions with the lessons, before being asked to re-evaluate their earlier response. This technique is an example of a practical implementation of the hypothesis-formation approach to interlanguage development discussed earlier. In a subsequent publication, Lian (1984) adds the possibility of allowing students to make multiple attempts to answer a question, with or without computer feedback messages, as well as the range of answer mark-up routines detailed by Burston above. Allowing these multiple attempts is also consistent with the hypothesis-testing approach, in that, on the basis of the feedback they receive at each attempt, learners can refine their hypotheses about the structure and function of the language used. Another contribution of Lian (1984) to answer evaluation is in the acceptance of less-than-preferred responses. In such instances, a student's answer is accepted, together with the display of a message informing the student of the expected response.

In the Lesson Sequence modules of *MMInteraktif*, certain elements of these answer evaluation techniques are incorporated. However, in the Taxonomy Layer, where the focus is on learner decisions on order and preference of task presentation, many of these techniques, which are based on teaching sequences rather than on individual learning sequences, are inappropriate and therefore do not appear. Thus, for example, if learners choose to do a language-intensive task such as discrimination of discourse markers before a prediction task, then a teacher's intuitions of expected wrong answers would not be as well-informed or accurate as if learners had chosen the tasks in an order more in keeping with research findings on appropriate teaching sequences for tasks such as these. In this case, the learner would be focusing on the development of different, higher-level

cognitive skills. This is another reason, to add to those already given earlier, for the restriction of activity types to those requiring minimal free-form input from learners.

#### **5.4.3.3.2 Feedback and Help**

Feedback has been well researched in the field of computers and language learning: when it should be given, what kind, how often, whether explicit or implicit, and how detailed. Regarding when feedback should be given, Nagata (1993: 331-2) comments: ‘the issue is not simply whether or not immediate feedback brings about learning effects; the effects of immediate feedback depend on what instructional goal is posed, what skill is focused on to develop, what type of learning task is required, and what kind of error analysis and feedback is provided in the program’. As the example from Joy *et al.* (1983) above illustrates, there can be sound linguistic and pedagogical reasons for not offering any answer evaluation or feedback at all at certain stages in the learning process. In addition, as exemplified in some of the *MMInteraktif* Lesson Sequence modules, there are cases when, for sound pedagogical reasons, open-ended questions are posed of the learner in order to set the scene, or to activate prior knowledge or schemata (Rubin, 1994), or sometimes in the role of advance organiser (Herron, 1994).

Although the ability to provide immediate feedback is an oft-cited advantage of CALL (Garrett, 1987: 174), the question of whether it is better to delay feedback or offer it immediately has been in contention since the early days of Skinnerian behaviourist psychology. Throughout the sixties and seventies, the positive Delayed Retention Effect (DRE) for verbal learning was the object of numerous studies. Sassenrath & Yonge (1968), for example, though not CALL-based, found that, on 60 multiple-choice items, subjects receiving informative feedback which was delayed by 24 hours produced reliably

higher performance doing a 5-day delayed test of the material taught than did subjects who received immediate feedback. On the basis of this study, Sturges (1969) also found improved performance with delayed feedback, but added that it was the *type* of feedback, rather than merely the delay, that produced the improved retention performance. In his study, subjects who received informative feedback including both incorrect and correct answers performed at a superior level to those who received only the correct answers in their informative feedback.

Subsequent studies took further consideration of the normal circumstances of test-taking in education – that it usually comes after a period of instruction – and endeavoured to determine whether the positive effects of delay of feedback were actually caused by the time-lapse facilitating the forgetting of initial incorrect responses. Peeck & Tillema (1978), administered a post-instructional test to fifth graders in three stages: immediately after the instruction, after a day, and after a week, with the students divided into three groups which received either no feedback, feedback after a delay of thirty minutes, or feedback after a day. A range of different questions types was also used to see whether previous positive DRE findings may have been influenced by question type, but this was not found to be an influencing factor.

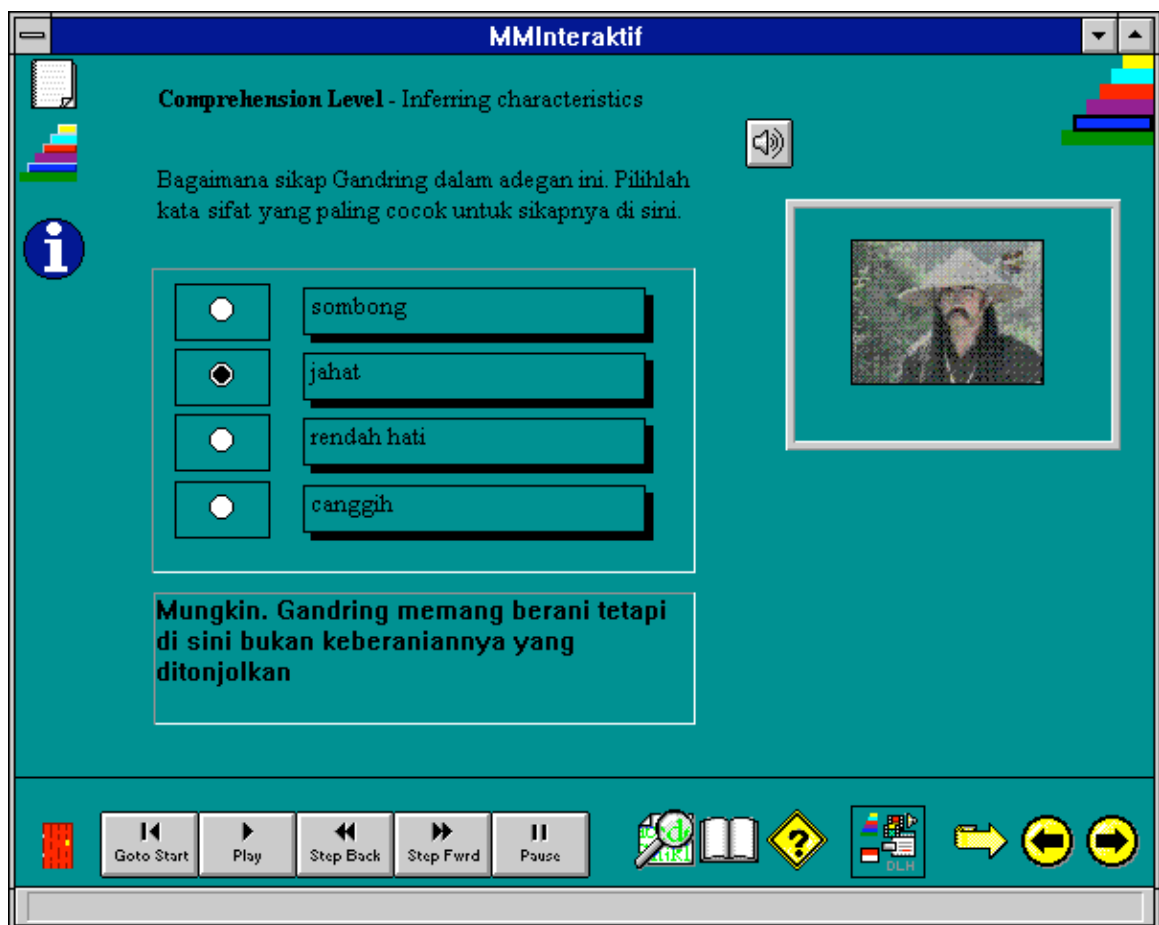
Peeck and Tillema's other findings supported the positive effects of delayed feedback, while showing the greatest positive effects for the slight delay of thirty minutes. Most interestingly, their results did not support the hypothesis that delayed feedback worked because students forgot their original responses, as there was a generally high level of accurate identification by students of their initial incorrect responses. This led the

researchers to hypothesise that it was the memory of incorrect responses, together with the delay in feedback, which actually assisted in improved performance.

More recently, in the CALL environment, the questions investigated have been focused on the nature of the most appropriate feedback. Bationo (1992) has investigated the effects of three forms of immediate feedback on learning, on the principle, derived from earlier studies, that ‘feedback functions both as a reinforcement and corrective information provider’ (1992: 46). His findings, with four groups of undergraduate university French students – one group which received written-only feedback, one group spoken-only, one group both spoken and written, and one control group – show the best performance on immediate recall for the combined feedback group. However, this performance is not maintained on the (two-day) delayed test, which shows no significant difference on retention of the material across all groups.

On the basis of his findings, Bationo suggests that future studies should allow learners to make their own choices on the kind of feedback they prefer, and that feedback be offered in their first language, two suggestions that have been adopted in *MMInteraktif*. One exception is the response messages attached to individual answers in the sample lessons illustrated here, which are given in the target language in order not to interrupt the flow of hypothesis formation (see Figure 5.9 below). However, in the *MMInteraktif* environment itself, being language independent, teacher-authors are free to use whatever language they choose.

Figure 5.9 Target Language Response Message



In an Intelligent Computer-Assisted Language Instruction (ICALI) environment, Nagata (1993) shows that Japanese language students find ICALI incorporating natural language parsing (NLP) and specific grammar descriptions related to their errors more helpful than a traditional CALL program without NLP. However, as Nagata acknowledges, the computing power necessary to deliver feedback based on NLP is large, and even in her experimental situation, the computer takes 3 – 4 seconds to parse each sentence. In addition, current research and technology in the area of NLP can still only accommodate limited sets of fairly regular or standardised language, such as Japanese verb tense or particle contrasts, or the conjugated or inflected verb forms in common European languages such as French or German. These were some of the major influencing factors

in the decision not to incorporate NLP in the *MMInteraktif* package at this stage, but to take the alternative path of restricting free learner input, and use the semi-intelligent response handling as advocated by Joy *et al.* (1983) and Burston (1990).

In her explication of her third desideratum for Intelligent Computer-Assisted Language Learning (ICALL), Oxford (1994) observes that in their provision of feedback, most ICALL developers, with the notable exceptions of Loritz (1993) and Frederiksen (1993), concentrate on learner errors rather than competencies and the provision of language assistance. This desideratum states: 'ICALL must provide useful, appropriate error correction suited to the student's changing needs' (Oxford, 1994: 3). Oxford goes on to urge ICALL developers to consider questions relating to learner assistance such as: 'Just what kind of language assistance do students need? When do they need it? Do all students need the same kind of help? Don't students' contrasting styles of language learning create different needs for help in different learners?' (Oxford, 1994: 6). She also suggests that ICALL developers increase their knowledge and understanding about language learning styles in order to design ICALL systems which accommodate these considerations. Again, this was taken into account in the design of *MMInteraktif*, by adopting a principle of free access for learners to the range of help and feedback facilities.

In an ambitious, 9-day study of the relationship between learner achievement on the one hand, and different kinds of pedagogic approach and error feedback on the other (see Figures 5.10 & 5.11 below), Robinson made exclusive use of computer instruction delivery with Junior High School Spanish students. Because of the short duration of the



study, only achievement on specific curricular items rather than proficiency gains could be tested.

Figure 5.10 **Pedagogical, content variables** (Robinson, 1989: 120)

Experimental Group		Control Group
1. Known others	vs	Anonymous characters
2. Integrated material	vs	Nonintegrated items
3. Meaningful practice	vs	Manipulative practice
4. Emotional/humorous content	vs	Dry facts
5. Background Content:		
Student choice through menu	vs	Program designation
6. Problem-solving activities	vs	Descriptive activities

Figure 5.11 **Error feedback variables** (Robinson, 1989: 120)

Experimental Group		Control Group
1. Student discovery of error correction:	vs	Program disclosure of error correction
• Error location		• Correct answer
• Error location with hints		• Correct answer with explanation
2. Implicit feedback	vs	Overt correction
3. Student-controlled or combined help	vs	Program-controlled or no help
4. Recycling of missed items:		
• Repetition of missed item at random intervals	vs	• Immediate repetition of missed items
• Immediate introduction of parallel item	vs	• Repetition of all missed items together at end

The major findings of Robinson's study are that initial interest and enjoyment of learning the language seem to be the best predictor of achievement in both experimental and

control groups, while ‘meaningfulness’ and ‘student discovery’ are the common variables related to greater achievement (1989: 130 - 1). Thus, in terms of cumulative effects, integration of the computer-based materials with the rest of the curriculum, and reinforcement in other parts of the course, foster greater achievement. Feedback, on the other hand, is most effective, in both the long and short term, when it ‘guides students to discover the answers’ (Robinson, 1989: 131). This is done through locating errors, accompanying them by hints, and offering implicit error correction through modelling and rephrasing.

Most importantly for *MMInteraktif* design, combined student-control and program-control seems to assist in this discovery process, as does the provision of parallel items to foster repeated practice and the use of reasoning strategies. The combination of control is achieved in *MMInteraktif* by the provision of progressively more structured layers, from the Browser, to the Taxonomy, to the Lesson Sequences. Parallel items are provided by means of selections from the Taxonomy.

In an early attempt to allocate more control of CALL software from the program to the learner, Cryle & Lian (1985) found that with increased control, some learners circumvented the program controls entirely and converted the lessons to guessing games, while others restricted themselves too often to word-level attempts, without making full use of the help, feedback, and learning and practice opportunities that the program made available to them. One area for improvement which these researchers suggested was in the speed of access to appropriate places in the audio text, which, it was hypothesised, would increase learners’ inclination to use the practice and tailored review material. With the level of technology now available, this improvement is possible for both audio

(through almost-instantaneous access to digitised audio files) and video (through laserdisc playback facilities, and digitised video either on computer hard drives, or on CD-ROMs). With minor program adjustments, *MMInteraktif* also has the capacity to incorporate media on the newer digital video discs (DVD) and networked digital video. It is this technology for which *MMInteraktif* is designed, for the presentation and exploitation of authentic listening and viewing comprehension texts.

The design of *MMInteraktif* addresses the problem of inefficient strategies used by learners in two ways. Firstly, the language texts provided by the package are authentic texts containing real-world information; and secondly, the tasks in the package represent one resource tool in a complete task-based CELL environment. Learners therefore need to obtain certain information from the materials in the package in order to complete other classroom-based or homework tasks. In addition, the integration of information about strategy use and application, with practice in using these strategies, means that in order to obtain the necessary information, learners are gaining practice in utilising more efficient strategies. In other words, the mode of delivery of the necessary information is the learning strategy environment.

In fact, the behaviours of learners recorded by Cryle and Lian are consistent with findings elsewhere in the literature on computerised training and development. According to Galagan (1987: 75), 'Observers have found that when scoring is present, learners will go through the program to optimize their scores. Their first goal is to win rather than to learn, but then they may go back through the program several times to experiment with alternatives'. Thus, the speed, flexibility and ease of access to the audiovisual media that is now technologically possible in CELL produces an environment, such as in

*MMInteraktif*, which is much more conducive to learner experimentation and self-directed exploration of the learning materials.

From their research into the design of, and experimentation with ‘traditional’ CALL, Demaizière and Blanvillain (1990: 25) have attempted ‘to rely on learners’ spontaneous analyses and intuitions about language’ to tailor their provision of intelligent feedback in their tutoring system. To this end, they identify four approaches to providing feedback to learners, which can be characterised as 1) descriptive, 2) prospective, 3) impressionistic, and 4) adaptative, and where each step subsumes the previous one. The descriptive approach is fairly self-evident in that it provides the learner with an initial description of the error, to signal its occurrence and its location. In their description of the next approach, Demaizière and Blanvillain comment that for the learner (unlike the researcher!) the next answer is often more important than the one already given. The prospective approach therefore aims to provide learners with positive and collaborative feedback which helps them activate their comparative and inductive strategies for the next answer.

Demaizière and Blanvillain formulate their impressionistic approach on the principle that ‘the necessary balance between helpful comments or reminders and overwhelming explanations has to be kept in mind’ (1990: 35). They thus suggest the use of humour or more striking messages, rather than detailed, technical ones, together with some prioritising of feedback on multiple errors, on the basis of the objectives of the session. While Demaizière and Blanvillain base their adaptive approach on the provision of question-specific error identification and the context of the question, as will be detailed in the next chapter, *MMInteraktif* leaves these determinations more to the learner.

All of these approaches have been observed in the Taxonomy and Lesson Sequences layers of *MMInteraktif*. Additional contextual information is added to messages informing learners when they have chosen the wrong response (descriptive and prospective), and access to the layers of grammatical help and audiovisual context attached to any specific lesson is learner-determined (impressionistic and adaptive). As part of their error messages, learners therefore receive contextual information to guide them to the correct choice. In addition, they can access grammar reference notes relating to that specific question, or, if this is insufficient or inadequately describes their problem, they can instigate a more detailed search for the point that is causing them difficulty using standard Windows Help file search procedures. As these search procedures are similar to those used for finding information in print-based dictionaries, learners do not face any unique difficulties with using them. If their problem is related to semantics or discourse, learners can access both audiovisual playback and a transcript of the context of the text segment on which that particular question is based. Alternatively, they can transfer to the Browser layer for a more independent exploration.

In these design features, *MMInteraktif* most closely resembles the hybrid or bridge environments discussed by Sussex (1990: 251), as representing a ‘combination of microworld and intelligent tutoring systems’. Whereas Sussex, following Cumming and Self (1990) advocates the separation of Help into task and discussion levels, in *MMInteraktif*, Help is available as a continuum, at the task level as realised in the Taxonomy and Lesson Sequences layers, and at the Browser level which most closely corresponds to Sussex’s discussion level. In the design of *MMInteraktif*, the Browser was designed first, to incorporate the most common uses of the traditional language

laboratory, and the concept of a relational audiovisual database in terms of its search facilities and the availability of various media. When subsequent levels were designed, it was then possible to incorporate the powerful, pre-existing features of the Browser into the structure of the Help facility. Thus, while it may be conceptually useful to separate task and discussion level Help, particularly in the context of intelligent tutoring systems, in the case of *MMInteraktif* the Help facility forms an integrated whole which is accessible to learners in all layers and at all times.

Student preference is another major issue in the consideration of help and feedback facilities. In one of the few studies to tackle this topic in relation to CALL, Brandl (1995) compared strong and weak students' preferences for different kinds of error feedback among college-level German students working on computerised German grammar exercises. Numerous studies have documented the relative effects of various kinds of written feedback for learners (Hendrickson, 1980; Lalande, 1982; Semke, 1983; Robb *et al.* 1986; Omaggio Hadley, 1993), but little agreement has been reached. As Brandl comments, this disagreement could arise from a number of variables, including those relating to learner characteristics and achievement levels, tasks types, and error types. His study therefore addressed two questions: '(a) When given computer feedback options, what options do students choose?' and '(b) What strategies or reasons determine how learners choose feedback options?' (Brandl, 1995: 199).

Records of students' elicitation of feedback were collected on-line, and an interview protocol was also employed to collect additional information on the reasoning processes they used to perform the activities. Feedback options consisted of display of:

1. A Right or Wrong (RW) message;
2. Show Error Feedback (SE) in which errors were highlighted from left to right;
3. Grammatical Feedback (F) in which errors were highlighted and grammatical hints provided;
4. Right Answer Feedback (A) which prompted students for confirmation that this was indeed what they wanted.

The results of Brandl's study show that high achieving (HA) students employ more exploratory strategies with reference to their errors, preferring to try to find their errors themselves rather than going straight to the correct answer, and use all feedback options more often. In his discussion of the results, Brandl mentions two reasons (after Garner, 1990) for the differences he observes in the strategy use of Low Achievers (LAs): poor cognitive monitoring, and primitive routines that get the job done. Faulty monitoring is demonstrated in Brandl's findings by the higher number of incorrect changes to their answers, and fewer intelligent guesses, on the part of LAs. With regard to the use of primitive (but ineffectual for learning) routines, Brandl's findings correspond to the arguments presented by Donato and McCormick (1994), mentioned in the previous chapter, that learning strategies need to be goal-directed in order for effective learning to take place. Both Brandl, and Donato and McCormick recognise that some students have goals other than improving their learning when making strategic choices, a point for which Cryle and Lian (1985) also discovered evidence in their CALL activities, as discussed above.

There is also evidence in Brandl's study for implicating lack of knowledge in the loss of motivation indicated by LAs choosing to abandon exploration of alternative feedback

forms and instead going straight to Show Error or Right Answer Feedback. Brandl's hypothesises, on the basis of the previous studies by Winne and Marx (1989), and Corno and Rohrkemper (1985), is that Low Achievers lack the linguistic and other information necessary to make meaningful use of the feedback options, causing them to achieve at a lower level, to lose task orientation or goal-directedness, and therefore lose motivation. Low Achiever interactions with the feedback options seem to support this hypothesis in that these students, for example, made numerous incorrect (non-strategic) changes to their answers, before resorting to the Show Error or Right Answer options indicating their assumption that their answers contained errors, and a sense of helplessness.

*MMInteraktif* is designed to avoid the possibility of learners bypassing the exercise of higher cognitive skills inherent in feedback requiring analytical processing, by incorporating strategies requiring these processes in all tasks in the Taxonomy, from the Knowledge Level upwards. When sufficient student interaction data is collected, the feedback facilities should provide a large amount of useful information on the strategies for gaining help and feedback employed by learners at different levels in a CELL environment. This will be discussed further in a later chapter. As Brandl concludes:

[...] the issue at stake here [...] is not so much an ideally tailored feedback message, but rather the impact of various kinds of feedback on the students' learning process. As the findings of this study indicate, generically stated explanations may not meet the needs of each particular student. Nevertheless, they may have a positive influence on students' reasoning processes and aggregately work to help them with their questions.



In sum, the computer can be implemented as a learning tool to support and motivate the students in their strategy use in language learning.

(Brandl, 1995: 209)

In a related study, though in a non-CALL environment, Aljaafreh and Lantolf (1994) examine the role of negative feedback as regulation and second language learning in the Zone of Proximal Development (ZPD), a Vygotskian term referring to ‘the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers’ (Vygotsky, 1978: 86). The actual level of development is a product of the completed developmental cycles, while the ZPD represents the potential learning of the individual which can be activated through the assistance of, or interaction or collaboration with, ‘more experienced members of society’ (Aljaafreh & Lantolf, 1994). These researchers discuss three mechanisms of effective help in the ZPD: graduated intervention, contingent help, and dialogic negotiation between the novice and the more capable individual.

The mechanism of graduated intervention hinges on determining a learner’s ZPD through initially offering highly strategic or implicit help, and progressively adjusting this to the more specific and concrete until the learner can make meaningful use of it and respond appropriately. As part of this process, therefore, help is contingent on the learner’s needs, and more explicit help than necessary will not be offered. Thus, when a learner begins to exhibit signs of independent functioning, and therefore control over a problem, the external help is withdrawn. From the combination of these two mechanisms arises the dialogic nature of the negotiation of help. In their study, Aljaafreh and Lantolf observe

five ‘levels of transition from intermental [external intervention] to intramental [learner-internal intervention] functioning as the learners moved through the ZPD toward self-regulation and control over the target structures’ (1994: 470).

On the basis of the observations of Burston earlier, and the findings of Brandl, and of Aljaafreh and Lantolf mentioned above, several principles of a help system can be extrapolated:

- some explicit, problem-specific help needs to be provided for learners, particularly those at lower levels of proficiency and/or independent learning;
- help needs to be layered, such that more motivated or independent (intramental) learners can extend their ZPDs; and
- help needs to be easily accessible and easily dismissed or bypassed (to allow both more independent learners, and learners with lower proficiency, and/or less independence, the facility to find their appropriate levels of assistance).

These principles form the basis of the layering of help and feedback in *MMInteraktif*.

#### **5.4.3.3.3 Learning strategies and negotiated feedback models**

Liou (1995) has specifically addressed the effects of awareness-raising of strategy use on language learning in a multimedia environment. By incorporating the on-screen display of messages and questions relating to learner strategy use, she simultaneously collected on-line ‘think-aloud’ data from learners interacting with the laserdisc lesson environment, and provided them with information on improving their strategy use. This data was then

compared with a control group who received no strategy information messages. Unfortunately, because of lack of machines, learners were only able to use the strategy program once, and therefore, according to Liou, the results might not be as statistically significant as they could be. In spite of this, however, she did find that the strategy group out-performed the control group on almost all of the post-test and delayed post-test measures.

From an artificial intelligence (AI) perspective, Bull *et al.* (1993, 1994), Bull (1994), and Bull & Smith (1995) have also experimented with the inclusion of learning strategies into language learning materials, by incorporating them into the Student Model component of their ILE (Intelligent Learning Environment). In an extremely ambitious and forward-looking project, Bull, Pain, & Brna (1994) describe an attempt to extend the concept of the student model in an ILE system to include more than just the students' domain knowledge, but also their learning strategies, and language awareness and reflections. The domain knowledge itself has been enhanced by the incorporation of understandings of interlanguage development, specifically developmental sequences in the acquisition of certain morphemes (Pienemann, 1989), and language transfer (Kellerman, 1977; Odlin, 1989), from research into second language acquisition (SLA).

Bull (1994) and Bull *et al.* (1993, 1994) hope to enhance learners' language awareness by allowing learners a dialogic interaction with the system. The system constructs its representation of learners' beliefs about specific grammatical structures of the language on the basis of learners' achievement on activities based on these structures. The dialogic interaction then allows learners to negotiate and refine the system's representation. This will ultimately lead to the cooperative construction and repair of the student model. In

this endeavour, the researchers have relied on Ellis' (1992) conclusions that 'although practice may have some value, what is more important is the development of explicit knowledge' (Bull *et al.* 1994: 7), that this consciousness has a delayed effect on implicit knowledge, and that by integrating practice and awareness-raising, the explicit knowledge will then be available to learners when they are ready to process it. The relationship between explicit and implicit knowledge and consciousness has already been discussed at length in sections 3.3.1, 3.41, 3.4.4.1, 3.4.4.4 and 4.2.1 of earlier chapters. As mentioned in the conclusion of the previous chapter, and the introduction to this one, these are also the principles on which the integration of learning strategies and language practice are based in *MMInteraktif*.

In a more recent research and development study, Bull and Smith (1995) have tackled the incorporation of targeted negotiation, not only to the student model of an ILE, but also to the domain knowledge itself, in this case the creative domain of melody composition. While both these developments are exciting and innovative, and could well change the previous conceptions of learners and learning in ILEs and AI, they are still very much at the prototype stage. As mentioned above in relation to AI, ILEs, and ICALL, while the computing power of machines is still expanding dramatically, the level of hardware power typically available to learners in educational institutions remains limited, and until this situation changes, the use of such systems for language learning will also remain limited.

In addition, as has been noted elsewhere, because of the complexity of language systems (Legenhausen & Wolff, 1990: 12; Swartz, 1990a: 220) it will be some considerable time before a reasonable ICALL system will be programmable, especially one which

incorporates the concept of a negotiated student model, as described by Bull *et al.* (1994). Even in the current model, the application is restricted to a very small number of discrete grammar points for English L1 learners of Portuguese as a second language.

#### **5.4.3.4 Control options**

Hubbard makes the comment (1992: 61) that while the issue of Control Options is a complex one, the CALL environment lends itself to allowing the learner considerable control in areas such as choice of activity type, mode or content of presentation, navigation, display of help or hints, and rate or pace of presentation. As will be detailed in the next chapter, the CELL software under discussion here allows learners control in these areas, together with deeper layers of help and exploration, and provides learners with the infrastructure, the information, and the practice to develop their learning strategies towards achieving a higher level of self-managed learning, or even autonomy. In particular, the three layers of access to the package – Lesson Sequences, Taxonomy and Browser – the on-line information about learning strategies, and the practice mode assist learners in their progress towards autonomous learning. As Frese (1987: 43) comments, ‘experiencing control means to have an impact on the conditions and on one’s activities in correspondence with some higher order goal’ and ‘if the environment does not provide the freedom to decide, the person does not have any control’.

In work environments, lack of control has been shown to induce stress, which is also implicated in feelings of helplessness and physiological effects (Seligman, 1975; Weiss, 1977; Karasek *et al.*, 1981). In a computer training environment, subjects who ‘were asked to develop their own hypotheses about the [computer] program they were about to learn and [who] were encouraged to explore’ (Frese, 1987: 48), performed consistently

better on various proficiency measures than subjects in a lock-step group, and were better able to transfer this training to another task. For our purposes, the two major aspects here are the determination of one's own goals, and the freedom to decide what actions to take on one's own initiative. The rationale behind the decisions made on how to provide and represent this control is discussed in the section on Screen Layout below.

#### **5.4.3.5 Screen Layout**

For Hubbard, the major determining factor for Screen Layout is the Presentational Scheme. However, in the design of a learner-managed CELL software package, a larger issue is how much of what components of the software to allocate to learner control, and how to assist learners to develop the strategies to benefit from this control, and to access the available features in an intuitive and consistent manner. Thus, some of the major aspects of screen layout on which decisions have to be made are listed in Figure 5.12 below.

Figure 5.12 **Screen Layout Considerations** (Source – original)

- differentiating between global (all tasks) vs local (single task) controls;
- task controls vs media controls;
- layout of interactive sections:
  - size and spacing of text/fields/buttons,
  - position (convenience vs confusion),
  - nature (fixed-size field, scrolling, expanding),
  - features (fixed, moveable, highlight, sound),
  - graphics, icons,
  - animation,
  - colour (which can attract or detract - England 1989),
  - allocation of screen 'real estate' to display task vs video,
  - quality and size of video window;

- maintain vs overwrite existing menu line;
- operation of navigation (mouse, keyboard, touch-screen etc.);
- links with other peripherals/facilities (videodisc, video overlay cards, CD-ROM, network);
- links with other screens and program components;
- entry, exit (& navigation) modes;
- any special features/emulations.

Gordon (1994: 107-8) has several useful guidelines and reminders for screen design, specifically in the area of layout of interactive sections. These are listed in Figure 5.13 below. Items 1. to 6. are clearly related and fairly common-sense points. However, very little research has been carried out in this area: how much information on the screen is too much, for example, or how can the notion ‘moderately’ packed be quantified? On the basis of findings in the areas of learning styles and strategies, we have seen that these decisions may, in fact, be based upon distinctly individual preferences. One area in which some conclusions can be found pertains to the size of the screen display in relation to both user performance (in this case, reading and comprehension), and user preferences. Reisel and Schneiderman (1987) compared novice computer user performance on a series of tasks requiring considerable on-screen reading and comprehension, and found that users on the largest of three screen sizes performed significantly better statistically than did those on either the intermediate or smallest screens. User subjective reactions also showed distinct preference for the largest display, though some concern was expressed by a significant minority of subjects that the largest display might prove cluttered in some circumstances.

Figure 5.13 **Gordon's Screen Design Guidelines**

1. Don't try to place too much information on the screen **but** include all relevant information;
2. Group characters such that they are moderately packed;
3. Items on the screen should be logically grouped;
4. Leave adequate space around each group;
5. A smaller number of groups is better than a large number of either groups or individual items;
6. Minimise the complexity of the layout;
7. Maximise visual predictability;
8. Use more than one signal to provide the same information e.g. colour **and** style;
9. Choose icons and symbols for input buttons that are consistent with learners experience of other applications or programs;
10. Strive to use words **and** icons;
11. Do not use hidden buttons, and if actions are to be sequenced on the one screen, remember that people's eyes usually scan from the upper left;
12. Use combined upper and lower case for text;
13. Use reverse video sparingly;
14. Messages should be brief, concise, specific, helpful, and comprehensible;
15. Use instructional prompts when learners are required to perform an action.

Another area of particular significance for designers of graphical user interfaces is the nature, appearance, meaningfulness, and location of icons on the screen. The issue of the meaningfulness of signs, particularly for the purposes of the graphical or intuitive representation of the function of an object is by no means exclusive to the computing world (Norman, 1988). However, in the case of CELL software, it is especially important for the instructional interface to be as language-independent as possible. This is because the end-users of the software may be speakers and readers of any first language, learning a range of second languages. One way of achieving this ideal is to use icons that are cross-culturally meaningful. However, this is no easy task, since, as we have seen in



previous chapters, the assumption that values can be transferred wholesale from one culture to another is unpredictable. In an excellent attempt to summarise iconic systems, evaluate their meaningfulness, and make some useful recommendations for instructional designers, Wood and Wood (1987: 97) define an icon as ‘any symbol, image or pictograph used to represent a concept, idea or physical object’. The three iconic systems they discuss are the universal language of medieval shop front signs, modern road and highway signs, and Chinese pictographs now represented as characters.

Wood and Wood regard the concrete symbols hung in front of medieval shops to have been effective because they were ‘readable’ across languages, and by illiterates. For the computing environment, this means that graphic symbols need to be chosen on the basis of a visual representation which is ‘transparent’ across languages and to computer novices: in other words, images that are found consistently in daily life in various cultures. In *MMInteraktif*, for example, the image of an open book is used to indicate where grammar notes can be consulted, and a door closing represent the action of exiting from the program.

Road and highway signs are noted as being significant by Wood and Wood, because of the variety of means such signs employ to represent not only words, and single ideas, but whole sentences and complex meanings. Thus, for example, the shape and colour of a sign may provide symbolic redundancy in iconic representation (yellow diamond = warning + pictograph of the actual danger in the centre). In graphical interface design, this means that it is important to choose basic icon elements carefully to ‘allow combinations that go beyond the simple semantics of these images’. A magnifying glass on top of an opening page, common in commercially-available programs for example, is

used to indicate where a learner in *MMInteraktif* can get more information or context for a segment of text. The presence of the icon, and its location, indicate that it is a link to something else, the magnifying glass indicates that a closer view is available, and the image of the book represents a source of reference.

The discussion of Chinese characters in relation to graphical interface design serves to remind designers that complex concepts can be represented iconically (the hourglass or watch symbol for ‘the computer is busy’); that, as with characters, combinations of icons can produce new meanings; and that, in both Chinese characters and computer icons, evolutions occur over time, from pictographs to increasingly arbitrary representations. A timely message Wood and Wood have for software developers is that the need of the user to feel comfortable with familiar surroundings in a new package is more important (for usability, popularity, and sales) than the need of the software designer to appear unique.

Sanders (1987) takes the perspective of a novice user of personal computers in a presentation of his personal reactions to learning to navigate new interfaces. He cites the research into hemispheric differences, as discussed in Chapter 3, and the relations between imagery and spatial internal representations, as being influencing factors in the preference he observes for combined spatial and verbal tasks. This observation also corresponds to some of Gordon’s guidelines presented earlier in Figure 5.13, specifically numbers 8. and 10. to do with the need for redundancy of words and icons. Sanders does, however, warn of the necessity for ‘correspondence between the “behavior” of the machine and the nature of existing human internal representation’ (1987: 227). In *MMInteraktif*, this warning has been heeded in the use of ‘drag and drop’ visual-kinesic techniques for some of the lesson templates. Thus, in a matching task, for example, the

name of a person can be dragged from an on-screen list and placed under the picture of the appropriate person.

Sanders also raises the major issue of the importance for learners to be able to build up an 'integrated set of cognitive schemata or an "internal model" of the man-machine interaction' (1987: 228). In *MMInteraktif*, learners are assisted in doing this by the provision of several organising screens or program maps on opening the program, and on opening each of the layers. These organising screens serve two functions: to provide learners with information on the program organisation, the contents, and the navigation; and also to give learners access, via built-in links, to the sections described. In addition, reminders of the program structure are constantly on-screen in the form of miniaturised icons of the various layers, as well as all navigational features.

In keeping with the advice of Wood and Wood (1987) above, and Sanders, the navigational features of *MMInteraktif* have been designed to reflect common sense, general understandings of how things operate (Norman, 1988), and other familiar iconic representational systems, such as the question mark icon for Help. In addition, navigational approaches have been adopted from existing programs and applications in common use, such as the incorporation of the Windows Help files architecture for accessing and searching the complete Grammar reference notes by topic.

Further to her more general guidelines above, Gordon (1994) also details guidelines relating specifically to the use of colour. These are outlined in Figure 5.14 below. These guidelines are based to some extent on the findings of researchers such as Martin *et al.* (1987) who, having found few guidelines in the research literature, investigated the

relative costs and benefits of five common forms of highlighting and presentation, namely: an arrow cue, a blinking cue, close boxing, red highlight on a light green background, and reverse video. Their findings show that the use of red colour highlighting produced the most benefit with very little cost (number of errors), while the largest number of errors with the least benefit was produced by both the reverse video and the boxing cues. The researchers attributed this to perceptual distraction rather than a lack of potential to attract attention. They also hypothesise, as a result of their findings, that ‘while the subject is attempting to process all of the locations in a parallel fashion, the color highlighting cue might allow a certain amount of automatic (resource free) focusing of attention on the highlighted location’ (Martin *et al.*, 1987: 95).

Figure 5.14 **Gordon’s Guidelines for the Use of Colour**

- Do not use bright colours for large areas;
- Do not use too many colours on a screen (suggested maximum: 9 colours);
- Use colours consistently for areas of the screen with the same function;
- Limit colours used for coding (suggested maximum: 8 colours);
- Use colours to draw attention, communicate, or organise information;
- Do not use colour images or text on colour background (i.e. use grey scale);
- Avoid using bright blue for text or small thin-lined graphics;
- Avoid the combined use of opposing colours, highly saturated colours, or colours far apart on the colour spectrum.

In another study on readability in relation to text and background colour, Fukuzumi *et al.* (1987) found that readability depended on colour difference, with the best condition being produced with a background colour of yellow-green. This is, therefore, the colour chosen for background in the Taxonomy and Lesson Sequence Layers of *MMInteraktif*.

As Sanders comments: 'The more a novice can capitalize on already available internal models, the less there is a need for learning, and the more the system is user-friendly' (1987: 229). In *MMInteraktif* where, as repeatedly stated, the aim of the program is to provide learners with a learner-centred mode of access to improving their listening and viewing comprehension, it is critical that minimum cognitive processing capacity should be required to working out how to use the system, how to navigate through it, and how to get the desired information. The *MMInteraktif* package has therefore been designed on the basis of current understandings of the most user-friendly aspects of instructional and interface design.

## **5.5 Summary and Conclusion**

Computer technology can now be used to manage learning by making available to learners an environment in which there is a range of useful language learning material, feedback which assists them in evaluation of their own progress, and a varied help system which provides them with choices from a variety of sets of information, whether it be in the form of dictionaries, transcripts, replays, or grammar explanations. This is the role of computers which has formed the basis for the instructional design of the computer-assisted listening activities discussed in this chapter and the next. By incorporating this management model with the learner-centred philosophy described earlier, a system has been designed in which learner control over the flow of the 'lessons' or activities is enhanced, while at the same time they are able to improve their listening comprehension skills and develop their facility in a range of learning strategies, particularly cognitive and metacognitive.

In this chapter, we have discussed the principles of instructional design followed in the construction of the architecture of *MMInteraktif*, and the rationale behind these principles. In the next chapter, the actual architecture of the system, and flow of control through it, will be presented. This will include a description of the components of the graphical interface, accompanied by a stage-by-stage commentary of how these components are linked, and how learners can use them. Some examples from Indonesian language learning will be used in illustration.