Towards a Synthesis of Multimedia and Intelligent Tutoring Systems

A dissertation presented in partial fulfilment of the requirements for the degree of Master of Science in Computer Science at Massey University
Palmerston North, New Zealand.

Emma A. K. Vethanayagam

1998

Abstract

Multimedia is being used almost in every field. This study is about the use of multimedia in the area of intelligent tutoring systems. This project studies the advantages and disadvantages of interactive multimedia and intelligent tutoring systems , and analyses the ways of combining these technologies in search of an interesting, learnable, flexible, compelling and technology-enhanced educational tool.

Educational packages need to be evaluated for effectiveness. When it comes to computer-based instruction, technical concerns such as multimedia effects are taken seriously and there is not enough emphasis on its educational value. There is not much concern about the appropriateness of the instruction method to the computer medium. This research proposes a framework for evaluating educational packages which include a number of issues.

Several pieces of educational software were evaluated using this framework and *Diagnosis for crop protection*, a multimedia software package that aids in teaching the process of diagnosing crop problems, was selected for modification, as a practical application of the theoretical work.

We studied different multimedia system development models and methodologies. We also analyses the cognitive issues and intelligent features that enhance the learnability.

Finally, the appropriate intelligent features and other factors that could enhance *Diagnosis for crop protection* to be a more 'active knowledge constructing' environment have been identified. The current version of *Diagnosis for crop protection* was represented using an appropriate methodology and the proposed changes were described in detail.

Acknowledgments

First of all, I am eternally grateful to my uncle Stanny Emmanuel, for his moral and financial support in everything to initiate my masterate degree. I would like to acknowledge New Zealand government and the NZODA postgraduate scholarship office for sponsoring the second year of my program.

Next, I would like to express my sincere gratitude to Ray Kemp, my supervisor, for helping me find some suitable area for my research, excellent supervision, valuable ideas and very prompt reviews. I really enjoyed working with you.

I would like to thank Terry Stewart, for granting permission to use *Diagnosis* and for being helpful in a totally unfamiliar subject area of plant pathology. Thanks to Alex Doyle-Bolecivic, who kindly lent a few packages for studying.

Thanks also due to Computer Science department staff and postgrad students, who provided a nice environment for me to work. I would love to thank all my friends and flat mates for their friendship, encouragement, letters and emails. I absolutely enjoyed the funny things we did together.

A very big thank you for my parents and grand mother for their love and prayers. Thanks to my loving brothers and sister for all what they are. Finally, thanks to Raymond for his love and for constantly being there for me all the time. I am very proud of you all.

Without you all, this dissertation would not be the same.

Thank you very much.

Table of Contents

3.0 Evaluating Educational Software

4.0 Designing Intelligent Multimedia Tutoring Systems

5.0 Cognitive Issues, Intelligent features and Human Issues

6.0 Diagnosis : From Game to ITS

Figures

Chapter 1

Introduction

1.1 Background

At the rate computers are being introduced in the areas of business, communication and entertainment it is inevitable that the field of education be compelled to accommodate computers in its curriculum at the same rate. Computer-based tutoring systems have been in use for the last few decades. And some of them are proved to improve learning.

Computer-based learning systems have many advantages over traditional instruction. Reduced learning time, instructional increased retention, increased motivation, increased safety and flexibility are some of these advantages. However, these advantages are subject to several conditions such as standard of the student, subject matter and target population. So, it is understandable that the success of computers in education is dependent upon how we use them.

Early computer-based systems used *programmed learning*, which were based on the behaviourist psychologist theory (Skinner, 1958 from Yazdani, 1987). Then the trend changed to Computer Aided Instruction (CAI) and Computer-Based Training (CBT). Later on, in the 1970's, with Carbonell's suggestion of introducing Artificial Intelligence to CAI, Intelligent Computer Aided Instruction (ICAI) systems were developed (Carbornell, 1970). In early 1980's Hartley and Sleeman (Yazdani, 1987) defined a model for these tutoring systems and these systems are called Intelligent Tutoring Systems (ITS).

There are different types of educational software such as tutorials, drills, tests, games and simulations. Tutorials, drills and tests are based on the theoretical aspects of knowledge, whereas simulation is based on the experiential aspects of learning.

The multimedia phase of tutoring systems is relatively new. And yet, it seems like radically changing the form of tutoring systems. Hodges and Sasnett (1993e) identify the return to nonliterate forms of documents and the development of simulation and visualisation as fundamental forms of expression as the reasons behind the growth of multimedia.

Multimedia systems combine a variety of information sources, such as voice, graphics, animation and video into a wide range of applications. The big picture shows multimedia as the merging of three industries : computing, communication and broadcasting (Furht, 1994). If computers does for images what it has done for text and equations, the result could transform work and play (Adam, 1993).

It seems as the computer is becoming more fun and more useful the less abstractly or in other words more precisely it can represent things. One plus point of multimedia systems is their ability to dynamically depict changes. It enables students to partially experience, if not fully, situations they would not otherwise have a chance to encounter. Multimedia systems hold the promise of great benefits in terms of increased productivity, efficiency, effectiveness and information usage. Multimedia enhances human understanding of complex information through better presentation technologies and appropriate combinations of these technologies for information presentation.

The computer is only a tool, which could be used with positive or negative results. This is true in the field of education as well. Whether potential benefits that the computer seems to offer are actually achieved depends largely on the teachers who use it. Neal and Shapiro (1994) mention in their MIT project report that their aim is to present this potential to teachers to stimulate their creativity and allow them to develop their ideas with a minimum of resistance and delay. This thesis is developed with the same aim, to make it easier for the teachers to present their problems in plant pathology, in a way that is interesting and learnable to the students.

1.2 Multimedia Tutoring Systems

Education has been one of the first and best clients of multimedia computing systems (Hodges and Sasnett, 1993d). Multimedia systems do not imply immediate changes from the design of today's curriculum, which evolve with social changes. But it suggests a redefinition of the style of learning to form interactive and multisensory information presentation.

In his work on Electronic Books, Webster (1993) points out the advantages of electronic books compared to paper-based books.

- Automation of cross reference, which may result in changes in the way people interact with information
- Ability to include dynamic illustrations
- Ability to dynamically modify and refabricate books by adding annotations, filtering or reconfiguring the presentation of their contents, or fabricating entirely new materials, either by hand or automatically.

As ITSs are emerging, the need for individualised multimedia explanations has surfaced. Multiple media can concisely and accurately convey required information and promote better learning. Multimedia explanations can increase the fidelity and bandwidth of the communication to the student. It may even employ AI, natural language generation and user modeling techniques to provide on-the-fly, individualised explanations.

Multimedia goes beyond textual explanations, incorporating one or a combination of graphics, video, speech, and sound to convey information. ITSs are making a jump to multimedia to provide explanations and help in the form of multimedia presentations (Goodman, 1995). Intelligent multimedia systems must address the complexity of multimedia explanations if they are to promote a better learning environment.

As the involvement of the multimedia interaction between the student and the courseware technology rises, the practice and coaching increases in quantity, complexity, realism and fitness. Integrated media is even more successful if the information conveyed in one medium is woven with that provided in the other medium/media and not simply presented in succession. It is easier to remember such material by associating a lot of details with a spatial visualisation than on its own (Miller, 1986 from Goodman, 1995). It is also perceptually and cognitively less taxing for the student during the learning process since the details are spread across more than one medium and trigger more than one sense.

It is understandable that it is the content of educational material, not the technology that determines the success of learning. However, presentation of the material is very important in learning.

1.3 Scope of the project

A few multimedia packages and intelligent tutoring systems were examined for the purpose of understanding their features and identify what features they need in addition to the available ones to be a good software. A brief description of these packages can be seen in Appendix A. These packages mentioned were tried out and subjectively evaluated for better comprehension of existing systems and selection of software for enhancement.

Some interesting points were noted during this sessions.

- Colourful features of the software (*OzSoils, RAS, Diagnosis*) prepare the user for learning by getting their attention.
- Authentic looking presentations (*Ozsoils, Dignosis*) are interesting to try out.
- Non-linear menu structures (*OzSoils, RAS*) give greater user control.
- Knowledge tracing and individualised, step-by-step instruction and help (*ANGLE, The Equation Solving Tutor*) encourage learning for novice users.
- Flexible systems (the ones that have authoring systems to include new problems - *ANGLE, The Equation Solving Tutor, Diagnosis*) are the more useful.

For each interesting feature there are more in depth issues that are yet to be formalised. The following questions can be respectively asked for each of the points mentioned earlier.

- Does the attention last until the end of the session?
- How closer we can get to the real- world? Is it possible to be real? If we have to stop somewhere near the reality, what is the degree of reality we have represent?
- Do the users make use of the freedom effectively?
- Is knowledge tracing possible for all domains? If not how can we imitate individualisation ?
- Is it possible to build problems as we do manually? How to enhance authoring facility?

After this subjective appraisal of these packages, it was decided to develop a general framework for evaluating educational software so that they could be judged more objectively, and in a way, neutrally. The framework was designed and used to evaluate the packages and *Diagnosis* was selected for modification.

Diagnosis does not have any formal documentation. So, it was decided to look at the design aspects of multimedia tutoring systems. We looked at design methodologies to see which methodology or methodologies could have been used to properly design Diagnosis. We also looked at different design issues to be considered while development and used the ideas for alteration of Diagnosis.

Development of good educational software is not only about good design. Hence, we looked at different cognitive and human issues and intelligent features that may increase learning. These theories were appropriately incorporated in the proposed package.

Based on this theoretical work, a detailed description of proposals for modification of Diagnosis were made.

1.4 Aims and Objectives

The main aim of this thesis is to see if it is possible to combine multimedia and intelligent tutoring systems and to analyse how it can be achieved to optimise the efficiency of this combination. The theoretical part of the aim is to search for answers to the questions raised in Section 1.3. The practical part is to alter *Diagnosis for crop protection* for better learning and easy authoring of problems. This aim was broken down into the following objectives.

- Theoretically analysing the role of AI and multimedia in Education. Practically trying out some multimedia and intelligent tutoring systems.
- Developing a framework for evaluating educational packages (Detailed description of the framework, hierarchy of characteristics of good educational software)

- Discussing technical and philosophical issues of learning system design and selecting the appropriate features for proposed changes for Diagnosis.
- Describing the existing system of Diagnosis (Use cases, Object Diagram and Interface slides)
- Detailed description of proposed changes (Textual description, Algorithm with test data for alterations and screen design)
- Ideas on further enhancement for Diagnosis

1.5 Organisation of research

Chapter 2 is a review of literature in the areas of intelligent tutoring systems and multimedia systems, which forms the basis for the rest of the work. Chapter 3 is about evaluating educational packages. It reviews several packages in the light of the proposed framework. This chapter reveals the plus, minus points of each package and highlights the areas to be improved.

Chapter 4 is on designing interactive educational software. This chapter mostly considers the technical design of educational software and discusses whether we really need a different design paradigm to design multimedia systems. It gives a detailed description of a educational software *Diagnosis for crop protection*, which is to be altered for theoretically better learning.

In contrast to Chapter 4, Chapter 5 is about the educational side of the design process. It elaborates the design considerations with regard to the education value of a piece of software and achieving its objectives. Chapter 6 proposes the alterations to Diagnosis, based on the design considerations discussed in Chapter 4 and in Chapter 5. Finally Chapter 7 contains the conclusion of the work and further work.

Chapter 2

Artificial Intelligence and Multimedia in Educational Software

2.1 Introduction

Computers are progressively infiltrating education. It appears that computer technology will play a significant role in education of the next millennium. In an information society, it is expected that computers will be used for teaching and learning.

Computer-based tutoring systems have been used in education for more than three decades. It started in the 1950's, with the development of a type of learning software called *linear programs* (Yazdani, 1987). Even though it was not a huge success at that time, computers did make an impact on education and training.

Techniques used in computer-based education can be roughly classified into *Instructional technology* and *Learning technology* (Zhao, 1995). *Instructional technology* is about teaching the students. In a sense it is controlled learning. Rather than letting the students do whatever interests them it guides them to learn predetermined material. In contrast, *Learning technology* facilitates learning by providing a realistic environment for the student to explore and learn.

Multimedia has taken the world by storm. Commercially it is being used very much for *edutainment* - a blend of entertainment and education (Adams et al., 1996). Multimedia's success led on to its use in education. Obviously, multimedia is very effective in building realistic situations for the students to learn by doing compared to traditional text- based systems.

Multimedia is important in education because it holds great promise for improving the quality of the education. Multimedia provide easy access to information that is rich with images, sounds and text. It is easy to create new ways of communicating ideas and explaining difficult concepts (Ambron, 1988).

This review describes the advantages and disadvantages of Interactive Multimedia (IMM) and Intelligent Learning Environments (ILE) and Intelligent tutoring systems (ITS), and analyses the ways of combining these technologies in search of an interesting, learnable, flexible, compelling and technology-enhanced educational tool.

2.2 Effective Learning

'The essence of education is timelessness' (Smith, 1986). Most of the material and skills that a student learn should be able to be applied to different situations and at different times. What the student learns should not always be outdated with time. The students should be taught to acquire generalised skills. Students have to develop clear insight of the subjects they learn and innovative ideas on the subjects. They must be able to restructure the skills to apply them appropriately in different situations.

Learning is the process of progressively acquiring a more complete subset of the expert's knowledge units (Ohlsson, 1987). For effective learning, presentation of the material to be learnt should be simple, clear and unambiguous (Adams et al., 1996).

Children start learning by trying out what they want to do. But, when it comes to formal education at school, they are restricted to the curriculum. They have to be confined to that circle, suppressing or giving up their own interests and curiosity to do things.

Interest is a terrible thing to waste (Schank, 1995). Literally, every invention is made out of curiosity of the inventor. Learning should be interesting enough for the child to be engrossed in it and enjoy it. But it is difficult for a child to find something interesting within a restricted list of topics that are available in school. Schools generally do an effective and terrible damaging job of teaching children to be infantile, dependent, intellectually dishonest, passive and disrespectful to their own development capacities (Papert, 1984).

There are different teaching architectures that are used for effective teaching and learning, such as simulation-based learning by doing, incidental learning, learning by reflection, case-based learning, acquiring from cases, learning by exploring and goal-directed learning (Schank, 1994).

Papert (1984) says knowledge acquired by personal appropriation is forever a personal power, a special rootedness in the sense of oneself. Students learn in the

deepest way in a subject when they have *intellectual love affairs* (Papert, 1987) with the subject. They tend to learn more when they take charge of their own learning.

In the beginning of the computer era, computers had more control over the users. With the increment of the user control, the computer has become a servant rather than a master. This led to the idea of teaching machine to be replaced by the idea of a learning machine. A review on educational technology reveals a paradigm shift from Instructional technology to Learning technology (Zhao, 1995).

2.3 Why do we need computer-based tutoring systems?

Ancient Tamil literature mentions a learning style called *Guru kulam* (Mahabharatham, 9th century). In this method the student stays with the guru's family throughout the course of learning. The student had to follow the guru's instructions without any hesitation regardless of the hardships involved. Besides the tutoring by the guru, the student is expected to learn by observation. The student had no control over the learning process. Selection of subjects (mostly life skills), tutoring hours, time for graduation are decided by the guru. Leaving all the negative points in this method, compared to today's learning styles, one good thing about this method is that the student was under the focus of the guru all the time. The student received highly personalised instructions and spontaneous feedback at the right time.

Times have changed. Cultures have changed. Attitudes of teachers and students have changed. Teaching and learning methods have changed. Numbers of students have increased very much and there is a scarcity of good teachers.

In a class of, say, some 25 students, within a very limited time frame, it is impossible for a human teacher to make all the students learn something. One way or the other, few of the students are going to be left out, in the sense, they are not going to learn what ever they are supposed to learn. Every child is unique. Every child has her own interests, own style of learning and own pace of learning. They need individual attention. Computer-based tutors can alter this situation drastically by providing individualised instruction for each student.

Moreover, learning rate depends on factors such as a person's background knowledge, culture and personal interest. Computer-based tutors could present users with tasks that interest them. That allow the student to be inquisitive and explore and learn. They can put learners in control of the task.

Social changes have implied the students opt to study at their convenient time. Computer-based tutors can be accessed at any time and at any place which makes the learning process easier for students and the teachers.

Students want to look 'cool' among the peers. They fear failure which causes embarrassment and make them lose 'cool looks'. Computer-based tutors offer users the possibility of recovery from failure without embarrassment. As Papert (1987) describes in his concept of 'Micro world', a computer-based learning environment is a safe place for exploration. The student will not get into trouble, nor will they feel 'stupid' or embarrassed with lower grades.

On the other hand, it is true that in reality, the student should be able to communicate with others and be able to express their ideas. In the real world they have to speak up for themselves regardless of what others might think or say. A properly designed computer-based learning environment could help the students in gaining the confidence to face reality, by providing a safer environment to try out before they face the real one.

Computer-based learning can provide access to multiple experts at appropriate moments to enhance learning. Expert systems also serve as tutoring systems. They could also be used for imparting knowledge from an expert in a field to a larger number of trainees (Yazdani, 1987).

There are complaints such as computers have not improved education and computer experiences are inferior to real ones (Lawler, 1987). But, evaluations by Marchionini and Crane (1994 in Marchionini, 1995) have demonstrated how technology provides both mechanical advantages over manual approaches and enables learning experiences that would otherwise have been impossible or highly improbable. Likewise intangible benefits such as high levels of satisfaction and motivation, improved self-esteem, a technologically skilled work force are often noted as outcomes of technological innovation in education.

2.4 Intelligent Tutoring Systems (ITS)

The earlier computer-based tutoring systems are known as Computer Aided Learning (CAL) packages. These were called *linear programs* and influenced by the behaviourist theory (Yazdani, 1986). In the beginning of a linear program the computer will output some text and the student will respond to it. After a sequence of such activities, finally the linear program informs the user whether s/he is correct or not. Most of the early CAL packages did not have diagnostic aids to trace the fault (Yazdani, 1986).

Intelligent Tutoring Systems are the result of applying Artificial Intelligence techniques to CAL. CAL packages enable the students to practice what they have learned, while ITS are supposed to be diagnostic. The development of one of the early Intelligent Computer Aided Instruction (ICAI) systems SCHOLAR is an effort in making computer programs to exhibit intelligent behaviour. It is appropriate to mention Carbonell's (1970) statement about SCHOLAR. 'It can be legitimately considered to be in the field of AI, because it answers questions not specifically anticipated, constructed questions on given topics and carried on a mixed-initiative contextual dialogue with human in a rather free and comfortable subset of English'.

Generative systems are ones that produce the teaching material by the computer itself. Such systems need only to be given general teaching strategies and they will produce a tree of possible interaction with infinitely large number of branches (Yazdani, 1987). ITS started as an enterprise attempting to deal with shortcomings of generative systems. It benefited from Artificial Intelligence (AI) which mainly deals with the problem of how best to represent knowledge within an Intelligent System. Various AI techniques have been tried in this areas. Production systems are recognised to be a way of modeling human behaviour. A production system is a method of organising knowledge into facts, rules and inferences (Yazdani, 1987).

In the 1970's the design of CAL systems reached a new level of sophistication. They managed to generate the teaching material itself by the computer. Over the years the algorithms for CAL development have improved the richness of feedback and the degree of individualisation (Yazdani, 1987).

Even though it is hard to distinguish between teaching or learning with the computer, Gibaud (1995) draws limits and identifies Intelligent Tutoring Systems and Learning Environments as two different main lines in this area of research. AI applications in education seem to fall at the two ends of a spectrum, with open ended problem solving environments (Papert,1980; Yazdani, 1984) at one end and individualistic tutoring systems (Sleeman and Brown, 1982) at the other (from Yazdani, 1987).

Individual differences in cognition implies that different learners need different instruction. Based on this fact, Adaptive Education (Glaser, 1976, Wang and Walberg, 1985 from Ohlsson, 1987) has sought to tailor instruction to such characteristics of the student as initial competence, educational goals, learning style and learning rate. Interestingly, it has been observed that different cultures have different styles of learning (Henderson, L., 1993).

Logan and Sachs (1991, in Henderson, L., 1993) distinguish between individualised programmes and personalised approaches. Individualised programmes create learning experiences and materials designed to assist a student acquiring mastery of predetermined content and skills to common standards. Such programmes are confined to catering for quantifiable characteristics such as rates of progress and achievement levels based on group testing or criteria established for a particular task or subject. By comparison, the personalised approach includes both quantitative learner characteristics (e.g. competency) and qualitative learner characteristics (e.g. individual and culturally appropriate ways of learning and motivational strategies).

Instead of adapting global traits such as learning style, the computer tutor can, in principle, be programmed to adapt to the student dynamically, during ongoing instruction, at each moment in time providing the kind of instruction that will be most beneficial to the student at that time. That is, the tutor focuses on the fluctuating cognitive needs of a single learner over time rather than on stable individual differences (Ohlsson, 1987).

Instead of adapting to content-free characteristics such as learning rate, the computer can be programmed to adapt to both the content and the form of instruction to the student's understanding of the subject matter. It is hoped that

the computer can be programmed to generate exactly that question, explanation, example, counter-example, practice problem, illustration activity, or demonstration which will be most helpful to the learner. It is the task of providing dynamic adoption of content and form which is the challenge and the promise of computerised instruction (Ohlsson, 1987)

If an automatic tutor is to cope with the variations and with types of student errors, it must understand what the student is trying to do. An ITS should be able to understand and appropriately respond to the student's action.

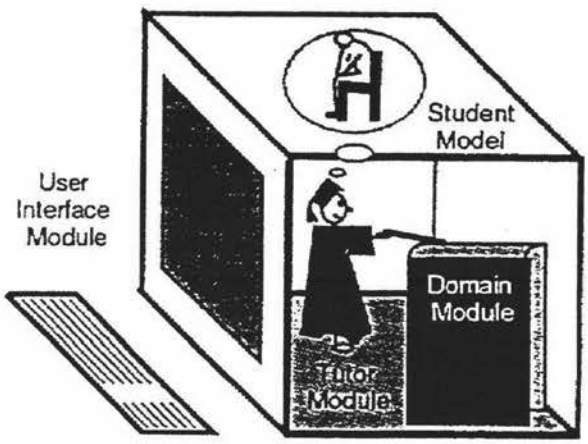


Figure 2.1 ITS Architecture (Perez et al., 1995)

The underlying structure of Advanced Computer Tutoring (ACT) (Anderson, Boyle, Reiser, 1985) principles seems capable of supporting subjects other than Geometry and LISP Programming, in which it has already been applied, as well (Yazdani, 1987). These principles are psychologically plausible with regard to Adaptive Control of Thought theory of Anderson (1982). The four components of ACT's ITS architecture are :

- Domain expert/ Ideal student : this module is capable of actually solving the problems in the domain
- Bug Catalogue : This is an extensive library of common misconceptions and errors in domain.
- Tutoring knowledge : this module contains the strategies used to teach the domain knowledge.
- User Interface : this is the module which administers interaction between the tutor and the student.

ACT tutors seem to incorporate a very dogmatic and authoritarian approach to education. This has some dangers, especially when the student follows a correct path which differs from the path that the system is following. The system diagnoses the error and provides feedback as a hint.

A general purpose shell is needed for the architecture for tool kits for the development of ITSs. With this shell, it would able to build tutoring prototype frameworks where the system could easily be changed from application in one domain to another similar domain (Yazdani, 1987).

2.4.1 Merits of some ITSs

ITSs offer the potential for adaptive education to tailor instruction depending on the student's characteristics. In principle, an ITS can dynamically adapt to the student while tutoring, and provide the best instruction to the student at that moment. In other words, ITS can focus on fluctuating cognitive needs of a student over time, rather than on stable individual differences.

The ITS can adapt to context-dependent characteristic of a learner, to provide the best content and the form of tutoring to the student's understanding of the subject matter.

ANGLE, Pascal Tutor and *Equation Solver,* drill and practice the student in the mathematical concepts that they have learned. Anybody can add new problems to *ANGLE* easily. The system is capable of solving the added problem and tutor the students on it. This provides great flexibility. The student can practice problems in geometry at anytime and at any place, provided they have a computer.

New problems can be added to *Equation Solving Tutor* as well. It tutors the student in solving the new problem. But, it cannot say the answer is right or wrong. In solving simultaneous equations, checking the answer is the best way to assess the progress.

ITSs provide immediate feedback for the student to access their progress. *ANGLE* provides step-by-step feedback and hints on the process of problem solving not

just on the final answer (ANGLE, 1989). If one selects a wrong concept to prove a theorem, it immediately indicates the selected concept would not help in proving the theorem rather than waiting for the student to end up with the wrong answer and then say it is wrong. *Equation Solving Tutor* and *Pascal Tutor* have the same step-by step feedback and appropriate hints.

Hints that *ANGLE* provide depends on the problem solving approach the student chooses to take. This feedback is more specific and very helpful for the student to identify the mistake.

Pascal Tutor has two feedback modes namely immediate feedback and error flagging mode. In both modes the tutor immediately indicates the error, but in the second mode, students have the option of correcting the error immediately or leaving it as it is until the end. This gives more user control over the tutor.

The individualistic tutoring systems does not allow the student to explore the topic of interest. In a way the student is denied the control of the interaction. But, the learning is guided by the tutor. This is especially good for drill and practice types of software such as *ANGLE, Equation Solving Tutor* and *Pascal Tutor*.

Being constructed around a psychological model of knowledge, *Pascal Tutor* has a 'skill meter' to dynamically estimate the student's knowledge acquisition and to provide customised sequence of exercises. *Equation Solving Tutor* has a 'Skillometer' to dynamically trace the student's progress as they solve linear equations.

Computer-based training is proved to speed-up learning and saves a lot of time and money. 20-25 hours of training with *Sherlock II* (troubleshooter for electronic faults in complex testing systems) produced learning equivalent to about four years of on-the-job experience (Feifer, 1983). *ANGLE* has been used in numerous experimental studies in classrooms and proved to double the performance of the student with an 8 hour tutoring. That is very promising. Studies on the use of *Equation Solving Tutor* in classrooms have proven this to be accelerating the student's learning effectively. *Pascal Tutor* has been used in high schools and improved the student's performance by a third.

2.4.2 Problems in ITS

Computer-based systems have a number of advantages. Likewise, they do have some disadvantages. One of the benefits in using computer-based learning systems is saving time or not wasting time compare to traditional instruction. But, in open ended problem solving environments, because of the unstructured exploration much time is consumed as opposed to the purpose of efficiency of computer-based tutoring.

Theoretically, we have different frameworks for producing tutoring systems which appears to be plausible concepts. But, do they really work - exactly as it is proposed - in practice? The concept of *microworld* seems to be a great idea. But, the notion of learning environments are not very clear and nobody knows how to make them (Lawler, 1987).

Most of the ITSs cannot offer the necessary support when the student gets lost the system. In that case, the benefits we achieve by providing individualised instruction may be of less use.

ITSs fall short of being any match for human teachers. In generative systems, there is a mismatch between the program's internal processes and those of the student's cognitive processes. None of the ITS have human-like knowledge of the domain of the subject they are teaching, nor can they answer 'why' or 'how' the task is performed (Yazdani, 1987).

2.5 Interactivity

Interactivity is a recent fashionable trend. *'Interactivity', 'Personalisation', 'Browsing', 'Manipulation', ' Divergent thinking', ' Response contigent displays', 'Discovery learning'* are some issues that have gained great mileage in the computer education revolution (Hooper, 1988).

Interactivity is not necessarily a good thing. It has no meaning if it does not enhance the presentation. Interactivity is mostly the ability to start, stop, review

and preview presentations. In that case, video tapes too have great interactivity. The goodness of interactivity depends on how appropriately it is used.

Video games used to be a big attraction when they were introduced. Most of them involved some level of learning and skill building, but those skills don't help in living our lives and they became less attractive. Interactivity is not by itself experimentally important. Crockford (1988) proposes a system called 'home interactive theatre' which tells a story and invites the user to become involved in it. It provides a safer environment, a sure chance to win with a blend of doubt about the victory. The presentations will be tailored to the user's response. This kind of interactivity may lead to actual learning of necessary skills.

A *gateway lab* (Cowley, Scragg and Baldwin, 1993) is an interactive, computer delivered tutorial, which helps students make the transition from lecture/text book knowledge to active and concrete use of that knowledge. In general, students using gateway labs, seemed to prefer them to text books. It seems to be more effective than text books especially in subjects like computer programming, which need drill and practice (Baldwin, 1996).

The findings of cognitive science that bear on interactivity are Visual learning, Visual memory and active engagement. 'Visual learning' implies people find it easy to learn when they are able to represent the knowledge to themselves in mental images. Visual memory suggests that the memory allocation for visual representation is greater than memory for verbal information. Active engagement claims deep, long term understanding requires active, prolonged engagement and construction of meaning (Anderson, T.G., 1988). Multimedia supports the idea of visual learning and visual memory. Its success depends on making the students actively involved in the learning process.

2.6 Multimedia

With the development of communication technology and computer technology, the world seems to move towards multimedia to enhance this synthesis. Papert (1993, in Hoogeven 1995) speaks of multimedia as the *megachange* in education. On the contrary, Hoogeven, (1995) describes it as an *over optimistic multimedia paradigm.*

Using the traditional teaching methods, students are supposed to understand a particular subject by reading and listening and may be doing some practical work. But in multimedia the students can be active in more ways than just reading and listening (Laurillard, 1994) such as being actively involved in learning by interaction or applying their knowledge to a practical situation.

The Working Group on Interactive Multimedia Pedagogy list the following reasons why multimedia should be used in education. However, only the first two reasons are specifically applicable to multimedia and the rest are applicable to any teaching system in general.

- When industry is promoting multimedia, students and educators expect to use it in classrooms.
- Today's children's experience has been radically different from previous generations. They are more pictorially oriented and this changing nature should be addressed to provide better education.
- Teaching sequence and learning sequence. It used to be linear. But, a study (Adams et al., 1996) proved that around 50% of the students preferred to break away from this linear sequence and use the material as a resource.
- Pace (some students need longer time than other students)
- Most of all, computer-based tutoring can relieve the educational system from the pressure of educating more students with shrinking resources.

The students from our technology oriented generation would prefer to use computers for education because most of the games they play are computer games and they are very familiar with computers. It is likely that they would be motivated by computer-based learning. Keller (1993) proposed a model called ARCS (Attention, Relevance, Confidence, Success) for systematically planning the

development and maintenance of motivation in instructional systems. This model emphasises establishing motivation, establishing interest, relevance of usage of multimedia, building confidence and indicating success.

2.6.1 Interactive Multimedia

Greenberger (1990) predicts that interactive multimedia will totally change the ways in which we think, work and learn. Interactivity is a major feature of multimedia. Knowledge should be chunked into small manageable packets and linked to achieve high integration. This orchestration together with interactivity provide greater learner control. Interactive multimedia products have the capacity to shift the locus of 'ownership' and 'control' in learning (Latchem et al., 1993). Interactive multimedia comes in the form of Interactive Video (IV), or a desk top program offering colour, sound, video, text and graphics; or Virtual Reality (VR).

The real strength of Interactive multimedia depends not only in the advanced technology, but in the courseware which enables the user to access and navigate the information and build, test and apply knowledge in logical and personally meaningful ways (Latchem et al., 1993). From the user's point of view, one of the most significant differences from the multimedia courses of past and present is not the range of modalities available, but their lack of integration and flexibility.

The effectiveness of video is limited by the fact that it is generally linear and passive. Although it could repeat a sequence, interactivity could be a redeeming feature (Singh, 1986).

Interactive video is seen as the most appropriate technology that is closer to the way we think and learn (Daulton, 1986). Interactive video cannot only respond to choices, but it can guide the choice and can cause profitable interaction for learners.

Multimedia interactivity includes two types of interactions : those that give rise to learning (mathemagenic) and those do not (non-mathemagenic). Mathemagenic interactions are computer-mediated activities that give rise to thinking and learning and which either allow or require from the user some overt response or

action to which the computer can subsequently react (Henderson and Patching, 1995). Meaningful learning requires thinking. While interactive multimedia does not directly mediate learning it acts as instructional interventions which mediate the learning activities that subsequently can stimulate thinking. The use of click-drag interactions does elicit a high degree of quality student thinking (Henderson and Patching, 1995). Multimedia alone cannot achieve this stimulation of thinking. Concepts such as challenge should also be included in the system to give rise to mathemagenic learning.

Interactive multimedia as a technology in education offers the possibility of allowing learners to participate in communities of practice and hence move from their status as novices into the realm of the expert. Simulating a world in which the learner moves toward full participation should prove a powerful environment for understanding and also help with transfer of concepts to new contexts (Hedberg and Harper, 1995). If a point can be made by using a dynamic visual sequence rather than a textual description then it is more likely to be quickly grasped and employed in new contexts. The user feels comfortable in the situation and is positively motivated to explore and develop new ways of conceptualising and exploring information landscapes.

If properly applied, interactive multimedia could serve as a medium for students and teachers alike to gain access to a world of appropriately structured, vital, growth-enhancing experiences (Henderson, J. V., 1992).

2.6.2 Advantages of Interactive Multimedia

Multimedia involves multiple sensory modes with the material delivered through the same single environment - the computer. The student's brain receives these inputs simultaneously. It is a substantially different way of engaging the user compare to the traditional teaching methods which normally do not present the student with something that use multiple senses such as animations. Multimedia has the capacity to provide the teachers and students a dramatic new environment for presentations. It will allow a level of interactivity that is not possible in viewing traditional film, audio or print materials.

The level of *multimediality* can be operationalised in terms of the number of information types used. The higher the level of multimediality, the more stimulation of senses, the more arousal, the higher involvement, and the stronger the recognition effects. The *level of congruence* is the degree to which different information types are used redundantly to express the same ideas. The basic assumption is that the high level of congruence is far more effective for recognition and extending the mental reference models necessary for effective information transfer or learning (Hoogeven, 1995).

Colours make multimedia courseware more legible. Pictures make the package attractive and also impact the student. Different learners can acquire knowledge fractions in different orders. Non-linear presentation of multimedia courseware enable the students to learn in their own order.

When teaching today's children, their electronic orientation should be taken into consideration. They are bored by traditional methods used to teach them (especially reading). In a study towards use of pictures in electronic stories, it was found that children enjoyed programs consisting of pictorial representation. It indicates visual clues helped them to retain and retrieve new words better than did nonpictorial representation (Shapira, 1995). *OzSoils* uses a few animation to explain certain concepts. Obviously, those animation and video clips are more useful in retaining what is learnt.

A 20th century philosopher Bronowski says that auditory senses connects us with other living things but that it is vision which we use to understand the physical world. The importance of our visual system is further affirmed by the fact that the information carrying capacity of our visual system is greater than for any other sense.

Computers can be used to enhance the informational value of an image pattern by animation or video or a combination of such elements. These images in turn can be used to enhance the power of our visual thinking, which is a legitimate and distinctive mode of thinking (Friedhoff and Benson, 1983).

2.6.3 Disadvantages of Interactive Multimedia

Simulated situations are not real. Whatever the experience we have with simulated environments, if we cannot apply the knowledge to a real- world situation, it is of no use. diSessa's (1987) statement further confirms this. 'Although it is easy to feel comfortable with the experience in natural surrounding, and frightened of the artificiality of the computers, it should be considered that what makes a good experience, independent of the instrument used to support that experience.

Too many colours cause visual fatigue. Flashy doesn't mean learnable. Text, pictures should be reduced to a minimum to be effective and learnable. *OzSoils* and *Remote Area Servicing* have achieved this to an extent by avoiding too many contrasting colours etc.

In non-computer learning there are not only student-system interactions, but also student-teacher and student-student interactions (Looms, 1993). These interactions should be considered in developing courseware. It should be noted that understanding of the subject vastly increases when shared. And students find it comfortable to discuss their ideas with fellow students. This is not only a problem of multimedia courseware, it is a common problem for computer-based tutoring.

When there are many navigational paths and the courseware is non-linear then the users become overwhelmed by the number of paths and topics available. Such situations tend to lose even the benefits of linear information document. *OzSoils* is more like a non-linear, non-cyclic representation of a text book in soil science. There is no record of the sections that have been covered already. This may result in the repetition of the same sections again and again. It wastes time especially if it has to be finished within a limited time.

In schools we use books in a linear order. They starts from the first page and go through every consecutive page. This trend led to educational software being more like a book, which Schank refers to as *'page turning architecture'*. Even though multimedia software includes video or graphics it does not involve active participation of the student. This is one of the reasons for its failure (Schank, 1994).

Braswell (1994) points out the top five reasons not to use multimedia.

- It is costly.
- Multimedia experience without follow-up may be useless for retaining what was learnt.
- Multimedia is not the only way to present material.
- Students should be aware of the numerous tasks involved in creating the software.
- Information should be able to be adapted by the instructor to meet different needs and levels.

However, one cannot deny the fact that it may have negative effects on students, teachers, institution of education and in society of the future (Adams et al., 1996). Students who use it as a primary vehicle for learning may end up with limited knowledge in the subject. Teacher are worried that it may put them out of job. The educators should make it a point to maintain and improve the standard of education.

2.7 Intelligent Multimedia Systems for teaching and learning

Some media and models of communication are more efficient and effective than others for certain tasks, users and contexts. In daily life when we converse, we use an array of vision, audition and taction. We have the natural ability to manage and exploit multiple input and output media, whereas computers do not. Consequently providing machines with the ability to interpret multimedia input and generate multimedia output would be a valuable facility for many key applications. The marriage of multimedia to ITS offers the potential to create intelligent systems that can instruct and demonstrate, using sound animation, and video.

Use of multiple media should support amplification of human abilities. Empirical studies (Krause, 1995, in Maybury, 1993) indicate that even well accepted multimedia applications can be irritating rather than being attractive and useful. These negative effects could be overcome by using intelligent agents.

The tutoring systems *ANGLE, Equation Solving Tutor, OzSoils* etc. display little or no intelligence that we take for granted in our everyday life. The most complex problem in developing an ITS is what kind of information should be presented at a certain stage of tutoring so that the system could exhibit a little of the intelligence that a human tutor would have. Automating the human tutor's role is to prod, hint, ask leading questions or ask for justifications, and guide the problem solving with just the right amount of intervention and feedback to keep the learner on track.

The 'ITS holy grail' has been software to achieve individualised instruction in all combinations of situations and learning needs. These personalised simulations would be impractical if each adaptation had to be programmed explicitly. The redundant development should be eliminated (Kaplan and Rock, 1995).

Woolf and Hall (1995) proposes an instructional model that uses intelligent simulation, dynamic links which are on-line generation of links based on student behaviour, and multimedia composition and creation. Producing multimedia presentation is more difficult than preparing a text-based document, because it needs the creation of text pictures and organising them.

Generating multimedia courseware can be divided into the processes of content selection, media allocation, media realisation and media coordination. These processes offer a number of challenges like temporal coordination of multiple media, relationship of textual and graphical representation, automatic design of graphics and modality selection.

The generation of multimedia presentation requires

- knowledge about the information to display
- the goal of the procedure
- characteristics of the user
- nature of the media
- degree of automation versus mixed initiative
- decision on saving/not saving the history structure
- plan about if/how animation were connected to represent abstract knowledge

Intelligent systems should be able to provide dynamic support. Dynamic support refers to the process of systematically decreasing amounts of assistance provided to the novices as they progress in expertise and gradually assume parts of the task initially accomplished only by an expert. Dynamic support encourages the progressive development of skill by the learner. It is simply the movement from computer controlled learning through mixed controlled learning to learner controlled learning (Reil et al., 1987).

Motivation and retention are the main concerns in distance education (Carswell and Benyon 1996). Recent studies indicated that multimedia learning alone may not sufficiently increase knowledge acquisition but when a gaming metaphor is employed it significantly increases knowledge acquisition and user enjoyment (Stratfold, 1994 in Carswell and Benyon, 1996).

The capabilities of intelligent multimedia systems include the ability to interpret possibly multimedia questions and automatically design multimedia answers to deal with follow-up questions and make backward references and to post-edit presentations. Further dialogue could be incorporated into multimedia interfaces, a complex model for pedagogue could be developed and multimedia presentations could be tailored to individual users depending on their psychological state, knowledge, abilities, attitudes, preferences, goals and plans.

2.8 Concluding Points

A great educational system is the one that makes learning a compelling experience (Ambron, 1988). Good educational software should be active, taking the idea of learning by doing. Educational systems should be similar to natural learning environments, in which people adopt goals, generate questions and try to find answers (Schank, 1994).

In producing educational software, all that matters is not the quantitative increase of in activity, but the qualitative improvement of the learner's cognitive process. So, the main challenge lies in locating and creating the mass of information that can be stored in the products and utilising this information in ways that promote understanding (Latchem, 1993).

The challenge of ITS lies in its potential for providing instruction which is dynamically adapted to the learner (Ohlsson, 1987). An ITS must have internal representation of the student in order to achieve this. The only way to decide what kind of description is needed is to consider the function of the description within the tutoring system. A tutoring system must have an internal representation of the subject matter in order to adapt teaching to an individual learner, it must be able to generate the particular presentation of the subject which suits the learner best.

Multimedia is not just hype. It has all the elements of a self-fulfilling prophecy, which does not imply that it is ineffective. It has a distinct entertainment value, and if carefully applied sometimes seems to contribute to more effective information and knowledge transfer which is valuable for many situations. Multimedia systems should have qualities such as an appropriate combination of rich communication media, an adequate level of interactivity, a high level of congruence of used information types and an adequate use of reference models to form a basis of a more realistic new multimedia paradigm.

The instructor's job is to motivate, guide, challenge, criticise, brainstorm, manage and lead the students (Schank, 1995). If multimedia is used wrongly, certainly it has negative values added. The real problem is identifying proper technology that is best for people.

There are not only technological problems. A profound problem in the learnability of the common sense knowledge is that the social forces can radically alter the cognitive impact of domains of common sense knowledge. Lawler (1987) shows it with a good example of tax changes on things.

Placing the learner in the middle of an interactive story provides the kind of context and motivation that promises maximum student interest and maximum learning. Resolving the gap between expectations and reality becomes a goal which, when achieved, creates a satisfying and memorable experience (Bielenberg, 1995).

As the National Science Foundation of USA mentions in its report (Woolf and Hall, 1995) in the next century, a personal computer could know from the inflection in your voice - or by a smile or frown - what you want it to do. The psychology of human-computer "interfaces" is just one thorny problem the center faces as researchers attempt to devise better ways to deliver sophisticated electronic presentations that combine text, still images, video, animation, and graphic elements to the desktop computer or the home. The center's research may one day make it possible for high- school students to conduct "virtual experiments" in chemistry or biology on home computers before coming to class.

In conclusion, it is possible to combine interactive multimedia and artificial intelligent to produce effective learning systems. But, it is unlikely that interactive multimedia will prove to be a better medium in general. Like all teaching methods, it will be particularly successful at some aspects of the learning process. It is up to the developers of educational systems to find appropriate ways to optimally synthesise the positive features from both the areas.

Chapter 3

Evaluating Educational Software

3.1 Introduction

Kifer (1995) defines evaluation as a disciplined inquiry to determine the worth of things such as programs, procedures, or objects in educational software. Winship (1989) defines it as the estimate of the value, worth appropriateness and usefulness of a piece of educational software. A thorough evaluation includes activities ranging from sales descriptions of software in magazines to in-depth critical appraisals conducted by academic researchers.

Evaluation is an integral component of ITS. It reflects issues such as the needs for overall quality assurance, curriculum design, learning management, discovery framework design, educational research and learning needs of the students (Fritz, 1994).

There are different types of evaluations that are used for different purposes. Technical evaluation is a critique about the technical aspects of the software such as robustness, reliability under normal conditions, design of its displays, ease of use, procedural instructions and prompts, user control and quality of color, sound, graphics and animation.

Educational evaluation is a judgment about the educational value of a piece of software as a teaching or learning tool with regard to the accuracy of content, its educational value, level of accuracy to suit the audience, achievements of objectives etc. It should be noted that this judgment must be firmly supported by a description of the educational context in which the package is used.

Formative evaluation is intended to provide evidence which can be used in the future development of the software. Formative evaluation has a vital role in the progress of multimedia because it is the means by which we build up our knowledge of what the medium can do (Laurillard, 1994). *Summative evaluation* sets out to make a judgment about the value of the software as an instructional tool in terms of its success or failure to achieve its objectives.

Winship (1989) defines *software review* as the critical appraisal of the educational value of software as a teaching and learning tool, whereas *software evaluation* is the judgment about the value and usefulness of a piece of software by means of

quantitative and qualitative measurement and methodologies such as pre test, post test or observing users. According to this definition our approach could be seen as a *software review* rather than a *software evaluation*. The ultimate aim of both review and evaluation is applying the results for the betterment of the software. The difference is the ways in which the results are obtained. The terms *software review* and *software evaluation* are used interchangeably here, because this approach does not disapprove the necessity for quantitative evaluation.

Evaluation assists in developing courseware and determine how well a lesson accomplishes its goals. Despite the importance of this process, external evaluation or 'validation' has been neglected in the past and only now is gaining the recognition it deserves. Brown and Hicky (1990), describe the validation process and demonstrate that a relatively simple and cost-effective validation unit can be of considerable benefit to any organisation involved in training.

McNaught et al. (1994) states that curriculum design should embrace planning, processes and strategies to ensure that student learning objectives are defined and achieved. The following spiral diagram of curriculum design shows how important evaluation is in achieving the objectives.

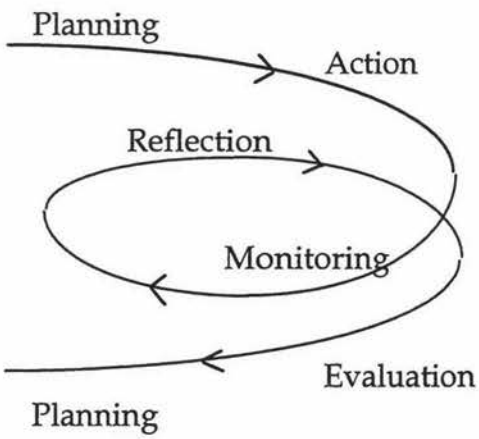


Figure 3.1 Spiral diagram of curriculum design (McNaught et al., 1994)

Maddux et. al. (1992) mention the findings of the Office of Technology Assessment (1988) in their survey on selection of educational software. 'As educational software become more and more sophisticated, product reliability will become an increasingly important factor in school's purchase decisions'. They also quote

Lockard et. al., (1987) who suggest that 'Virtually any gathering of educators with computer interests will at some point prompt discussion of the appalling quality of courseware on the market'.

The above statements clearly shows the importance of evaluation for educational software. The following sections propose an appropriate framework for evaluating educational software and review five educational software packages *ANGLE, OzSoils, Diagnosis, Investigating Lake Illuka,* and *Exploring the Nardoo* using the proposed method.

3.2 Why Evaluate?

Wills (1987 in Maddux et al., 1992) points out that a 'distressingly large number of educational programs are poorly designed, some won't even work when you try to use them on your computer, and many are a waste of time and money'. It is understandable that there is a 'quality lag' in educational software. Because of these inefficient packages, evaluation is necessary to prove that they are worth using.

There are many reasons for the poor quality of educational software. Lockard et al., (1987, cited by Maddux et al., 1992) identify a number of them. There is a lack of theoretical base for most courseware, which will eventually lead to unstructured software. Another big reason is, when it comes to computer-based instruction, technical concerns such as multimedia effects are taken seriously and there is not enough emphasis on its educational value. There is not much concern about the appropriateness of the instruction method to the computer medium.

Another factual reason is the initial scarcity of educational programs and the desperation felt by educators who have consequently failed to demand high-quality programs. Mandell and Mandell (1989) (from Maddux et. al., 1992) cite the pressure educators felt to rush into computer education as one of the reasons. The initial scarcity of high quality educational software resulting from a rush by manufacturers to be first in the market place is also a reason for quality lag. General lack of interest in educational software among computer retailers and

software stores had led to the poor quality of software. And last but not least there is a scarcity of personnel with the expertise to select high-quality courseware.

Guba and Lincoln, 1981 (in Alexander and Hedberg, 1994) describe a few purposes of evaluation. Evaluation help improve the software entity; it constructively criticises the entity; it adapts the entity to a particular context and it certifies the entity in the new context.

Foxon (1989) points out the five main purposes of evaluation in training and development programs identified by Bramley and Newby (1984):

- Feedback: Linking learning outcomes to objectives, and providing a form of quality control
- Control: Using evaluation to make links from training to organisational activities and to consider cost effectiveness
- Research: Determining relationships between learning, training, transfer to the job
- Intervention: Results of the evaluation influence the context in which its occurring
- Power games: Manipulating evaluation data for organisational politics

Evaluation proves that a particular software is worthy of use in a particular area for a particular group of users. For example, in their evaluation process, Atkinson and Burton (1991) demonstrate that results from evaluation using test scores, quizzes, homework assignments and questionnaires indicate that a simulation environment is an appropriate component of MIS instruction. Foxon (1992) demonstrates how post course evaluation is an excellent means of obtaining in-depth data about course participants' learning, skill development and behavior change.

We understand that most of the education software is of poor quality, in terms of learning. On the other hand it should be understood that it is extremely difficult for us to develop educational software which could achieve all its objectives at once. After all, even human tutors cannot achieve that. One plausible solution to this problem is evaluation. Especially for ITS technology, review is critical to avoid repeating the errors in CAI. Evaluation is necessary to estimate the extent to which pedagogically important aspects of human tutor interactions have been encoded

into the ITS. It is necessary to estimate the quality of the program and to demonstrate it is the least expensive instructional alternative.

3.3 Widely used evaluation techniques

There are different methods that are used for evaluating educational software. In general, questionnaires, observation techniques and interviews are used for this purpose. Questionnaires are used to gauge opinions on attitudes to computers, reaction to the program, ease of use, learner control, content, value and use of multiple media. Observation techniques such as videotaping the users are used to observe user and identify if they are *'thinking aloud'*, which means if the users are making any gestures to express their satisfaction or frustration in the process of learning. Interviewing users is one way of knowing what they think about the system. Waiting for users to report on problems is another way of finding out the weaknesses of the system. There are quantitative methods that are used for evaluation. Pre-test, Post-test, comparing treatment groups with previous groups, comparing test scores are few of the quantitative methods.

An evaluation methodology suggested by Foxon (1990 in Foxon, 1992) is as follows. It is a very general frame work which could be used for any software. However, it describes the important basics for an evaluation methodology.

- Purpose: What do I want to know? Why?
- Focus: Where will I source the data?
- Tools: How will I collect the data?
- Feedback: Whom will I tell and how?

Beattie (1994) describes the following methodology for evaluation. It consist of vital questions that are to be answered at different stages of development of educational software. It is more elaborate and more appropriate to educational software compare to Foxon's methodology.

- Identify the task: identify the problem and judge whether current methods are adequate and would the computer system be efficient?
- Make sure that the computer system provides support in achieving learning objectives
- Analyse the best way to present your material and make sure someone will use the software
- Specify what you want to teach
- Determine what it is really about
- Use different research methods such as Triangulation, Pilot testing, Follow up Before/After testing, Case Studies (assessment and test results, Audit trails, Expert criticism, Journals/Diaries, The *'one on one workthrough'* or video, Unobtrusive observation or video, Interviews/ Informal discussions, Student questionnaire/Polls) and Comparative studies
- Include evaluation information in your marketing to justify why should they buy it

The design of the evaluation mentioned by McNaught et al. (1994), for their evaluation of computer-based teaching of veterinary systematic bacteriology and mycology, involved obtaining feedback on the module design, consideration of student attitudes to the changed curriculum, and an analysis of students' ability to solve problems and discuss relevant microbiological issues. This information was obtained through Pilot formative evaluation, On-screen questionnaires given to students after each case study, Informal interviews with students and colleagues throughout the course by the staff teaching the course and an educationalist and an end of course questionnaire.

Shute and Glaser (1990) presents a large-scale qualitative evaluation of *Smithtown*, an ITS designed as a guided discovery world to assist individuals in becoming more systematic and scientific in their discovery of laws for a given domain. In their elaborate study of individual difference and learning they demonstrated that *Smithtown* fared very well compared to traditional instruction in economics.

Even though a variety of evaluation methodologies exist, none of them are general enough to evaluate any educational software. Everyone of them is meant to evaluate a certain package. Section 3.6 tries to suggest a very generalised approach to quantitative evaluation that could be used to evaluate any educational software.

3.4 Issues to be considered in evaluation of educational software

Learning is the main objective of educational software. As Beattie (1994) says the fundamental evaluation question should be 'What's being learnt by the students?'. But most of the evaluations are more concerned about how much the student 'like' the package rather than how well it teaches the students. Generally, the more the student learns from a piece of software the more s/he tends to like it and vice versa, because the success gives him/her confidence. Even though this 'liking' and 'learning' are related to each other, they are separate qualities. While serious consideration is given to the matters related to attractiveness of the package, priority should be given to teaching and learning objectives.

Beattie (1994) claims that a good evaluation needs to establish whether a learning problem is being overcome and then delve deeper in an attempt to investigate the specific learning experiences students are having as a result of using the software.

Interactivity, style, adaptability, validity of strategies, cost-effectiveness, the target user group are some criteria for evaluation. Evaluation needs to focus on the whole curriculum package as well.

Serious design issues like knowledge and information structure, representation and instructional strategies, navigation learning paths and cognition, relationship between novice and expert information structures and information manipulation, presented metaphors and their links to non-linear information structures also need to be considered.

According to Legree and Gills (1991), required experimental conditions in performing a quantitative evaluation are a computer-based teaching system, human tutor and group instruction. These conditions are critical in obtaining meaningful results. Comparing the effectiveness of the modeled human tutor approach to classroom instruction determines the appropriateness of the subject matter and the modeled approach. These conditions are necessary in proving that constructing an ITS is efficient and effective compared to other instruction methodologies.

3.5 Intelligent features

The intelligence of a computer-based tutoring system lies in its ability of instruction to achieve its educational goals for learners in a way that is palatable and acceptable to them.

As Legree and Gills (1991) quote Bloom (1984), students taught by human tutor perform approximately 2 standard deviations better on instructional exams than students taught by group methods. It can be assumed that all the learning objectives can be assumed to be most effectively taught in a one-on-one environment. And human tutoring is not monolithic and can vary in content and approach.

The explicit emphasis on bugs and their detection has been one of the most important contributions of AI to the general theory of cognitive processes (Suppes, 1984).

Major ideological contributions of ITS are

- concept of errors and bugs
- importance of generating qualitative models of learners in order to adapt instruction
- mixed initiative dialogues

Issues recommended by Littman et. al. (1985) to be included in an ITS are

- How critical bugs are,
- What category the bugs fall into,
- What caused the bugs
- What tutorial goals are appropriate for tutoring bugs,
- What tutorial interventions would achieve the tutorial goals.

These issues can be added in a more in-depth evaluation.

By informational feedback, the student has the opportunity to try the exercises again after correction (Cohen, 1985 in Lanza and Roselli 1989). In this way, she can

carry out free and at-will training whenever she desires, without interrupting the execution of the lesson and while continuing to have the assistance of the system. Feedback is a very important issue in evaluation.

Use of interactions as the training strategy reflects the need (Francis and Ottenstein, 1985; Carroll and Kay, 1988 in Lanza and Roselli, 1989) to provide effective learning by means of an instructional system which feeds the student's interest and is able to satisfy the needs of more motivated and independent students.

3.6 Proposed appropriate evaluation technique

There are different evaluation models that focus on different issues. Each has its own strengths and weaknesses and may be more applicable to one setting than another.

Kifer (1995) mentions four basic models of evaluation. A *Traditional model,* proposed by Tyler (1949) emphasises consistency between goals, experiences and outcome. In this model, evaluation is essentially the process of determining to what extent the educational objectives are being realised. The objectives aimed at are to produce certain desirable changes in the behavior patterns of the students. Thus, evaluation is the process of determining the degree to which these changes in behavior are actually taking place. The *Design-oriented* model proposed by Stufflebeam (1983 in Kifer, 1995) emphasises collecting information from a variety of sources to provide a basis for making better decisions. The *Case Study or Ethnographic model,* proposed by Stake (1967), emphasises an understanding of what is being evaluated. The *Goal Free and Integrative model* suggested by Scriven (1967,1983) emphasises determining all of the effects or outcomes and deciding whether the effects, either intended or not, are desirable.

Kifer (1995) gives the following figure for viewing evaluations. An evaluation may contain aspects of each of these dimensions.

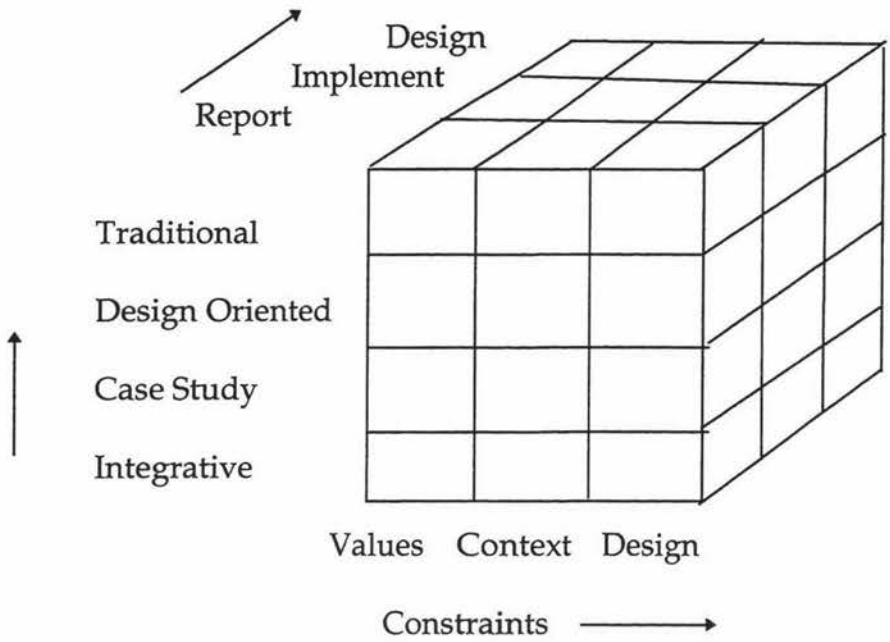


Figure 3.2 A heuristic for viewing evaluation (Kifer, 1995)

Alexander and Hedberg (1994) classify the models as Objectives-based (Tyler, 1949), Decision-based (Stufflebaum, 1967), Value-based (Scriven and Stake, 1983) and Naturalistic (Lincoln and Guba, 1981) respectively. They recognise that the most naturalistic approach for evaluation is a series of issues in the form of a checklist.

Because of the simplicity and appropriateness of the checklist of issues, a naturalistic approach is adapted to the proposed framework and the issues. The issues in the check list are mostly objective-based issues. It also includes a number of intelligent features that enhance the learnability, and presentation way on issues that increases the retainment of facts, and design issues that makes a more structured piece of software. Depending on the purpose and depth of evaluation, the effective dimensions of interactive learning systems (Reeves, 1992) can be added to the check list. In that case, an analog scale should be used instead of digital scaling.

Dimension	Range		Explanation
	From	To	
Epistemology	Constructivism	Objectivism	Appropriateness of epistemology to the tutorial
Pedagogical Philosophy	Instructivist	Constructivist	Appropriateness of pedagogical philosophy to the tutorial
Underlying psychology	Behavioural	Cognitive	Appropriateness of underlying psychology to the tutorial
Goal-orientation	Sharply-focused	Unfocused	Appropriateness of goal-orientation to the type of tutorial and domain
Instructional sequencing	Reductionist	Constructivist	Appropriateness of instructional sequencing for optimum learning
Experiential Validity	Abstract	Concrete	Appropriateness of experiential validity to the domain
Role of Instructor	Teacher-proof material	Equalitarian facilitator	Appropriateness of role of instructor to the type of tutorial and domain
Value of errors	Error-less learning	Learning from experience	Appropriateness of value of errors to the domain
Motivation	Extrinsic	Intrinsic	Appropriateness of motivation for optimum learning
Structure	High	Low	Appropriateness of structure to the type of tutorial and domain
Accommodation of Individual differences	Non-existent	Multi-faceted	Appropriate individualisation
Learner Control	Non-existent	Unrestricted	Appropriateness of learner control to the type of tutorial, domain and target population
User Activity	Mathemagenic	Generative	Appropriateness of user activity for optimum learning
Cooperative learning	Unsupported	Integral	Appropriateness of cooperative support for optimum learning

Figure 3.3 Dimensions of Interactive Learning Systems(Reeves, 1992)

Winship (1989) lists the most common evaluation criteria identified by the Educational Software Evaluation Consortium. These were selected from a pool of 320 criteria by the consortium members and ranked as follows.

Correctness of content presentation: the program is free from content, informational, computational, grammatical and syntactical errors.

Content presentation: the pedagogical content is presented in a clear concise, logical and manageable fashion and in sufficient depth of instruction and/or practice so that learning will take place.

Use of technology: this is an appropriate use of computer technology, such that the program take full advantage of the computer's capabilities and provide students with a learning experience.

Integration into classroom use: the program can be effectively and easily integrated into classroom use; the software lends itself to use within a classroom time frame; effective and appropriate teacher support materials are available; the program can easily used by the teacher.

Ease of use: the program is user-friendly

Curriculum congruence: the content directly supports the curriculum.

Interaction: interaction is effectively achieved for the target audience; there is a sufficient amount and a sufficiently high quality of it to promote learning

Content sequence/ level: there are multiple levels of difficulty with appropriate incremental steps among the levels; the development sequence and the difficulty of the levels are appropriate to the target audience.

Reliability: the program is free from programming and technical errors

User control of program: the user can control the rate, amount and sequence of presentation

Feedback (general): the program correctly assesses the student input and provides appropriate and effective feedback messages

Objectives: the objectives are clearly stated and met

Motivation: the program is motivational

Branching: there are branches to provide individualised instruction according to each student's needs.

Negative feedback / help: corrective feedback messages or help screens are provided as needed
Content modification: the content can be modified by the teacher

Content bias: the content is free from - race, sex, cultural, ethnic, stereotyping, violence - bias.

Teacher documentation: the documentation is comprehensive, easy to understand and effective.

User support material: there are user support materials which are appropriate and effective.

Color, sound, graphic, animation: when present, these features are used effectively to enhance the program

Screen displays: screen displays are effectively and appropriately formatted

Management system: there is a management system which provides an effective means for record keeping and/or assignment control

Other than the above mentioned characters, the following issues also could be considered while evaluating educational software.

Concept of errors and bugs: different depth of errors are treated differently and appropriately to motivate students.

Type of tutorial: the tutorial is test or drill and practice or simulation and if that suits the domain and promotes learning.

Cost effectiveness: the system is worth using and less expensive.

Target population: the type of tutorial and content suits the target students

Figure 3.4 shows one possible way of representing the checklist. It has characteristics and a scale ranging between 1 and 6 meaning Completely Unsatisfactory, Unsatisfactory, Satisfactory, Good, Excellent and Not Available (to

point out the issues that do not exist). A hierarchical diagram of these characteristics are shown in Appendix B.

Figure 3.5 shows another form of representation. It is similar to the previous model. But, it has an advantage of a comment column that provides a means for justifying the scale given. This model has been used in evaluating the packages.

Characteristic	C.U	U	S	G	E	N.A
Correctness of content presentation					*0	
Content Presentation				0	*	
Use of technology					*0	
Integration into classroom				0	*	
Ease of use					*0	
Curriculum congruence				0	*	
Interaction					*0	
Content sequence / level				0	*	
Reliability					*0	
User control of the program				0	*	
Feed back (general)					*0	
Objectives				0	*	
Motivation					*0	
Branching				0	*	
Negative feedback/ Help					*0	
Management system				0	*	
Screen displays					*0	
Color, sound, Graphics, Animation				0	*	
User support material					*0	
Teacher documentation				0	*	
Content Bias					*0	
Content modification				0	*	
Concept of errors and bugs					*0	
Mixed initiative dialogue				0	*	

Figure 3.4 Checklist for Evaluation I

Key: *.....* Most practical educational software

0------0 Ideal educational software

Ideal educational software would have 'excellent' rating for all characteristics. However, in reality most practical software would have these characteristics ranging from satisfactory to excellent.

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*0		Free from informational errors
Content Presentation				*	0		Clear presentation
Use of technology					*0		Appropriate
Integration into classroom				*	0		Effective integration
Ease of use					*0		User friendly
Curriculum congruence				*	0		Directly applies to curriculum
Interaction					*0		Effectively achieved
Content sequence / level				*	0		Multiple levels of difficulty
Reliability					*0		Free of technical errors
User control of the program				*	0		User can control the sequence etc.
Feed back (general)					*0		Correctly assesses student
Objectives				*	0		Clearly stated and met
Motivation					*0		Motivational
Individualised instruction				*	0		Personalised instruction
Negative feedback/ Help					*0		Corrective feedback
Management system				*	0		Means for record keeping
Screen displays					*0		Appropriately formatted
Color, sound, Graphics, Animation				*	0		Effective enhancement of presentation
User support material					*0		Appropriate material
Teacher documentation				*	0		Comprehensive documentation
Content Bias					*0		Free from ethnic, cultural etc. Bias
Content modification				*	0		Modifiable by teacher
Concept of errors and bugs					*0		Differentiate between errors and bugs
Mixed initiative dialogue				*	0		Initiation by user and system

Figure 3.5 Checklist for Evaluation II

3.7 Evaluation of some educational packages

In the following sections we evaluate a few examples of educational software. In the diagrams that describe the structure of each piece of software, the convention used is as follows. Ovals with thin arrow lines represent multimedia elements used for presentation, thick arrow lines show the presentation order, rectangles represent the system and user activities.

There may be some inconsistencies noticed in the comment columns of the checklists. While two characteristics are given same rating, but the comment may be considerably different. This is because of the consideration given to the achievability of the characteristic.

3.7.1 Angle 3.3j

ANGLE 3.3j geometry tutor was developed to test the ACT* theory of cognition (Anderson, Boyle, and Reiser, 1985). The four Main features of ACT* are as follows.

- Student is represented as a production set
- Problem-solving behavior involve goals
- Minimising the load on working memory
- Knowledge compilation

ANGLE 3.3j is a computer-based tutor for geometry theorem proving developed at Carnegie-Mellon University, Pittsburgh, USA. It is a 'Cognitive tutor' based on research in Cognitive Psychology and Artificial Intelligence.

Unlike most of the computer-based tutors, *ANGLE* provides learner-centered, step-by-step feedback and hints on the process of problem solving, not just on the final answer. It records students actions and analyses student's progress and skill development. New problems could be added easily and the system can tutor the students on it.

ANGLE was developed to tutor high school students in one of the most difficult subject area, theorem proving in geometry (Williams, 1980 in McKendree, 1989).

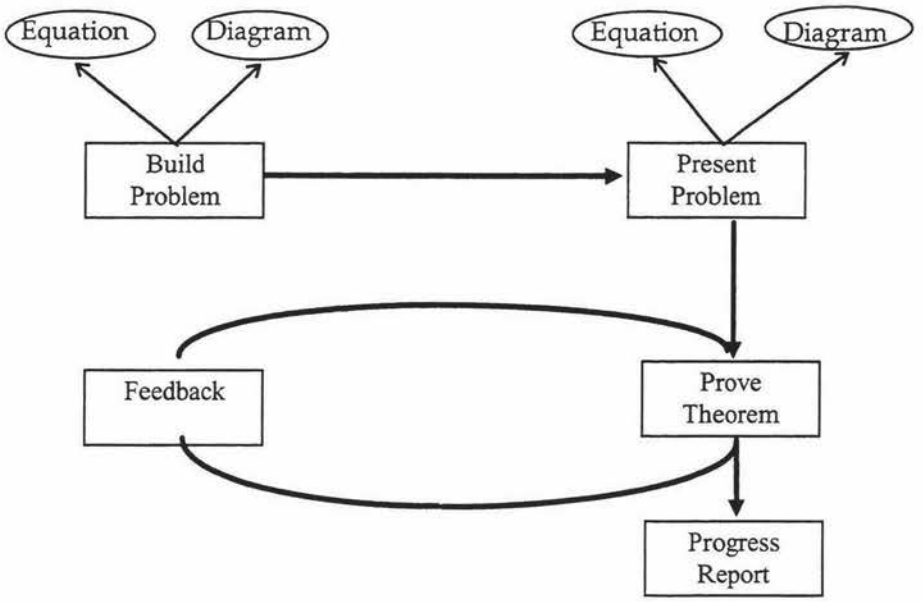


Figure 3.6 General structure of Angle

New geometry theorem proving problems can be built in terms of given equations and diagrams, using the available tools. When the student tries the problem it is presented in the same terms it was built and s/he is asked to prove the theorem. During the process of theorem proving, the student is given an appropriate feedback message if they try to do something that is irrelevant to the proof. In the end the student is given a report on how well his/her progress in geometry theorem proving.

Subjectively, results of evaluation of *ANGLE* were not very positive. The interface of the software was not great compare to doing the proof on paper. Paper-oriented people may find the interface unfriendly. The drawing and rotating facilities are limited. For a subject like geometry, drawing is particularly useful especially if a person is attempting the problems and not represent it in the way *ANGLE* represents it.

People who are familiar with the subject may dislike it, most probably because of the interface. On the other hand they don't have to try the package anyway. However, commenting on it qualitatively, it is a very promising tutor. It has the potential to tutor a student who is more electronically oriented and has little knowledge of geometry theorem proving.

Evaluation form for Angle

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*		Error free
Content Presentation				*			Fairly understandable
Use of technology			*				Effective use
Integration into classroom				*			Mostly fit -used in schools
Ease of use			*				Not very easy to use
Curriculum congruence				*			Fair
Interaction				*			It is an ILS
Content sequence				*			Three levels of difficulty
Reliability			*				Works OK
User control of the program				*			User can select the level of difficulty etc.
Feed back (general)				*			Very good feedback on the bugs
Objectives			*				Probably met
Motivation				*			Fair
Branching					*		Comparatively personalised
Negative feedback/ Help				*			Good
Management system					*		Keeps student record
Screen displays				*			Fair
Color, sound, Graphics, Animation						*	
User support material				*			User manual
Teacher documentation				*			Clear, not specific
Content Bias						*	No bias
Content modification				*			Can add new problems
Concept of errors and bugs					*		Differentiate
Adaptation to learners				*			Provide individual help
Mixed initiative dialogue				*			Provided

Figure 3.7 Checklist for ANGLE

3.7.2 The Equation Solving Tutor

The Equation Solving Tutor was developed by Steven Ritter using the Tutor Development Kit developed by Ray Pelletier. It presents linear equations and monitors the solution step-by-step as the student work on it.

It provides individualised help. Feedback can be customised to be general or specific. The first time the student asks for hint on a particular problem s/he is given a very general hint. The next time it provides a more specific goal to the problem and the third time the student asks for help they are given the specific action to perform for that particular linear equation.

It also has a *skillometer,* which presents the assessment of the student's skill development. For this purpose *The Equation Solving Tutor* has a knowledge tracing facility to track student's progress throughout the problem solving session. It increases the student's estimated skill if s/he solves the problem without any help.

New linear equation solving problems can be built. Since the tutor contains rules which enable it to solve any linear equation, it can instruct the student in solving any equation. Also it is flexible enough to accept all solutions that take the student on a path to the correct solution, even though it recommends a particular solution path (*The Equation Solving Tutor,* 1993). Figure 3.8 depicts the structure of *The Equation Solving Tutor.*

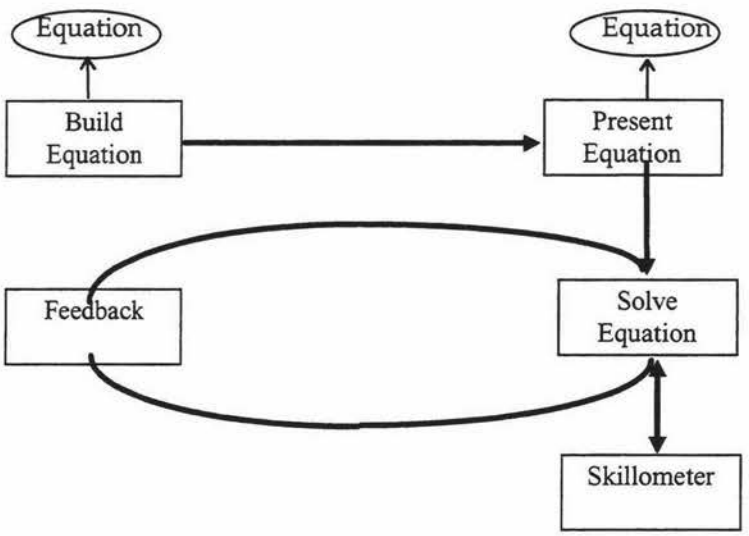


Figure 3.8 General structure of Equation Solving Tutor

Evaluation form for The Equation Solving Tutor

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*		Error free
Content Presentation				*			Fairly understandable
Use of technology			*				Effective use
Integration into classroom				*			Mostly fit. Used in schools
Ease of use				*			Relatively easy
Curriculum congruence				*			Fair
Interaction				*			Its an ILS
Content sequence				*			Arranged as groups of problems
Reliability			*				Works OK
User control of the program				*			User can select the group of problem, feedback type
Feed back (general)				*			Very good feedback on the bugs. Also different levels.
Objectives			*				Probably met
Motivation				*			Fair
Branching					*		Fair amount of personalisation
Negative feedback/ Help				*			Good
Management system				*			Keeps student record
Screen displays				*			Fair
Color, sound, Graphics, Animation						*	
User support material				*			User manual
Teacher documentation			*				General instructions
Content Bias						*	No bias
Content modification				*			Can add new problems
Concept of errors and bugs					*		Differentiate
Adaptation to learners				*			Provide individual help
Mixed initiative dialogue				*			Provided

Figure 3.9 Checklist for The Equation Solving Tutor

3.7.3 OzSoils

OzSoils is an interactive multimedia educational package for teaching soil science. It uses a windows environment with mouse clicks to turn pages. It is like a book in most ways. It has a non-linear menu structure which gives a lot of user control.

OzSoils present the material in an attractive way using text, video, audio and animations. The student can click on the menu to navigate through the material. Once the student is finished with a section, follow-up questions are displayed. These are multiple choice questions and the student has to click on the correct answer. Then the system display number of attempts made to answer the questions. Figure 3.10 represent how *Ozsoils* works.

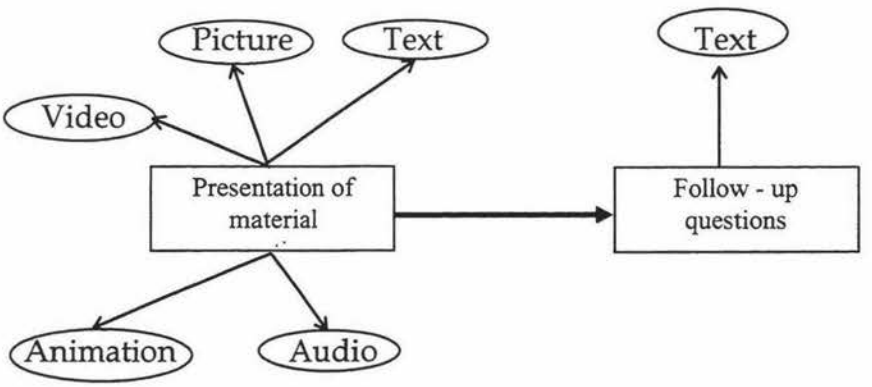


Figure 3.10 General structure of OzSoils

Evaluation form for *OzSoils*

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*		Error free
Content Presentation				*			Very catchy
Use of technology					*		Appropriate
Integration into classroom			*				Used at least in one university
Ease of use					*		Very easy to use
Curriculum congruence				*			Can directly use
Interaction					*		Non-linear presentation
Content sequence						*	No sequence. Too many choices. May be confusing for some.
Reliability				*			
User control of the program				*			Almost complete user control.
Feed back (general)		*					Not enough to assess
Objectives			*				Depend on user
Motivation					*		Catchy
Branching					*		Can branch anywhere
Negative feedback/ Help						*	
Management system	*						No student record
Screen displays					*		Appropriate
Color, sound, Graphics, Animation					*		Appropriate
User support material				*			None, but not needed
Teacher documentation						*	
Content Bias		*					Biased to Australian soils
Content modification	*						No real need to be modified within Australia
Concept of errors and bugs	*						
Adaptation to learners	*						
Mixed initiative dialogue				*			

Figure 3.11 Checklist for OzSoils

3.7.4 Diagnosis for crop protection

Diagnosis for Crop Protection is a multimedia software package that aids in teaching the process of diagnosing crop problems. The user is presented with a problem scenario and must interrogate the program to diagnose the situation. Information is presented in the form of a text window, and optional graphics and video. By looking at different plant parts, the growing environment, and by asking a variety of questions, the user gradually accumulates knowledge about the problem and makes a diagnosis.

Feedback consisting of the correct answer, the key points that should have been observed, and the points that an experienced consultant would typically make to diagnose the problem are given once a final diagnosis is entered by the student. A points score and accumulated costs are also shown. This feedback can be saved to disk for later assessment by teachers.

A Scenario Builder complements *Diagnosis*, easily allowing the creation of many different scenarios for the same crop, and for a variety of crops. The user decides on a crop problem, and enters textual information and selects relevant images, sound segments and/or video clips.

Diagnosis has a comprehensive feedback system. If permitted by the scenario creator, the user may view the solution, recommended action, general debriefing comments, total cost and points accrued. Other feedback notes are also supplied.

Figure 3.12 shows the basic functioning of *Diagnosis*. For a detailed description see Appendix C.

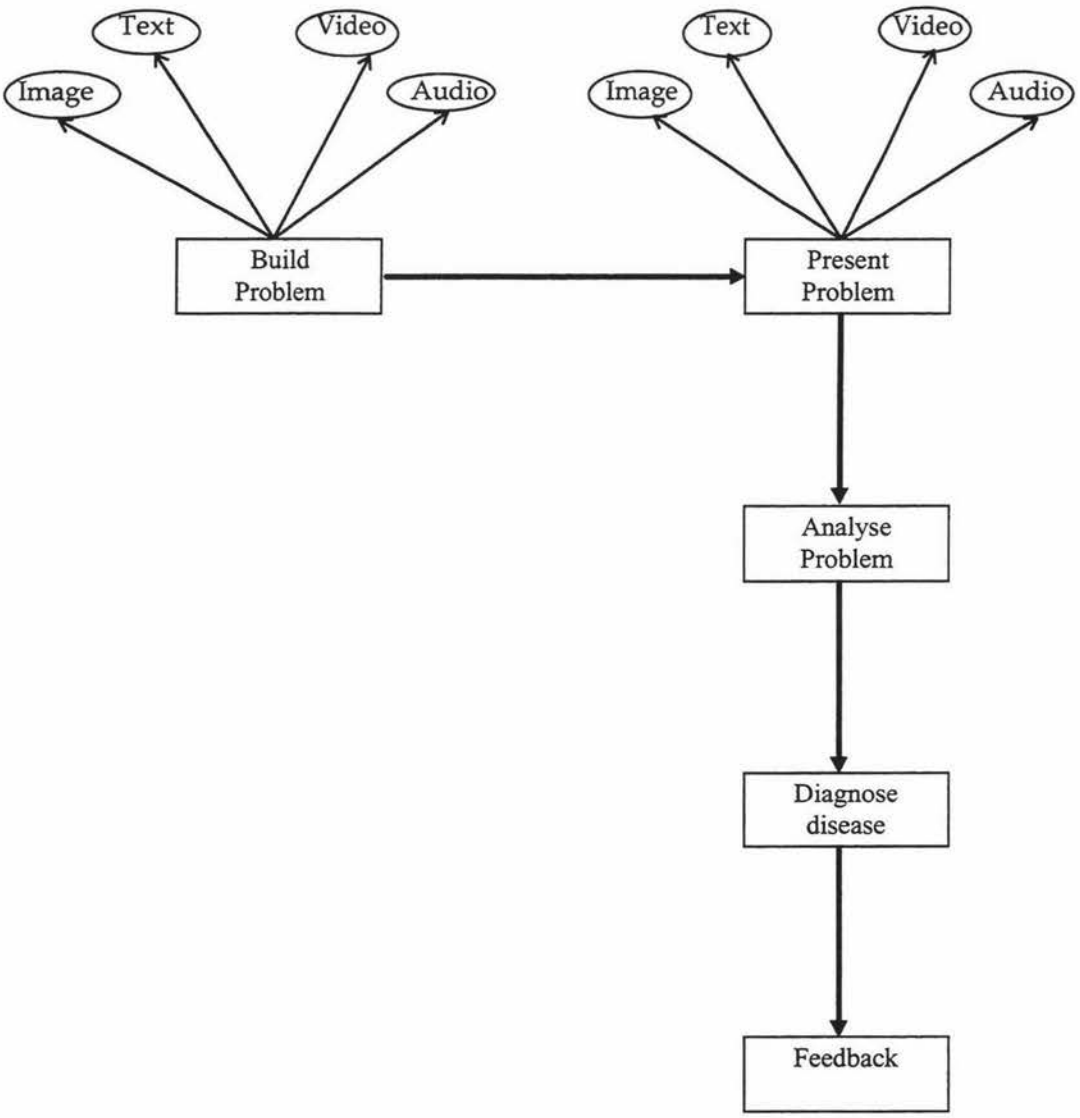


Figure 3.12 General structure of Diagnosis

Evaluation form for Diagnosis

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*		Error free information
Content Presentation				*			Very interesting
Use of technology					*		Appropriate
Integration into classroom			*				Used at least at Massey
Ease of use				*			Easy to use
Curriculum congruence				*			Directly fits
Interaction					*		Good
Content sequence				*			Indirect - depend on problem
Reliability				*			
User control of the program				*			
Feed back (general)		*					Poor
Objectives			*				Hopefully met
Motivation					*		Gaming metaphor
Branching						*	
Negative feedback/ Help						*	
Management system				*			Keeps record
Screen displays				*			Appropriate
Color, sound, Graphics, Animation				*			Appropriate
User support material			*				Readme file
Teacher documentation				*			Clear
Content Bias				*			Not biased
Content modification				*			Teacher can add any problem
Concept of errors and bugs						*	
Adaptation to learners						*	
Mixed initiative dialogue				*			

Figure 3.13 Checklist for Diagnosis

3.7.5 Investigating Lake Illuka

Investigating Lake Iluka was developed by the Interactive Multimedia Unit, Faculty of Education, University of Wollongong, Australia. It is a simulation of an imaginary lake environment designed to support the teaching of ecology. Even though the lake is imaginary, the skills that the students develop as they investigate lake Illuka are not at all fictional. It is an ideal preparation for a future field study trip (Investigating Lake Illuka, 1993).

There are four ecosystems - Urban, Mangroves, Estuaries and Open Lake. The student has to select any one of these ecosystems to investigate. The ecosystem could be explored simply by clicking at various areas of the system. The student can learn about plants and animals that live in the ecosystem. For more information the student can visit the media room. There are newspaper clippings, radio discussions, video clips and a reference book in the media room to be read/watched.

There are three types of tools to measure various characteristics of an ecosystem. Physical tools available are a thermometer for testing air, water or soil temperature, an anemometer for wind speed and a hygrometer for air humidity. Chemical tools may be used to test nitrogen, phosphorus, oxygen and pH. The biological tools provide information about water quality with tests for suspended solids, pesticides and oil as well as nutrients. It also has a user notebook for collecting information directly from the package.

The potential of interactive multimedia learning has been harnessed in this new approach to the design of learning materials. *Investigating Lake Iluka* is designed to run on any Macintosh computer capable of displaying colour.

Figure 3.14 shows the basic structure of *Investigating Lake Illuka*. All the activities use text, audio, video or image elements. For simplicity of the diagram these are omitted.

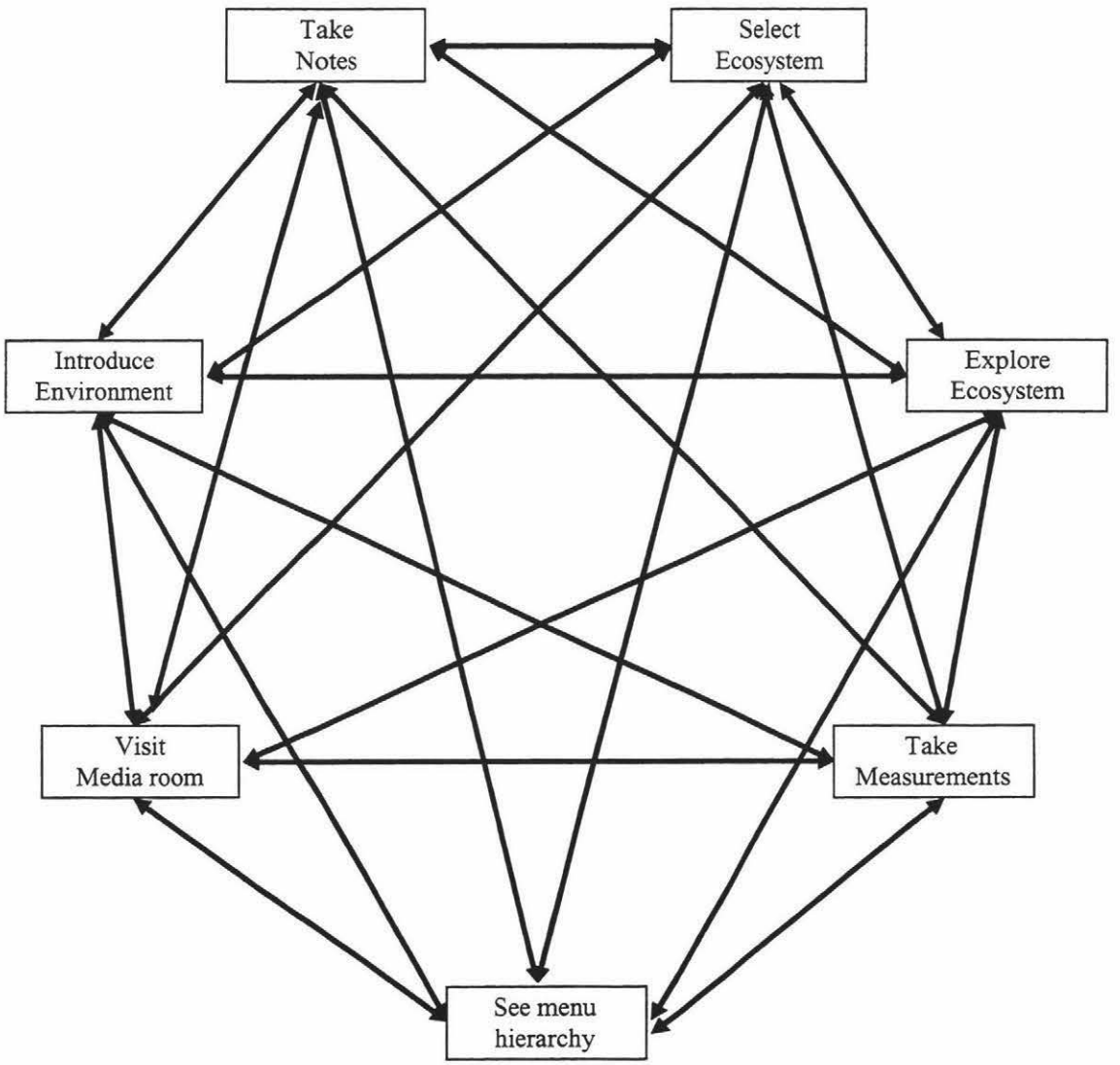


Figure 3.14 General structure of Investigating Lake Illuka

Evaluation form for *Investigating Lake Illuka*

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*		Error free information
Content Presentation				*			Very interesting
Use of technology					*		Appropriate
Integration into classroom			*				Used at least in a few high schools
Ease of use				*			Easy to use
Curriculum congruence				*			Directly fits
Interaction					*		Good
Content sequence				*			Indirect - depend on student
Reliability				*			
User control of the program				*			
Feed back (general)	*						Poor
Objectives			*				Hopefully met
Motivation				*			Depend on the student
Branching						*	
Negative feedback/ Help						*	
Management system				*			Keeps record
Screen displays				*			Appropriate
Color, sound, Graphics, Animation				*			Appropriate
User support material			*				Hand book
Teacher documentation			*				OK
Content Bias	*						Based on an imaginary lake environment
Content modification						*	Teacher can add any problem
Concept of errors and bugs						*	
Adaptation to learners						*	
Mixed initiative dialogue				*			

Figure 3.15 Checklist for Investigating Lake Illuka

3.7.6 Exploring the Nardoo

Exploring the Nardoo was developed by the Interactive Multimedia Unit, Faculty of Education, University of Wollongong in collaboration with the NSW Department of Land and Water Conservation, Australia.

It simulates a changing island river catchment, from its prestine state late last century to a development affected environment today. The catchment contains the Nardoo, a river passing through four regions namely Upper catchment, Upper middle catchment, Lower middle catchment and Lower catchment. The time zones are The prestine state, Early habitation, Later habitation and Present day.

Exploring the Nardoo has been designed to complement the study of ecology and the physical geography at the senior school level. It has a clear curriculum statement, which list the objectives in developing knowledge, skills and values in the areas of Biology and Geography. In addition to this, it has a well documented teacher support material, which is quite useful in introducing the package to the class.
It also list the student requirements. Experience in self directed study, investigation processes, problem solving and small group work. Besides the student should have basic computer literacy in word processing to use the notebook.

The main feature of *Exploring the Nardoo* is Personal Digital Assistance (PDA). It is a multipurpose device which allows the student to navigate the river, take measurements, view and collect news items, make notes and get help. PDA has five separate modules called Navigator, Viewer, Notes, Tools and Help.

Other than PDA, there is a Water research center (WRC), which holds most of the information the students need to complete their investigation. WRC has a computer catalogue, a Plant and Animal Book, Film, television and radio clipboards, River investigation notice board, a Filling cabinet, a simulator and a clipboard folder with news paper clips. All of these contain information on the River Nardoo in different media. Students are aided in their writing and editing of their project report by the presentation guide, text tablet and an editing tool.

Figure 3.16 shows the basic structure of *Investigating Lake Illuka*. All the activities use text, audio, video or image elements. For simplicity of the diagram these are omitted.

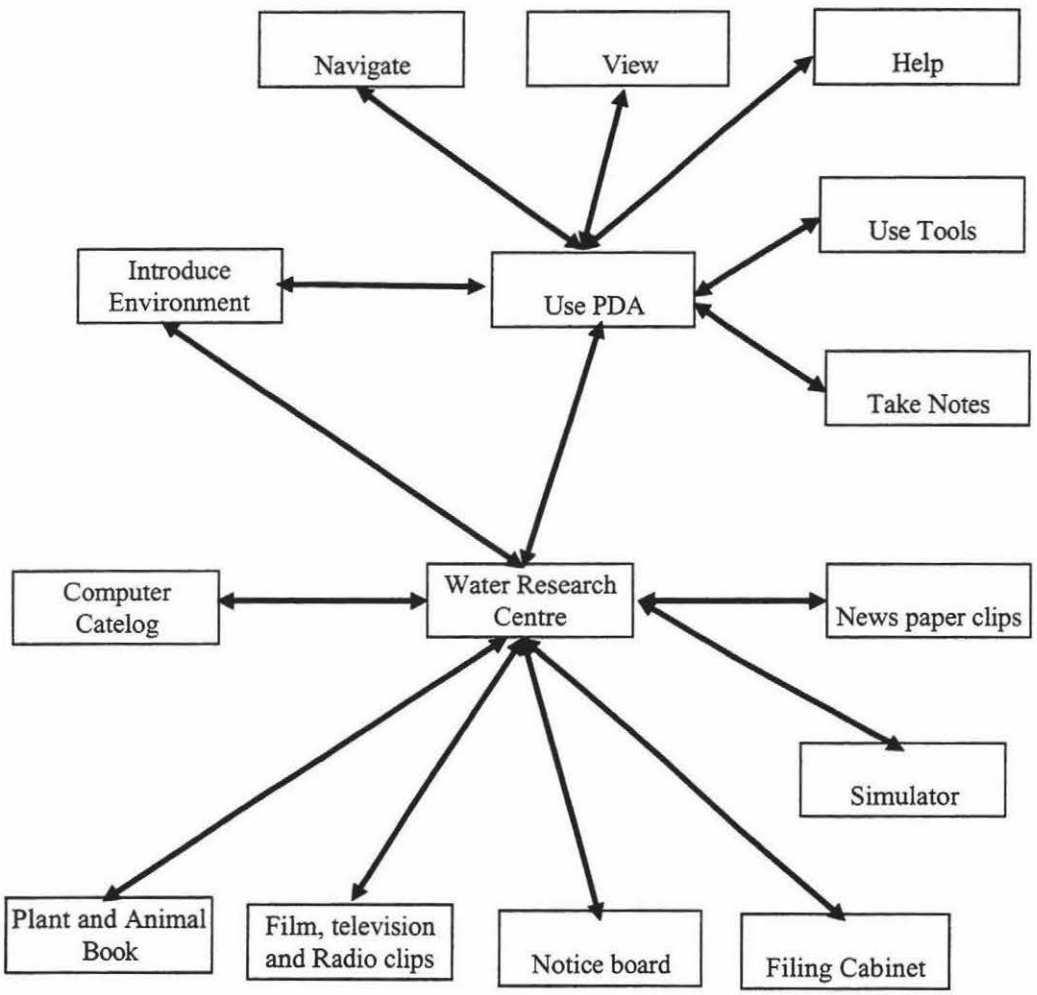


Figure 3.16 General structure of Exploring the Nardoo

Evaluation form for *Exploring the Nardoo*

Characteristic	S C A L E						Comments
	1	2	3	4	5	6	
Correctness of content presentation					*		Imaginary information
Content Presentation				*			Very interesting
Use of technology					*		Appropriate
Integration into classroom			*				Used in high schools
Ease of use				*			Easy to use
Curriculum congruence				*			Directly fits
Interaction					*		Good
Content sequence				*			Indirect - depend on the student
Reliability				*			
User control of the program				*			
Feed back (general)		*					Poor
Objectives			*				Hopefully met
Motivation					*		Gaming metaphor
Branching						*	
Negative feedback/ Help						*	
Management system				*			Saves student notes
Screen displays				*			Appropriate
Color, sound, Graphics, Animation				*			Appropriate
User support material			*				Readme file
Teacher documentation				*			Clear
Content Bias	*						Biased
Content modification						*	
Concept of errors and bugs						*	
Adaptation to learners						*	
Mixed initiative dialogue				*			

Figure 3.17 Checklist for Exploring the Nardoo

3.8 Reflection on evaluation

Reviews should make clear the educational viewpoint of the reviewer. Here is my view. Being on the constructive side of epistemology, I believe that it is not really worth knowing something, that cannot be retrieved and applied at the right time and at the right situation. In other words, what we call intelligent behavior is being able to survive in situation using our knowledge directly or indirectly. In the light of this opinion, learning is not only a process of knowing things, but more importantly it is a process of properly organising the things that we know, in order to retrieve them quickly when needed.

So, basically learning is:

- knowing some facts
- being able to retain the facts for a longer period of time
- being able to identify the appropriate piece of information applicable to the situation
- being able to retrieve the fact from the memory
- being able to correctly modify it, if necessary
- being able to successfully apply it to the situation

Among the above listed steps, an ITS should be able to fully promote the first two, and to let the learner develop the skills in the rest. ITSs help learners in achieving all these faster by providing them the right situation to play with. It should be able to increase retention of the learnt things in a decreased time period.

In order for the learner to learn some facts, ITSs should present the fact in a learnable way. Visualisation of a concept speeds up the retrieval of the same at a later time. In *OzSoils*, the formation of NaCl is animated. Personally, it would be very effective in retaining that piece of information. However, the more original (innovative) the perception is the more stronger the view will be.

OzSoils is more or less like a book with attractive pictures and animation. It has non linear menus. But the question is do we really need that for this package? Are we making use of the non linearity? Or it is more learnable if used linearly? Such issues should be seriously treated.

The most important fact is, learning depends on the learner's qualities. Only thing we could do is motivating the learner and giving them the control over learning. Making use of it is totally depend on her. It is almost impractical to develop a system that could adapt to every user as each has unique personal qualities.

Section 3.7 reveal that the packages evaluated have their own merits and demerits. The major advantage of the cognitive tutors (*ANGLE, The Equation Solving Tutor*) is knowledge tracing and hence highly individualised help. It is mostly because the domain is mathematics, which is easier to be represented as rules. The not so fantastic feature is their interface. With an improved, friendly interface they would be better learning tools.

The rest of the packages are in non-procedural domains. So, they do not have knowledge tracing facilities or personalised instruction. *OzSoils, Diagnosis, Investigating Lake Illuka* and *Exploring the Nardoo,* all have a big plus point which is their use of multimedia, not because it is colorful, but because it is appropriate.

OzSoils has a attractive interface. But it is more like a book with standard chapters. *Investigating Lake Illuka* and *Exploring the Nardoo* are also interesting software, but they are not flexible enough to present different problems. *Diagnosis* is the only package, which appropriately uses multimedia and has an authoring system to tailor new problems.

Figure 3.13 shows *Diagnosis'* advantages and weaknesses. It has a distinct combination of use of multimedia and authoring system for building problems. Even though it has zero degree of personalisation it has room to accommodate this facility. The complexity of its domain may make these changes difficult. But, it could be understood that is has the potential to be a very good learning system, if provided with some intelligent features.

Chapter 4

Designing Intelligent Multimedia Tutoring Systems

4.1 Introduction

We are already in the age of multimedia. Focus magazine wrote in early 1997 that the impact of multimedia - the simultaneous transmission of sound, images and text - reaches beyond the sphere of communications. It is profoundly transforming the very notion of exchanges, whether economic or cultural, and spewing a new social philosophy (Focus, April 1997).

Interactive multimedia offers unparalleled potential in addressing problems of time, space and distance in instruction and learning (McNamara,1994). Traditional methods of delivering training and education have had limited success in meeting the fast-paced and high-quality demands of learning environments today. Multimedia can change the look and feel of learning by providing an opportunity to reach people with different learning styles, different skills, and in different geographical areas; it offers the potential to reduce the learning curve and accelerate the learning process (Reynolds and Ehrlich, 1992).

Even though multimedia learning systems are said to have an unparalleled potential, it all depends on how well we make use of it. The system should designed to be more learnable. Interactions should be clearer and more efficient. Interfaces should offer better support for the users' goals. Information should be presented more clearly and effectively. Resolving these problems are much more difficult than identifying them. Serious interface problems are, in a way, semantic problems and these semantic problems cannot be solved through good interface techniques alone. Also, these cannot be solved by Artificial Intelligence alone. What is needed is to address these problems in a synthesis of two perspectives.

In spite of all these problems, the overriding purpose for developing these systems is to provide a substitute experience. So the ultimate aim of the developer is to produce a simulation that emulates as closely as possible the real world experience based on sound and appropriate educational, cognitive theories.

This chapter looks at the technical considerations regarding the design of multimedia systems. It analyses how the design and implementation of good interactive intelligent multimedia learning environments can be made efficient and easier, in terms of technological issues. In section 4.4.2, a design model for

developing interactive intelligent multimedia learning systems is developed, that could be specifically applied to the existing teaching package *Diagnosis*.

4.2 Design Issues

Good design is all about effective communication. With the introduction of multimedia, the alphanumeric computing has taken a new shape. We need new design techniques as well as new hardware to accommodate these changes.

ITSs are sophisticated forms of instructional systems and they incorporate ideas from cognitive science, computer science and AI. In designing an ITS, the aspects of the above subjects should be taken into consideration. Being an Instructional Designer, Pellone (1995) believes that the instructional design theories will never specify particular design practice at other than a most general level. It is up to the designer to apply the theory effectively to the particular problem domain.

In ordinary classroom instruction, the information is presented to the student, and the tutor guides him/her through initial use of information and then lets him/her practice. Finally the student learning is assessed by some way such as test or assignment. Basically these are the steps required in the educational software design. Each step has to focus on various technical and educational features during the development of the system.

4.2.1 Feasibility and Use of multimedia

When developing educational software, once the problem is defined, the very first thing to think about is, 'Do we really need a computer-based system?'. It is not logical to develop a system just because it is fashionable. The use of computer-based system must have an advantage over the traditional instruction, in some way.

Also the use of multimedia should be appropriate. Ellis (1994 cited by Rees, 1995) mentions that while multimedia is more interesting, exciting and absorbing, it is no more effective than text based computer-based education in bringing about

better, longer lasting learning outcomes. Further Rees (1995) says is difficult to read and absorb text on a computer screen compare to the same on paper. Electronic page turning can become tedious and may not be advanced as the printed presentation of the same.

The beauty of *Diagnosis* lies on its appropriate use of multimedia and the gaming style of interaction that challenges and motivates the user. The package presents pictures, videos and textual information about the pathogens that are present in the plants. In some cases it is very difficult to introduce the students to the real environments. It may be time consuming and may be costly. In such cases *Diagnosis* is convenient for the students to explore authentic looking situations. However, it is important that the student really sees the problem presentation to be closer to reality and should be able to identify the same in actual situations.

4.2.2 Type of instruction

Educational software can be of the form of Tutorials, Drills, Simulations, Tests, Games or an appropriate combination of the above. The selection of type depends on the problem domain and the learning requirements. Rees suggests that the higher order forms of Computer-Based education such as simulation are inherently more interesting and provide greater control than drill and practice programs.

Diagnosis is said to be a multimedia tutoring systems. But, to be more exact, it could be described as an 'educational game' that uses multimedia because there is only a limited use of non-interactive multimedia. It seems that if it is a simulation type of software it will be more effective (See Chapter 7 for elaborate discussion). But, considering the cost and complexity of the design and implementation, the current version seems to serve the purpose as a learning tool.

In general, tutorials are provided for presentation and guidance and used for concept and rule learning. Drills and games provide practice and are mainly used for verbal learning. Tests help assessment of learning. Simulations provide for any of presentation, guidance, practice or assessment and are used to develop skills

and attitudes. In most circumstances simulations appear to enhance learning while games enhance the student motivation (Alessi and Trollip, 1991).

4.2.3 Motivational Issues

Motivation is very important in learning. The more students are absorbed in the contents of the system, the more they learn. Incorporation of problems that challenge students is one way of motivating them. Multimedia has its special attraction of colour, sound and pictures, which are fascinating for the student.

The gaming type of challenge in *Diagnosis* is a motivational issue. And it has a balanced mixture of multimedia, just enough to make the software attractive and not too much for the student to be lost in the interface.

4.2.4 Feedback design

The amount, type and timing of feedback provided to the learner are all important. A details analysis of these issues is presented in Chapter 5. This section looks at the design of feedback.

Aversive or intimidating feedback can interfere with learning. It creates an unfriendly environment. Feedback should not destroy the motivation to continue. On the other hand, insincere feedback should also be avoided.

Irrelevant information in feedback does not help the student. At the same time, if the feedback is too attractive, the student may deliberately make mistakes just to see what happens. Feedback should not be motivating the students to deliberately make mistakes.

Feedback is not always essential. Useless feedback such as saying something that is obvious should be avoided.

4.2.5 Screen Design

In instructional displays, one of the first concerns is to design an effective, user-friendly interface. Issues such as screen layout, text, menus, colour, graphics and animation should be carefully planned in developing an efficient interface, especially for interactive environments.

The five functional areas of a screen identified by Heins (1984, from Morrison et al., 1995) are as follows.

- Orientation area, which display information such as how many more frames remain etc.
- Directions for the user in a consistent location.
- Learner display response area.
- Area for informative error messages.
- Options available to the user in a consistent location.

Hannafin and Hooper (1989) identify five functions of screen design, which match the above mentioned areas in the screen.

- focusing attention
- developing and maintaining interest
- promoting deep processing
- promoting engagement
- facilitating navigating through lessons.

A screen should not necessarily have all the above areas or functions. The design should depend on the context, with balanced presentation.

4.3 Problems in Multimedia system design

The basic difference between multimedia systems and single media systems is that in most circumstances, multimedia systems use different media to express one concept. Evidently, the basic problem with developing multimedia systems is how to combine different media to effectively convey a single concept. The fact that there is no conventions for combining media makes the problem totally difficult.

People use different gestures together with speech to get across their message to others. Their actions, words and tone are synthesized very well so that the message could not be misinterpreted. It appears that people do that without thinking much about it. Especially in everyday communication with others, we don't really give enough thought about the words, gestures and tone we are to put together or how to put them together to convey the message effectively. In some circumstances they convey their point simply with a silence.

What people could do without even thinking is not at all easy to be imitated by the computer: How to make an intelligent and meaningful combination of text, video, audio and images. We need some way to synthesise different types of information to express the message correctly.

Neal and Shapiro's (1994) picture of the anatomy of an intelligent multimedia system, gives a clear idea about the features of such systems. They define an intelligent multimedia system as an integrated working environment with a human computer interface designed as an intelligent agent with the ability to conduct dialogue with the user in coordinated multiple media. See Figure 4.1 for details.

Conduct dialogue with the user
- Adhere to respected principles of conversation
- Adhere to respected human factors guidelines for HCI and information presentation, including maintaining the context of the dialogue and maintaining consistency in displays and presentations

Maintain knowledge and belief models to enable the system to understand user inputs and compose system outputs.
- Track and model the dynamic focus of the dialogue in order to maintain context during the dialogue
- Model the user's task and the state of the user's accomplishments and progress with respect to the tasks.

Maintain knowledge bases of information about
- Modalities and user interaction
- World knowledge
- Application-specific knowledge

Act as an intelligent assistant for accessing and using application systems, through such activities as
- Assisting the user in finding relevant information on topics of interest
- Assisting the user in finding, selecting, and accessing appropriate tools to apply to the task
- Assisting and guiding the user in the accomplishment of tasks

Providing explanations and multimedia presentations to aid in user comprehension of relevant information.

Accept and understand input expressed in multimedia language
- Provide the user with flexibility in the media that is selected and combined for expressing input to the system.

Decide how information and responses are to be presented to the user
- Select modalities/media for information presentation
- Compose the output in multiple modalities
- Present the multimedia output in a coordinated manner.

Manage the windows by intelligently performing the window operations to relieve the user of the burden of performing the chores.

Figure 4.1 Anatomy of an intelligent multimedia system (Neal and Shapiro, 1994)

Even though we know what we need to do, we still do not have a clear cut methodology for how to do it. It is difficult to define a generalised frame as the design largely depends on the domain and type of instruction. There are a few theoretical suggestions and experiments in this multimedia design problem, specific to their respective problem domains.

Figure 4.2 describes one way of accessing multimedia information using cohesion. In their earlier research, Espinosa and Baggett (1996) have already shown that when material is presented passively to a learner, the hierarchical structure of the material plays an important role in learning. But if the information is presented interactively, for free exploratory learning, cohesive elements are more important than a hierarchical structure. This approach gives access to information based on one's being able to follow a line of interest.

Two elements are cohesive if they refer to the same concept, even if they occur in different modalities. The material is divided into units, representing a node in a graph. These nodes are connected together by links. A person using the system can traverse the material by travelling along the links.

Select a set of material to be browsed.
Divide the material into units, which will be nodes in the graph
Designate one or more parts of each unit as cohesive elements of a given type
Give each cohesive element type a unique short name which will be available to the user
Specify, for each element in each unit, what its predecessor and its successor will be in terms of access.

Figure 4.2 Responsibilities of the Designer (Espinosa and Baggett, 1996)

MIT project Athena has developed several multimedia learning environments. Navigation is one of them. Navigation is a visual learning environment. It is an

experiment in educational computing that was designed to integrate two aspects of knowledge, theory and experience, by presenting the theoretical components of sea navigation in the context of a simulated experience. The environment encompasses a variety of hazards for the novice pilot. The challenge can be increased by switching on a blanket of fog to reduce visibility.

Bringing simulation to the level of personal communication is a challenging prospect. Simulation certainly qualifies as a form of expression, but not many people are skilled in using it in this way (Hodges and Sansett 1993a).

Dans le Quartier St. Gervais (Schlusseberg, 1993) is another learning environment, developed for Project Athena, which shows how a database of information is transformed into an explorational learning environment that lets students explore, observe and learn about French language and culture. It develops skills and helps students acquire the knowledge needs to close the gap between language in a classroom and language in its native environment.

Experience from *Dans le Quartier St. Gervais*, reveals that multiple paths of access allow fluid movement through information to keep balance between the amount of information presented and the students' needs and interests. Supplementary options and comprehension aids should not appear except when requested.

Further in their work on Design for new Media, Hodges and Sansett (1993c) compares multimedia design to traditional film analysis, which has two divisions, one to construct individual screens (*mise-en-scene*) and the other one to treat the combination of screens (*montage*).

A context (parallel to mise-en-scene) is a set of information and control mechanisms associated with them. The problem is to select and arrange parts that make up a self-contained context. The important issues in context development are composition and balance of information, dynamic elements (decorative displays, video content, video data handling) in a composition and how they affect overall design and controls and actions (consistency, relative importance, transient action) available to the user and how these are organised and represented. In multimedia design *montage* refers to combination of contexts in space, which means

arrangement of more than one context in a screen and time which means the transition from one context to another in sequence.

The division of *Mise-en-scene* and *montage* is one way of handling problems in multimedia design. It makes it easier for the designer to concentrate on one screen at a time and later on combining them to make an efficient presentation.

In this section different media handling methods are discussed. None of them are directly applicable to *Diagnosis,* because it does not deal with interactive multimedia. It only uses multimedia elements to present problems. However it does have the problem of what media to incorporate at each step of the problem construction. In this case, the builder is the one who constructs the problem and s/he is the one who should be aware of these techniques. Then again, we assume the builders to be ordinary teachers, who probably are not multimedia experts. Consequently, media allocation is an even bigger problem in *Diagnosis* compared to the systems developed by multimedia experts.

4.4 Design Models

4.4.1 Models and Fidelity of Models

The basic idea in designing a system is to think of a solution with an open mind and to systematically consider issues. Quinn (1994) points out that the artifacts that potentially limit the best design are our tendency to solve new problem as we have solved old ones which might preclude finding a superior solution with creativity and our limited working memory capacity which makes it difficult to keep constraints in mind.

Design of instructional material can be done in two ways. One is just designing the system, of course, by a Guru in that area. The other way is applying a system of logic to develop the system. The first option is very difficult to implement. But there are quite a few models developed using logic.

A model is usually considered to be an abstraction and simplification of a defined referent system, presumably having some noticeable fidelity to the referent system (Hayman, 1974; Logan, 1976 cited by Morrison et al, 1995). A more precise definition by Rothenberg (1989) is 'Cost-effective use of something in place of something else for some cognitive purpose' (from Kemp, 1995). The cost may be the reason why we needed a model in the first place.

Models of instructional design have descriptive, prescriptive, predictive, and/or explanatory elements in varying degrees. Listing the purposes of a model will help determine what aspects of the original system are to be modelled and what form (Physical object, computer program, formal description or mental model) the model is to take (Kemp,1995).

Andrews and Goodson (1995) identify the following purposes of models of systematic instructional design

- Improving learning and instruction by means of the problem solving and feedback characteristics of the systematic approach.
- Improving management of instructional design and development by means of the monitoring and control functions of the systematic approach.
- Improving evaluation processes by means of the designated components and sequence of events, including the feedback and revision events, inherent in models of systematic instructional design.
- Testing or building learning or instructional theory by means of theory -based design within a model of systematic instructional design.

Fidelity refers to the degree of correspondence between the model inside the computer and the model that we wish the user to acquire (normally a mental model) (Kemp,1995) or how closely the system imitates reality. It is believed that increasing fidelity would lead to better transfer of knowledge. Transfer of learning refers to the extent a student can apply what is learned in instruction to a new situation, usually intended real performance. It is a complicated function which includes not only initial learning at the time of instruction, but also the *similarity* of the instructional situation to the performance environment, the *perceived* similarity of the instructional situation to the performance environment, and the student's level of *motivation* (Alessi and Trollip, 1991).

Alessi (1988) proposed that fidelity/learning function at the time of learning may be represented by figure 4.3.

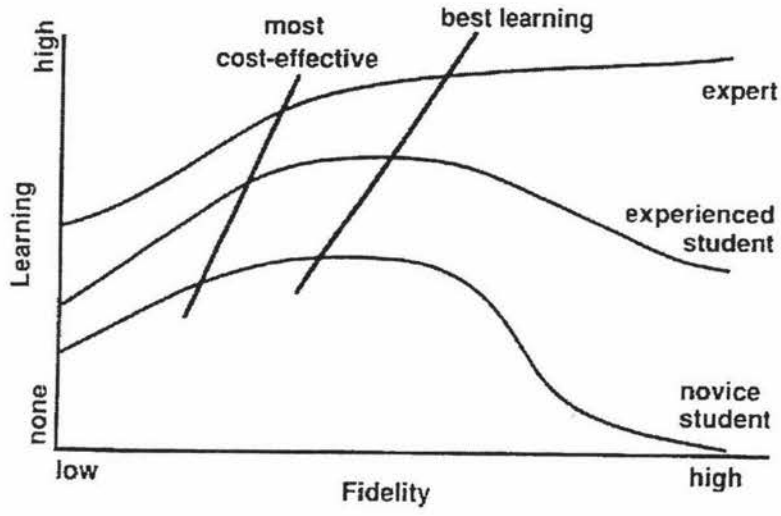


Figure 4.3 Hypothesised relationship of fidelity and learning (from Alessi andTrollip, 1991).

The following can be interpreted from Figure 4.3. For a novice student, while low-fidelity instruction produces learning, some increase in fidelity might increase learning. For an experienced student high-fidelity produces better learning. For an expert very high-fidelity systems or even real situations leads to effective learning.

4.4.2 Design Methodologies

There are different methodologies available for the design of computer-based instructional multimedia systems. This section list a number of such models and points out their advantages and disadvantages.

Andrews and Goodson (1995) list the following tasks in model development. These tasks are generic and applicable to development of any systems. While carefully considering selection of media, there is no specific guidelines as to how to design for multimedia.

Task	Definition
1	Formulation of broad goals and detailed subgoals stated in observable terms (output)
2	Development of pretest and posttest matching goals and subgoals (Tests)
3	Analysis of goals and subgoals for types of skills/learning requires (Analysis)
4	Sequencing of goals and subgoals to facilitate learning (Sequencing)
5	Characterisation of learner population 'as to age, grade level, past learning history, special aptitudes or disabilities, and, not least, estimated attainment of current and prerequisite goals. (Learner attributes)
6	Formulation of instructional strategy to match subject matter and learner requirements (Strategy)
7	Selection of media to implement strategies (Media)
8	Development of courseware based on strategies (Development)
9	Empirical tryout of the program with learner population, diagnosis of learning and courseware failures, and revision of courseware based on diagnosis. (Tryout/revision)
10	Development of material and procedures for installing, maintaining and periodically repairing the instructional program (Install/maintain)
11	Assessment of need, problem identification, occupational analysis, competence, or training requirements (Need)
12	Consideration of alternative solution to instruction (Alternatives)
13	Formulation of system and environmental descriptions and identification of constraints (Constraints)
14	Costing instructional programs (Cost)

Figure 4..4 Tasks in model development (Andrews and Goodson, 1995)

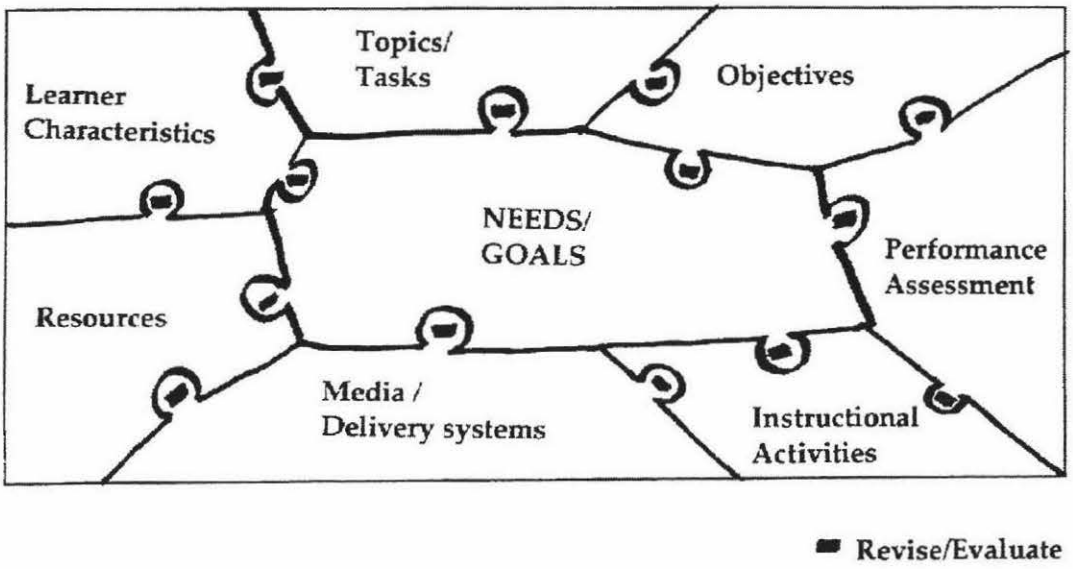


Figure 4.5 Model for Multimedia Design (Reynolds and Ehrlich, 1992)

Figure 4.5 shows a graphic representation of the design process described by Reynolds and Ehrlich (1992). The processes can be reiterated at any point and so it is not linear. It is derived from the traditional instructional system design models. It does not emphasise the impact of multimedia in the design. Basically, it is a conventional model presented in the form of a puzzle with interlocking pieces.

Godfrey's Multimedia Development Life Cycle (Figure 4.5) is comparable to the traditional Systems Development Life Cycle. The Life Cycle of computer-based learning project proposed by Dymalla (1994) is quite similar to this model.

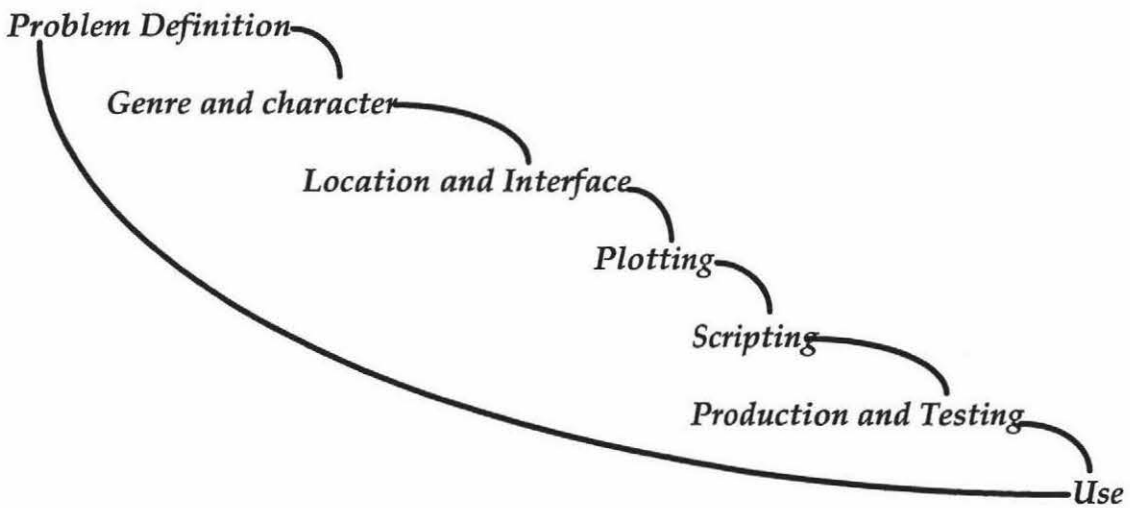


Figure 4.5 : Multimedia Development Lifecycle (Godfrey, 1995)

The design envelope proposed by McNamara emphasises multimedia. It has three components - user, design and management requirements. It is a multi-dimensional model, which converges towards expert and novice states.

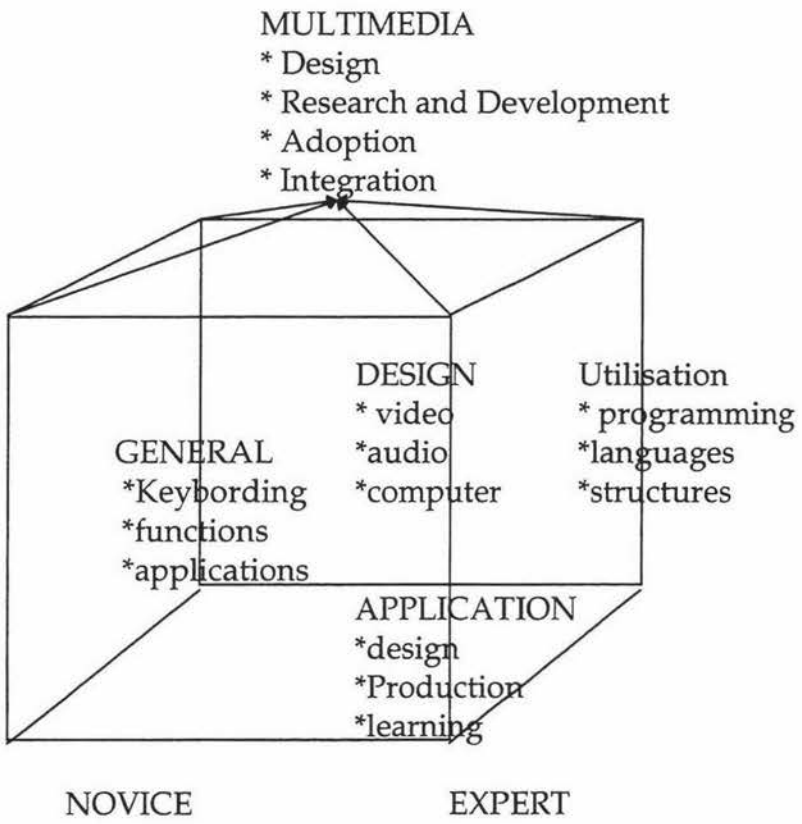


Figure 4.7 Design envelope for multimedia (McNamara, 1994)

McNamara claims that the design envelope permits the incorporation of many considerations in developing multimedia systems for learning such as the attributes of multimedia and its components, an appropriate theoretical framework for investigation and development, management issues in the adoption and implementation of interactive multimedia systems.

Interface Communication Management (ICM) proposed by Litchfield (1994) is a user-centred approach to the design of multimedia. It is influenced by the theory of 'Adult Communication Management'. Litchfield says that due to multimedia's capacity and requirement for interaction, the designer's principal task is to collaborate together and successfully manage interface communication between the user and the screen based media information. This approach mostly

concentrates in achieving multimedia's potential for improving human communication.

According to Quinn (1994) the process of developing instructional technology involves aesthetic considerations that are not amenable to the regime of science. Games, in particular, require elements of imagination and creativity. Quinn presents a methodology for designing educational computer games based upon what's known about how people think, learn, and design :

Analysis
- determine target performance
- determine learner characteristics

Specification
- determine pedagogical approach
- situate the task in a model world
- elaborate the details
- incorporate underlying pedagogical support
- map learning activities to interface actions
- map learning concepts to interface objects

Implementation
- prototype

Evaluation
- pilot test

(Repeat)

He justifies the repeat or more than one-pass analysis by saying that the inherent complexity in human activity and the social setting in which it occurs means that the initial information collection was not likely to produce all the factors that are relevant in the design.

Hedberg et al. (1994) proposes the following multimedia design model in a constructivist framework :

Phase 1: Takes the basic information derived from a needs assessment and converts it into a description of the project space

Phase 2: Reviews the basic description and seeks to link the elements through an appropriate instructional or presentation strategy.

Phase 3: Expresses goal of linking the ideas into a potential interaction structure.

In his paper on Creating Interactive Learning, Barker (1994) mentions a paradigm called MAPARI (Mimicry, Apprenticeship, Practice, Assessment, Refinement and Improvement). A model based on this paradigm must accommodate the following basic functions : knowledge acquisition, skill development, skill rehearsal, problem solving and self-realisation. The Integrating Learning Design in Interactive Compact Disc model of learning design is based upon MAPARI.

Integrating Learning Design in Interactive Compact Disc (ILDIC) is one of the projects in research programmes of DELTA (Developing European Learning through Technological Advance) initiated by European Community (EC) (Barker, 1995). Basic perspectives of learning design in ILDIC model are as follows.

- learning theory mix (behaviourism, cognitivism and constructivism)
- instructional position mix (provision of information, tutoring, remediation, reflection experience, counselling)
- machine character mix (tutor, instructor, coach, assistant, book, toolkit, tester, simulator, database and toy)
- environmental factors (home environment, workplace setting, open learning center, conventional classroom, virtual classroom, conference situation)
- mode of use
- locus of control
- extent of intervention
- aesthetic features
- content
- role of technology

SMISLE (System for Multimedia Integrated Simulation Learning Environments) is a project that aims to define simulation based exploratory learning environments that incorporate instructional support for learners and to provide an authoring toolkit for such systems (Barker, 1994). Progressive model implementation, assignments, explanations and hypothesis scratchpads are four types of instructional support measures that have been implemented in SMISLE. These measures are designed around five different models pictured in Figure 4.8

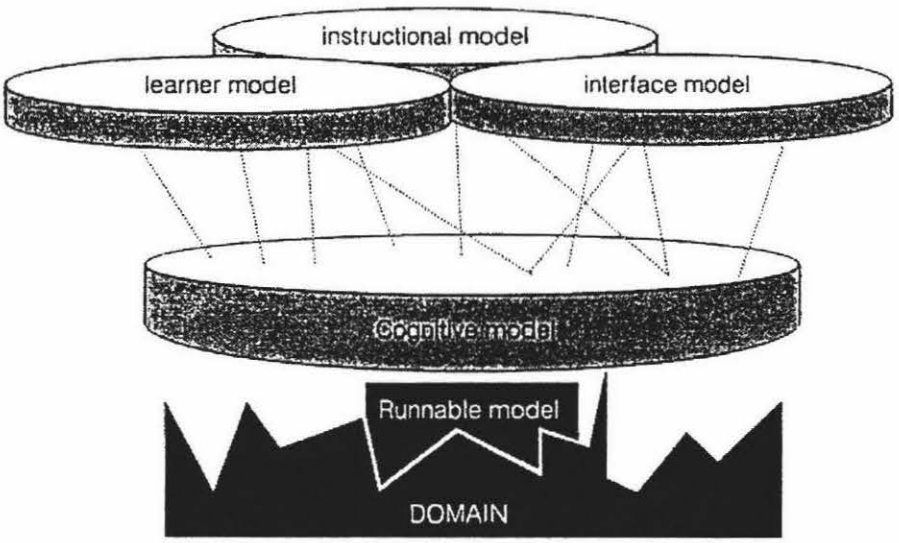


Figure 4.8 Different models identified in the SMISLE project (Barker, 1994)

The cognitive model is a combination of computational model, operational model and Extensions. The runnable model is the one that makes the simulation run. It has the abstract description of a domain. Its objective is to enable an optimal fast and efficient simulation. Instructional model contains instructional functions. Learner model store or infer information regarding learners. It is used for activating the instructional model. The Interface model contains all kinds of graphical objects, windows etc. enabling the user to interact.

In the light of the above mentioned design methodologies, it appears that *Diagnosis* could be modelled using a traditional development methodology with detailed analysis and design consideration for the use of multimedia, especially to the authoring of the problems.

4.5 *Diagnosis* Design

4.5.1 Problems in *Diagnosis* design

The design issues in Section 4.2 and design model in Section 4.4 should be considered in designing *Diagnosis*. Since it is already implemented as a package it does not really matter, except for the proposed modification.

Among the design issues discussed, use of multimedia and type of instruction of *Diagnosis* is very impressive. The use of multimedia is necessary for *Diagnosis*, because the domain of plant pathology has a very high usage of non-textual information such as pictures and diagrams. The issue that needs proper thought is feedback design, which is incorporated in the Builder module.

Diagnosis was developed using a traditional software life cycle methodology, with much less consideration about multimedia. The multimediality or the usage of multimedia is relatively small. Therefore, handling media is not a big problem as it would be in a simulated environment with interactive multimedia. As discussed in Section 4.3, it a hidden problem, which needs greater attention. How does one select media for optimum presentation of the problem?

An approach to the solution of this problem is to provide help and guidance in selecting the media throughout the construction of the problem and accommodating these help facilities in the authoring system. Since most of the problems being constructed are unique, we cannot have a generalised pattern of media allocation. Selection of media highly depends on the nature of the problem. So, providing the guidance in this regard is extremely hard, unless the authoring system understands the nature of the problem.

Diagnosis has an authoring system for the tutor to construct problems. The problem will be presented according to how it was constructed. So, the authoring of problem should be well defined for the presentation to be realistic. People do not normally consider how simulations are constructed. Here the main problem is making the author of the problem think about the construction as well as presentation of the problem.

Efficiency is different from ease; in order to be considered easy, the encoding must of course be efficient, but in addition the authors must be able to see how to achieve the objectives. Whether or not the writer can do so depends on how closely the conceptual framework embodied in the tools matches the way that person thinks about what s/he wants to make (Hodges and Sansett, 1993b). The main challenge is to design an authoring system for *Diagnosis* that is clear enough and efficient enough to become truly easy to use. A detailed description of this proposed authoring system is given in Chapter 6.

4.5.2 Representing *Diagnosis*

In many ways the interaction and interactivity that was described above is very similar to the basic mechanisms that underlay the paradigm of Object Orientation that is used in computer systems design and software development (Winbald, Edwards, and King, 1990). In object-orientation, an *object* or *instance* has state, behaviour and identity; the structure and behaviour of similar objects are defined in their common class. The state of an *object* encompasses all of the properties of the object plus the current dynamic values of each of these properties (Booch, 1991). In this paradigm, an object reacts to the message that another object sends just like the system reacts to the user's actions.

Since *Diagnosis* is interactive software and was already programmed using a object-oriented methodology, it was decided to represent its design using Object Modeling Technique (OMT, Rambaugh et. al, 1996) which is a Object-Oriented methodology. Object diagram for *Diagnosis* is given in Appendix E.

Essential modeling is an extension and adoption of Jacobson's use case model of Object-oriented software engineering (Constantine and Lockwood, 1994, Constantine, 1995). Essential use cases provides a flexible, technology-free initial design model. They capture the essence of the system and they are use-centred. For a detailed description of the interface of *Diagnosis*, essential modeling is used. Appendix B describes essential use-cases of *Diagnosis*. Some screen designs from *Diagnosis* is given in Appendix D.

4.6 Discussion

The unique characteristics of multimedia are technological richness, broadness and diversity of the media. In conventional media, the users are viewers, readers and listeners, whereas in multimedia they are users. This radical change certainly has an impact on society. The following statement proves that. In 1992, video game publishing, grossed more than the combined US film and music industries (Litchfield, 1994).

McNamara et al. (1994) say 'undoubtedly, the composition of a vast array of visual information (both still and motion) and aural information (both real and simulated) in combination with ever increasing speed of access and memory capabilities foreshadow the possibilities of an educational and training revolution'. However, many of the limitations commonly attributed to interactive multimedia may be minimised by careful instructional systems design.

Some argue that tried and tested methods of conventional system design development are readily adapted to these new media whereas others recognise new design methodologies are needed to develop multimedia systems. While Godfrey says 'new wine in old bottles' Wills and Stuart wants to move from 'Paperless Book' to something different just like we moved from 'Horseless Carriage' to 'Automobile'.

It is hard to say which side of the argument is right. Both sides can be justified depending on the system. As long as a good system could be designed, the methodology is acceptable.

A major strength of all computer-based instructional technologies rests on their ability to provide motivation, enrichment and elaboration. The parameters that constitute a good design are a balanced mix of technology and educationally effective learning theories.

One should make sure that the proposed approach addresses the learning need, meet the expectations of the target learners, enables learners to do things they can't do using existing resources, motivates the learners to interact with the program

and other learning resources, and places control in the hands of learners as far as is possible within the parameters of the program.

Most of all it should be remembered that ' it is the student that is intelligent, not the computer' (Ellis, 1994 in Rees, 1995) - until humanlike computers are made.

Chapter 5

Cognitive Issues, Intelligent Features and Human Factors

5.1 Introduction

People need people. According to Hegel, we need the recognition of others in order to feel confident of our own identity. But, he also suggests that we'd prefer not to reciprocate; we want to master others. That's why the idea of a computer with which we can truly interact with another person, is so appealing. The simulated person in the computer wouldn't insist that we pay attention to its needs. It would devote itself to meeting ours (cited by Harvey, 1994).

Even though discovery learning is an excellent way to learn, life is too short for that. It is impossible for a person to learn various things s/he wants to learn within a lifetime, especially by discovery learning. A reasonable solution for this situation is guided discovery systems, that have the advantage of discovery learning and takes less time to discover things. Computers are more likely to be beneficial in a number of other situations as well. For instance, if the cost of instruction by other methods is very high, or if safety is a concern, or if the material is very hard to learn by other methods, or if extensive individual student practice is needed, or if student motivation is typically lacking, or if there are logistic difficulties in traditional instruction (Alessi and Trollip, 1991) computer solutions seem to be effective.

Computers can also assist instruction by providing tools (e.g. providing a calculator in a math tutor). Thus it frees students from physical requirements and enables them to make conjectures and explore relationships (Yerushalmy and Houde, 1986). Taylor (1980) identifies three possible roles for the computer namely Tool, Tutor and Tutee. The computer acts as a tool when the student uses it and appropriate application software to perform a task. It is a tutor when it takes on the teaching function, presenting a series of screens of information, test the student and give feedback. It is a tutee when the student teaches the computer and in doing so learning something as well (e.g. computer programming).

In order to design and produce better computer leaning systems, understanding of human cognition is necessary. Even though researchers have studied the areas of cognitive skills, acquisition of cognitive skills, and modeling of cognitive skills, the principal questions such as what is knowledge?, how it should be represented?,

how can it effectively be extracted and represented for novice learners? do not have definite answers.

This chapter seeks to find the appropriate cognitive tools and intelligent features that would effectively and efficiently enhance the educational value of *Diagnosis*, if incorporated. Basically it is a search, with respect to *Diagnosis*, for the features that makes the lessons attractive to the users, instructionally effective, and motivating as well as for the human factors that affect the learner's role in managing the learning.

5.2 Learning theories

Instructional design and development must be based upon some theory of learning and/or cognition. Effective design is possible only if the developer has awareness of the theoretical basis underlying the design.

Gagne (1977 in Steinberg, 1991), from the view of educational psychology, conceptualises learning in terms of categories of skills and capabilities and the conditions under which they are learned. He groups the diverse learning outcomes into the following five categories.

- *Intellectual skills:* the rules and concepts that constitute a considerable proportion of school learning.
- *Verbal information:* the capability of verbalising information.
- *Cognitive strategies:* employed by the learner to control their learning.
- *Motor skills*
- *Attitudes.*

Bradford (1979, in Steinberg, 1991), from the point of cognitive psychology, explores learning, remembering, and understanding from a process perspective. He presents a framework of four components - learner characteristics, critical task, nature of material to be learned, nature of learning activity - and emphasises that

the most significant idea underlying this framework is the 'interaction' among the components. Learner characteristics are the attributes such as prior knowledge, structure of that knowledge in memory, beliefs and expectations, knowledge about one's own knowledge, developmental maturity and experience that are intrinsic to each individual.

Regardless of the theoretical perspective, both Gagne and Bradford identify four components that are central to learning. *Target population* (who is learning), *Goals* (what are they supposed to learn), *Task* (the materials and skills involved), and *Instruction* (the externally planned activities).

From his research on cognition, Anderson (1983) concludes that there are three stages of knowledge acquisition namely encoding, proceduralisation, and composition. Encoding is representing the facts as production rules. Proceduralisation is reorganising the facts that are in a declarative form into procedures through experience. Composition means groups of rules are collapsed into a single rule that combines their effects. Anderson (1988) describes three approaches to encoding knowledge into knowledge bases. The *Black box* approach does not require us to actually codify the underlying human intelligence. The *Glass box* approach is basically entails the standard stages of development of an expert system. And the third approach is Cognitive models of the learner.

Knowles (1990) labels theories of youth learning as *pedagogy* and theories related to adult learning as *andragogy*. The pedagogical model is a teacher-directed model which gives the teacher the full responsibility of making all decisions about what will be learned, how it will be learned, and when it will be learned. In contrast, the andragogical model is a student-centered model which considers the self-directing nature of adults, their previous experience, and their readiness to learn when they need to know or do something. However, both the models assume the following factors- need to know, the learner's self-concept, the role of the learner's experiences, readiness to learn, orientation to learning and motivation (Pellone, 1995).

Benamou and Celentano (1995), proposes the following educational strategies:

Program oriented paradigm	a course is made of interactive applications, whose sequencing is determined by predefined methods.
Information oriented paradigm	the learner has access to a web of educational material and s/he can navigate according to some information exploration function.
Simulation oriented paradigm	learning by doing in a real world like environment.
Strategy oriented paradigm	A set of pedagogical rules are applied to the network linking the concepts to be learned, allowing the system to plan the presentation order of the concept to the student.

Liddle et al. (1996) proposes an approach which says that acquiring domain specific knowledge which initially requires a decomposition of the training objective into primitive generic tasks. And they try to provide training, based on domain independent learning styles and training strategies. They divide generic tasks into the following essential behavioral subtasks:

- *Interpretation* is the process of generating the current state of the world from observation and/or physical measurement.
- *Identification* involves determining the past values from the current (interpreted) values of the system.
- *Prediction* determines predicts future value of the system states from the current value.
- *Decision* is the process of generating conclusions from the interpreted, identified or predicted state of the system.
- *Execution* is the process of turning the decisions made into actions.

Constructivist cognitive science argues that learning is a constructive process in which the learner is building an internal representation of knowledge and a personal representation of experience. A shift from this cognitive science theory is 'situated learning'. There is a need for the learning experience to be situated in *real*

world contexts (Brown, Collins and Duguid, 1989; Ragoff and Lave, 1984), which means that the task is not isolated, but it is part of a larger context (Bansford et al., 1990). Figure 5.1 pictures the difference between these two paradigms.

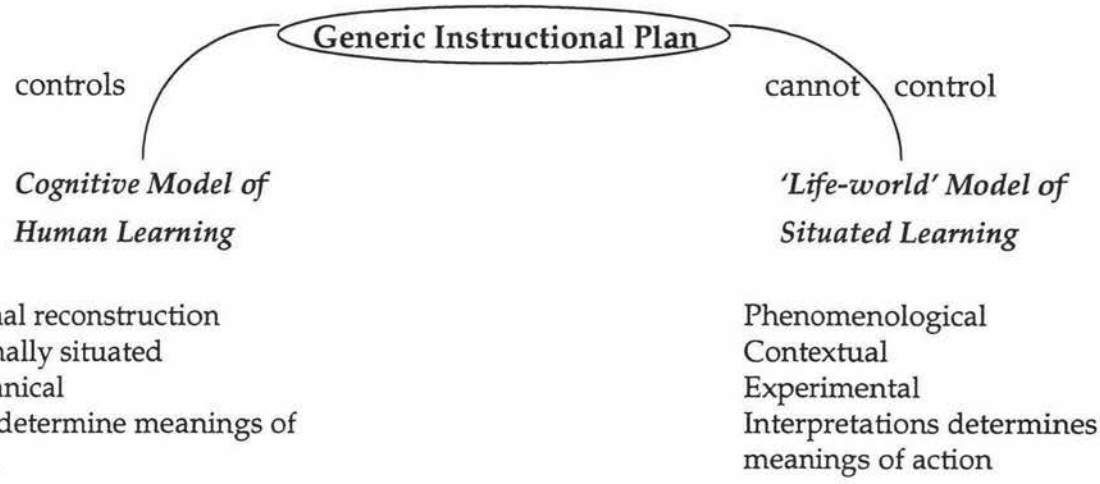


Figure 5.1 Cognitive model/'real-world' dilemma (Streibel, 1995)

The problematic aspect of the cognitivist paradigm is *Can human beings reason and learn in a situation where they have to deny the contextual nature of their thinking and knowing?* Learning is no longer a matter of ingesting externally- defined, decontextualised objects, but a matter of developing context-bound discourse products. This epistemological paradigm shift is clearly defined by Brown (from Striebel, 1995).

Cognitive learning	*Situated learning*
Decontextualised	Contextualised
Knowledge	Practice
Goals	Expectancies
Tasks/Problems	Activities
Solipsistic	Interactional
Formal	Coordinated
Definitional	Constraints
Problem solving	Dilemma handling
Looking at	Looking for
Explicit theories	Implicit theories
Reference fixed	Reference negotiated
Efficiency	Rationality

Figure 5.2 Epistemological paradigm shift (Brown,1988)

Educational systems like *Investigating Lake Illuka* and *Exploring the Nardoo* are based on situated learning theory. They prove that interactive multimedia and situated learning together can be of high educational value. It could be argued that simulated learning environments cannot be situated learning environments for the context are not exactly like real world contexts. This argument is valid for some situations, in which transfer of learning does not occur. On the other hand, we cannot deny the fact that these 'authentic looking' situations do prepare the learner for better performance in 'real situations'.

The optimum mix of the learning theories differ from one situation to another depending on the nature of learning and the training activities that are to be encouraged and promoted. In addition to learning principles, elements like title screen, objective, entry level, online assistance, questions, branching, remediation, feedback are also useful as instructional enhancers (Pellone, 1995).

Diagnosis is based on a combination of learning theories. It could be viewed as a learning by doing environment, based on constructivist theory. It also could be seen as a simulated environment based on situated learning, even though the simulation is not very authentic looking. Most of all it could be viewed as a game, which makes it very attractive. Rather than simple drill and practice, interaction with puzzle-type games are meant to make the learning environment more stimulating (Woolf and Hall, 1995).

The lack of pedagogical support in games reflects the general lack of emphasis that has been placed on games. But the motivating effect of computer games provide a strong argument in their potential for instruction (Hedberg and Harper, 1995). *Diagnosis* would be a better learning tool if it provide more learning support.

From all three view points of learning theories, implementation of *Diagnosis* appears to be suitable. It is up to us to decide which combinations are more appropriate. It could be a learning by doing game in constructivist framework or a situated game in situated learning paradigm. Due to practical limitations, the alterations to *Diagnosis* are proposed from the first point of view. However, considering the suitability for knowledge transfer in the domain of plant pathology, possibilities for situated gaming are analysed in Chapter 7.

5.3 Cognitive Issues and Human Factors

Schiffman (1995) says, designers should have an understanding of the principles of human physical, emotional, social, and mental growth and development. A knowledge of how socioeconomic status, IQ, sex differences, cognitive styles, creativity and motivation may affect learning is also important. This background provides valuable insight into the target population. There is much for the computer-based instruction designers to learn from the field of cognitive psychology. The more they understand the target population the more they can design the system just for them.

Lippert and Trollip (1986, in Allessi and Trollip, 1991) describes an existing approach which is a combination of learning by doing and learning by discovery. This approach enables the user to apply the logic of the structure of knowledge to segments of one's own store of knowledge in order to create a computerised knowledge-base. Chief claims from the application of this approach are:

- Improved problem solving, in identifying problems, defining refining and representing them. exploring, assessing and acting on alternative strategies, identifying and evaluating their effects;

- Gains in procedural knowledge, of analysis of scope, meaning and critically of concepts, pattern recognition, synthesis of discrete steps into logical procedures, testing of knowledge bases for accuracy, validity, consistency and coherency.

- High motivation, due probably to practical, goal-directed activity, conceptual conflict, challenge, curiosity, competition;

- High transfer to other tasks, promoted by development of skills of organisation of knowledge, identification of its essential features, understanding, of basic principles and the similarity of the classroom training and testing to real-life problem solving.

They also point out are some problems and interactions in this approach - individual differences in cognitive style and learning style, age as a factor, difficulty and natural structure of the content under study, motivation to learn,

large amount of time necessary to employ the technique and the large individual differences in the amount of time required to reach criteria.

The process of instruction involves presenting the information, guiding the student, practice and assessing the student learning. Hedberg (1985) suggests the following strategies or design heuristics for educational systems. Interactivity of computer visual tasks, degree of learner control over sequencing and content, need to comprehend temporal relationships, need to organise and retrieve large numbers of visual displays, instructor manipulation of the visual presentation of the concept to be learned and learning manipulation of visual information while maintaining sequences coherently. However, focus should be on effective performance and problem solving rather than the ability to remember facts and repeat theory without real understanding about its applicability.

5.3.1 Comprehension

What we perceive must be interpreted and integrated into our current knowledge of the world (Anderson, J.R. 1977). We should be able to classify, evaluate, manipulate and apply the information, in addition to storing and retrieving. The type of learning desired must determine the type of presentations and activities of a lesson (e.g. whether to use multiple choice questions or short answer).

Feifer (1983) explains the concept of 'graph mapping'. Comprehension of text can be enhanced by having learners construct a graphic map representing the text. Learning to construct a graphic map requires individual feedback. Automated instruction is the most reasonable means of providing this feedback - by inferring the underlying cognitive structures.

Decoding (sounding out) skills also influences comprehension. Good readers can devote most of their attention to comprehension because they decode almost automatically. Readers who expend considerable effort on decoding have few resources left for understanding. Some research even suggests that slow decoding is a cause of poor comprehension (Lesgold, 1983).

5.3.2 Metacognition

Metacognition is knowledge about your own knowledge and cognitive processes. It is knowing your capacity limitations, knowing some learning strategies and when to apply them. Metacognition involves actively monitoring and regulating your learning activities (Brown, 1977). Paris and Winograd (1990) defined metacognition as knowledge about cognitive states and abilities that can be shared among individuals, including the affective and motivational aspects of thinking (from Hedberg and Harper, 1994).

The ability to explain and defend decisions, is related to the development of metacognitive skills - thinking about thinking. Metacognition can be thought of as cognitive awareness. It is multidimensional. Osman and Hannafin (1992) outlined several components of metacognition.

Metamemory includes skill with different memory strategies and an awareness of how to choose which is appropriate for a given task.

Metacomprehension includes a range of skills that include being able to detect when one fails to comprehend something and is able to take appropriate remedial action.

Self regulation is an individual's ability to make fine adjustments to errors detected when the instruction provides no feedback.

Schema training has to do with getting learners to generate personally relevant structures for understanding material and becoming less dependent on structures provided by instruction.

Transfer is the ability of the learner to apply a strategy to an unfamiliar and dissimilar learning task.

One method of helping learners respond to greater metacognitive demands is to provide opportunities for checking their own progress and to provide advice about metacognitive strategies to the learner (Schwier, 1995). Metacognitive support should be employed in tutoring systems to make it more useful to the students. It is also proposed that higher order thinking skills causes focusing on important points, helps students gain familiarity with text structure, aids

retention, generates useful alternative texts to supplement materials read and causes active participation in learning (Bianco and McCormic, 1989 in Schroeder and Kenny,1994).

Research on 'note taking' has demonstrated that users who organise and elaborate on the information collected are more likely to create retrieval cues, link to prior knowledge and make the new information personally relevant. About the notebook in their software *Exploring the Nardoo* Harper et al. (1995) point out that it equips students to view and then critically evaluate or compare different representations of the same information and gives an opportunity to establish cognitive links between different media forms. Another educational system *Investigating Lake Illuka* provides metacognitive support through cognitive self-management, provision of prompts and experience of experts (Hedberg and Harper, 1994).

Careful and guided use of supportive structures such as *genre templates (generic templates)*, will provide support for the poor, slow students by reducing the amount of reading, lowering the cognitive load and orienting the work more towards the student-centered learning (Harper et al., 1995).

5.3.3 Interactivity

Five basic classes of systems identified by Checkland (1972 in Barker, 1995a) are

- Natural systems
- Transcendental systems
- Designed physical systems
- Designed abstract systems
- Systems that involve human activity.

An important class of human activity system is that which deals with learning and training to facilitate the development of an informed society. These kinds of systems are referred to as knowledge and skill transfer systems, which can be organised into two broad classes - ones that depend upon direct human interaction and those which are in some way mediated by technology (Barker, 1995a).

Undoubtedly, computers currently offer one of the most exciting and potentially useful mechanisms for the development and delivery of software to support knowledge and skill transfer. The combination of close monitoring, feedback and corrective action are basic requirements of all interactive learning and training processes.

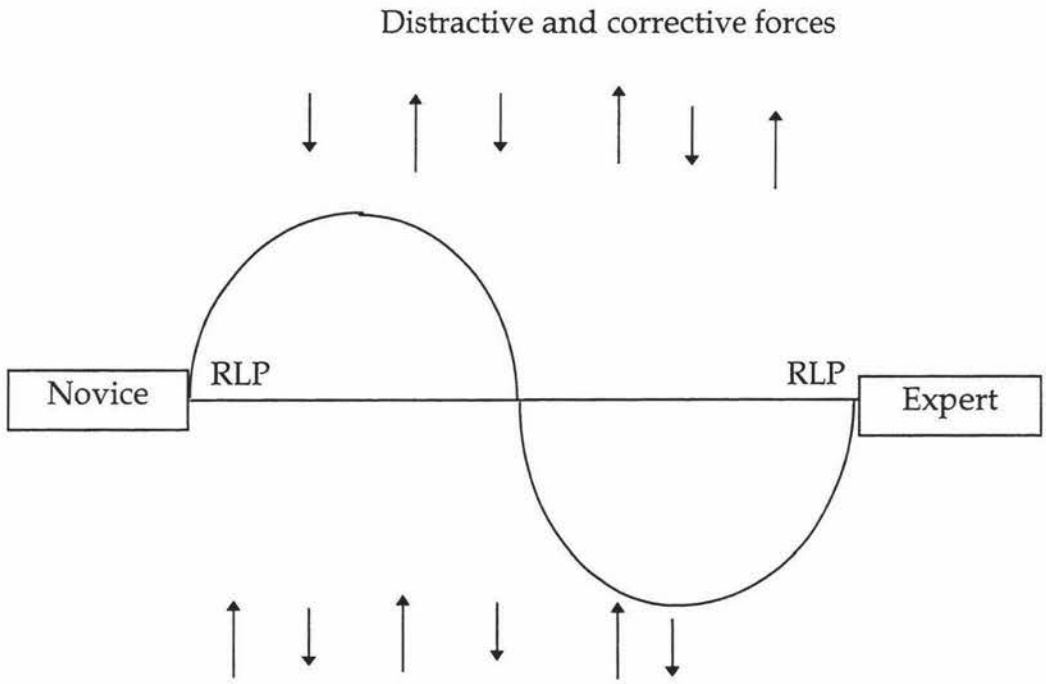


Figure 5.3 Interactive Learning (Barker, 1995a)

The linear transition required learning path is the ideal learning line from novice to expert. The non-linear paths are the results of the effect of distractive and corrective processes that might influence the learning (Barker, 1995a). Mal-learning means learning the wrong things. It is analogous to the creation of cognitive bugs and mal-rules in Intelligent Tutoring Systems (Sleeman and Brown, 1982; Self, 1988). Sometimes mal-learning can lead to the development of cognitive dissonance, a conceptual conflict between two or more cognitive structures (Gardiner and Christie, 1987).

In their project on an authoring toolkit that aims to define exploratory learning environments based on simulation that incorporate instructional support for learners in such a way that effective and efficient learning will result, de Jong et al. (1994) implemented the following instructional support measures - progressive

model implementation, assignments, explanations and hypothesis scratchpads in order to provide support to the learner.

The interaction is seen primarily from its function as a purposeful dialogue. The aim of intelligent tutoring dialogue will be to make the most of available interfaces and computational power while retaining as many natural dialogue features as possible (Peter-Brown, 1989). Interactions and physical scheduling of related information must be planned with special care in order to ensure the software is educationally effective.

5.3.4 Perception and attention

Learning depends on the learner attending to the instruction and correctly perceiving the knowledge or skill. Gaining the attention of the student is the first event in preparing the student for instruction (Gagne and Briggs, 1974). Perception is the foundation of learning and every individual operates at different levels of symbolic perception or processing.

Perception is constantly strained by many competing stimuli. Attention may falter during interaction or be attracted to different stimuli than the ones desired (Alessi and Trollip, 1991). Effective instruction depends on presentations designed for easy and accurate perception. Factors that facilitate perception include *detail and realism*, the *use of sound versus visuals, color, characteristics of text* such as font and size, *animation, position of screen elements*.

For proper perception to occur, the attention of the student must be maintained throughout the lesson, not only in the beginning. To maintain the attention, factors that affect the attention should be carefully considered. These factors include level of student involvement, personal interest of the student, prior knowledge of the student, difficulty of the lesson, novelty, familiarity, pacing and variety.

5.3.5 Memory

What we perceive must be carefully stored in the memory to be retrieved for later use. Human intelligence is capable of handling vast amount of information. But, we cannot assume that the important things are not only perceived but also properly stored all the time. Two principles, the principle of organisation and the principle of repetition (Fleming and Levie, 1978) underlie almost all methods of enhancing memory.

Showing the student how to organise the new information aids the recollection of the information for them. When the information is vast or hard to organise, the principle of repetition is used to memorise it.

The information presented to the student should be organised so that it will be represented in his/her memory. Otherwise, it should be repeatedly stressed - of course using different media - for it to be memorised and to enable it to be recalled when needed.

5.3.6 Active learning

For active learning to occur, the student should be immersed in the system. Interactive multimedia can provide a context for developing 'legitimate peripheral participation'. It refers to the engagement of a novice in a socially-based practice in which they can perform the same range of skills as an expert. There are a few dimensions that provide legitimate peripheral participation - immersion, fidelity of representation, active participation, creating an environment for practice and supporting network learning environment (Hedberg, 1995). These dimensions form the basis for the development of appropriate supports for learning from Interactive Multimedia.

Interaction not only maintain attention but also creates and stores new knowledge and skills. Even though it is extremely hard to design interactions which are frequent, relevant and increase learning, it is a must for effective interactive learning.

5.3.7 Motivation

The constructivists argue that the learning outcome depend on factors like the learning environment, the prior knowledge of the learner, the learner's view of the purpose of the task and the motivation of the learner (Hedberg and Harper, 1994).

Motivators should be used which are intrinsic to the instruction rather than externally applied (Lepper and Chabay, 1985). Keller suggests four factors - maintenance of attention, relevance of the material, student confidence, student satisfaction - are essential to motivation. Malone and Lepper (1987) define two broad categories of motivators, personal and interpersonal. Personal motivators are further classified into *challenge, curiosity, control* and *fantasy*. Interpersonal motivators are classified into *cooperation, competition* and *recognition.*

An individual is viewed as a problem solver. The challenge is in solving a problem by applying a skill that the person values. Challenge is thought to be motivating because it seems to be giving a feeling of self-esteem to the individuals. For an activity to be challenging, it has to have at least one goal. Moreover, if the person is sure about achieving the goal, it is not challenging. Therefore, an element of uncertainty is necessary.

Curiosity may be related to sensory variations such as changes in light or sound or cognitive factors like a discrepancy between what we expect to find in a given situation and what we actually find. Cognitive curiosity is stimulated when students are interested in a topic and are confronted with a situation that is incomplete or apparently incongruous.

A person is motivated in an empowering environment, one in which the outcome depends on her own responses. Control is an intrinsic motivation that comes from a sense of self determination. There are individual differences in the motivational value of learner control and motivation itself in general. Both theory and experiments stress that there is some optimal level of motivation. For example challenge must be within an achievable range. Fantasy helps to satisfy emotional needs and enables students to vicariously experience power, success, fame and fortune.

Target population should be considered in providing motivation. It should be noted that the motivators that affect the small children are not necessarily motivating for older children or adults (Steinberg, 1991). When deciding on motivators for a particular software, finding the processes by which motivators affect learning, comparing and contrasting why some computer assisted lessons motivating and others not together with special attention to the target population and nature of the subject matter would help identifying the motivators that are appropriate to the student model.

5.3.8 Locus of control

Locus of control means whether control of sequence, content, methodology and other instructional factors are determined by the student or the system or some combination of two (Alessi and Trollip, 1991). The source of responsibility for managing many aspects of instruction is called the locus of control.

The evaluation of the student, diagnosis of needs, selection of further study material are performed by the instructor who takes into account the student's individual long term objectives, learning pattern to date, history of errors and difficulties etc. The instructor provides feedback by tracking the users response pattern, identifying patterns that are sub-optimal for the task in hand, diagnosing the probable knowledge-deficiency and interrupting the user (Romiszowski, 1987). It cannot be justified that the assessment should be done by the student him/herself for self assessment is biased in most of the cases. Even though learner control is supposed to be motivating, in cases like this it is necessary to be controlled by the instructor.

It is very difficult to specify a clear cut locus of control for all learning systems. It depends on the nature of the subject, objective of the learner, time available etc. Learner control is discussed in detail in section 5.4.3.

5.3.9 Visualisation

Nonverbal visual material can increase learner motivation and concentration. Visual materials can prepare students for learning by grabbing their attention with pictures or animations related to the subject matter. Visuals can illustrate, clarify, or supplement verbal information (Dwyer, 1985; Steinberg, 1984). Visuals can also provide an alternative mode for learning. In many real world settings, visual displays are the major, if not the sole means of transmitting information.

Although a considerable body of research demonstrates the effectiveness of visuals, the mere presence of graphics or specific display features (color, animation etc.) does not automatically guarantee better instruction. As the number of color-coded cues increases, the value of colour coding decreases (Dwyer and Lamberski, 1982-83).

There are different theories on knowledge representation in memory. One view states that all knowledge is stores according to its meaning in the form of logically complete statements, called propositions (Anderson and Bower, 1973; Anderson, 1980). In contrast another view proposes that a person's knowledge is stored in two separate systems, verbal and perceptual (Paivio, 1974). Studies of memory for picture strongly suggest that there is a different kind of memory for pictorial material than linguistic (Haber, 1970). However, none of these theories are proved to be correct.

Most studies show that if material is organised, people perceive, understand, and remember it better than if it is not organised. Instructional designers can support learning by directing learners' scanning and processing strategies (Duchastel, 1982; Waller, 1982; Reilley and Roach, 1986).

Visuals can aid comprehension and retention but they do so automatically. Colour can be motivating and increase attention (Heinich, Molenda and Russell, 1985), but it does not necessarily increase comprehension (Tullis, 1981). Likewise, realistic visuals are not necessarily more effective than simplified ones for all type of objectives (Heinich, Molenda and Russell, 1985; Dwyer,1985).

Appropriately designed, displays enhance learning. Designed without an understanding of how people derive meaning from them, displays can have no effect or can even interfere with learning. Display design synthesises the techniques of graphic designers, instructional designers, and psychologists' understanding of how visual information is processed and how it facilitates learning.

5.3.10 Cognitive Tools

Harper et al. (1996) describe how technology offer teachers the opportunity to shift from their traditional instructor role to mentor and co-learner. Teachers have to become more disposed to view learning as an active, creative, and socially interactive process and to view knowledge as something children must construct and less like something that can be transferred. Additionally software tools need to be available that supports the learner in construction of this knowledge. Don Norman's view of software for classrooms explain what cognitive tools are. ' The trick to teaching is to entice and motivate the student's excitement and interest in the topic, and then to give them the proper tools to reflect, to explore, compare, and contrast, to form the proper conceptual structures' (from Harper et al., 1996).

Jonassen and Reeves (1996, from Harper,1996) list the following to sum up their foundations of cognitive tools research:

- Cognitive Tools will have their greatest effectiveness when the are applied to Constructivist learning environments.

- Cognitive Tools empower learners to design their own representations of knowledge rather than absorbing knowledge representations preconceived by others.

- Cognitive Tools can be used to support the deep reflective thinking that is necessary for meaningful learning.

- Ideally, tasks or problems for the application of Cognitive Tools should be situated in realistic contexts with results that are personally meaningful for learners.

Learning Research and Development Center's (LDRC) research on Advanced Cognitive Tools for Learning, is about developing software and materials to support students learning critical inquiry skills by working together in teams to research and report on real scientific problems and learning skills for everyday life such as identifying an issue, organising their thoughts, forming hypotheses, planning a project, finding and evaluating new information, communicating their results, and working cooperatively with others.

Small groups of students can use these cognitive tools to access information databases about a simplified yet real scientific topic, test their hypotheses with simulations and hands-on activities, record their inquiry process in diagrams that make questions, hypotheses, data, and the relationships between them explicit and concrete, receive 'coaching' on the above activities from LRDC's software, and prepare reports and share their results within a classroom or with other network-based communities.

They claim that their software is designed to combine the best features of existing types of educational software, while avoiding their disadvantages. The advantages of the approach include:

- Software will be available to students, teachers, and parents on a variety of computers at different locations.
- The interfaces will be familiar to anyone comfortable with a 'Web Browser' such as Netscape.
- Software is designed for both collaborative and individual learning.
- It provides individualised coaching without requiring expensive computers.
- It provides interactive simulations along with intelligent guidance in their use.
- The databases and materials used by our software will be usable in other software as well.
- It will be easy to add other software tools to the existing tool-set.

Further they identify that reflection permits the work of abstraction and organisation of knowledge to go on without the mental load present during actual task performances. Hence it overcomes limitations of mental capacity and allows the abstractions that produce transfer. Cognitive tools stimulate reflection and so help transfer.

Diagnosis already has a notepad, that enable the student to take notes on the problem presented. It only has text editing, and no facilities to handle video, audio or images. If it does not have multimedia editing capabilities, the note book is not any more effective than a piece of paper. Making this notepad work and enabling the student to take notes from the problem may enhance *Diagnosis*.

5.4 Intelligent Features

For a semi-simulated environment like *Diagnosis*, providing some intelligent elements to help the student may enrich learning. Hypothetically, it would be of higher educational value if it combined the advantages from interactive multimedia and intelligent tutoring. Takeuchi and Otsuki (1988) identify the functions of Intelligent Tutoring Systems as mixed initiative dialogue and highly individualised learning environments. While *Diagnosis* is capable of interactive learning, it lacks personalisation or context dependent immediate feedback.

Capabilities of a typical Intelligent Tutoring System include:

- Understanding topics of dialogue and generating questions about the topics tailored to an individual student according to the student model and educational objectives provided by the author (Takeuchi et al., 1986, in Takeuchi A. and Otsuki S., 1988).

- When a student's response is incorrect, identifying the origin of the error and then updating the student model in order to coincide the model with the student's state of understanding

- Guiding erroneous students toward awareness of their mistakes

- Answering students' questions. In this context *why* reasoning takes an important role. The questions asking *why not* are answered by inferring by using the function of identifying error origins (Otsuki et al., 1985, from Takeuchi A. and Otsuki S., 1988).

In general, Intelligent Tutoring Systems have the following components: knowledge of teaching, student models, teaching expertise and a man-machine interface. In the case of *Diagnosis*, the knowledge of teaching is very complicated and hard to represent as rules or semantic networks. A simple alternative is to have an authoring model to construct the problems, a student model to personalise instruction and a proper interface for effective communication.

5.4.1 Student Modeling

Students do not all learn alike or at the same rate. Some instructional methods are better for some students. Good software will adapt to the learner, capitalising upon his/her talents, giving extra help where the student is weak, and providing motivators each student responds to. A tutor should consider the student's individual differences and emphasise conceptual understanding.

Modeling the student is one way of providing personalised instruction. Modeling the student involves tasks such as selection of the relevant learner attributes and their values, validation of the attributes and their values, selection of possible overlay values for the attributes, and keeping track of the overlay values and assessment of the initial learner knowledge.

However it should be noted that research suggest that learner control options may not always be a feasible method for individualising instruction (Carrier, Davidson and Williams, 1985). Contextual adoption can also be effective when controlled by the program rather than the learner (Anand and Ross, 1987).

A system built with a strong student model can greatly benefit the teaching process. This tutor tracks student skills along with a general acquisition factor, and

uses this information for topic selection, problem generation, problem presentation and dynamic feedback and scaffolding (Stern et al., 1996). User modeling is an important component of many systems that seek to adapt their behavior to users in order to interact more intelligently. It dynamically models the knowledge about the user and utilises it to direct system behavior in interaction.

A user model is a knowledge source in a system that contains explicit assumptions on all aspects of the user that may be relevant to the behavior of the system. User models are beneficial if the system seeks to adapt its behavior to individual users, or the system assumes responsibility for ensuring the success of user-system communication, or the class of potential system users, or the potential uses of the system are diverse.

User models have been of help in identifying potential obstacles in a user's plan (Allen, 1988), recognising when a user's query does not reflect the user's underlying goals (Pollack, 1986), tailoring responses according to the user's perception (McKeown, 1985) or knowledge (Paris, 1988) or correcting user misconceptions (McCoy, 1988).

A student model can be represented by maintaining a history of the student's performance, and/or by calculating the acquisition factor, which represents how quickly new information is learnt, and/or calculating retention factor, which says how well the material is recalled over time. Figure 5.5 pictures how a student model uses knowledge about students.

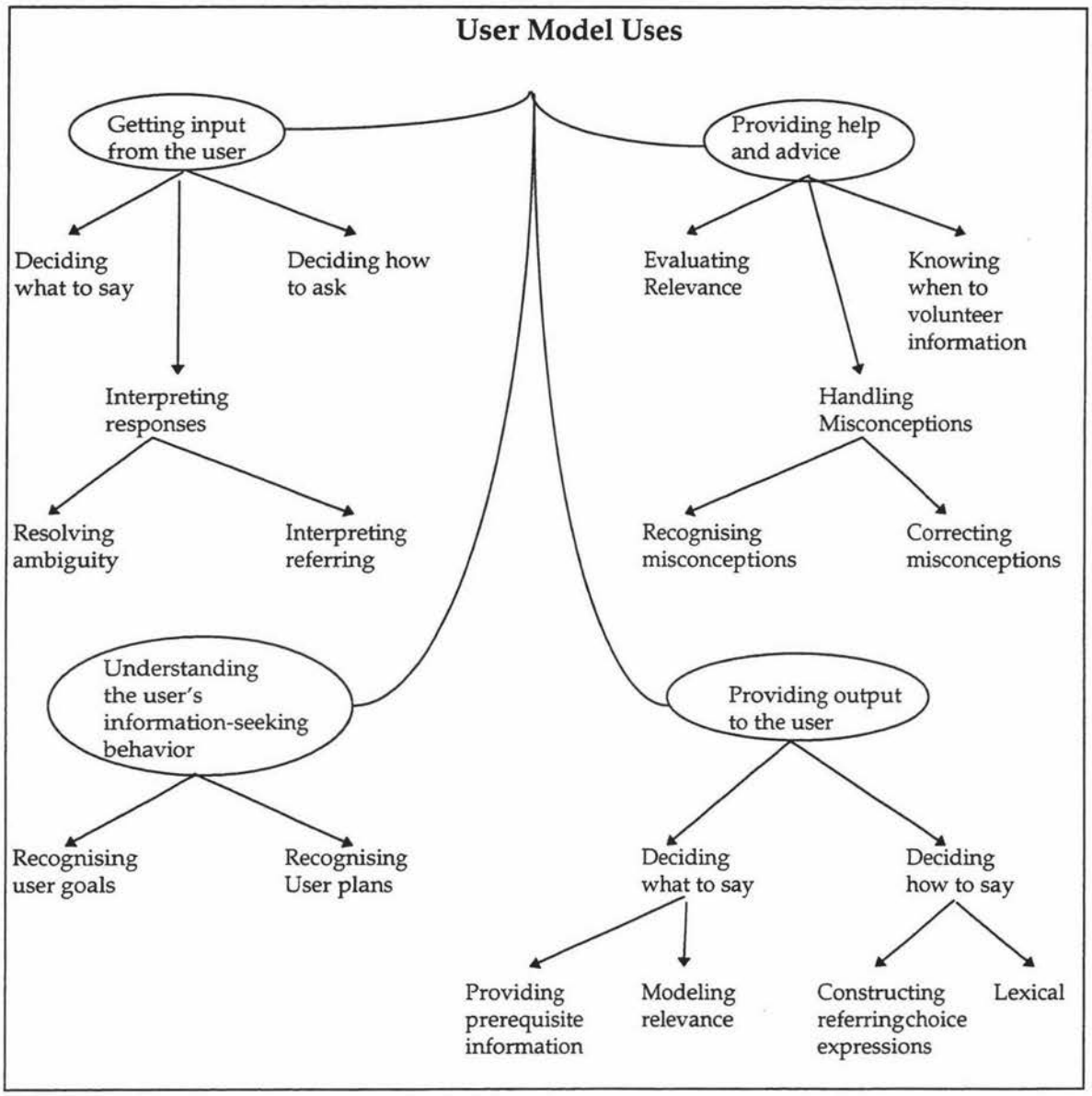


Figure 5.4 User Model Uses (Kass and Finn, 1996)

The student model is the part of the system which enables it to answer the questions: what can the student do? what does the student know about? what type of student is s/he? what has the student done?

A student model could be considered as a four tuple <*Procedural knowledge, knowledge, Individual traits, History*>. Procedural knowledge is the procedures for performing tasks. Conceptual knowledge is the descriptions of concepts.

Individual traits are the description of students. History is the description of significant events that have happened in the past. Conceptual knowledge and Procedural knowledge together describe what the student knows.

A detailed list of functions of student models is given by Self (1988):

Corrective function	Help eradicate bugs in student knowledge. It has sub-functions of Bug-identification, Direct-remediation, Indirect remediation, Counterexample, Solution tracing, Retrospection, Try again, Tactical withdrawal.
Elaborative function	Help extend what is described by correct but incomplete knowledge of the student.
Strategic function	Help initiate more significant changes in tutorial strategy than the tactical decisions of corrective and elaborative functions.
Diagnostic function	Help resolve the contents of the student's knowledge. It diagnoses the student model as well as the student.
Predictive function	Help determine the student's likely response to tutorial actions. It predicts the student's performance and learning.
Evaluative function	Help assess the student and system.

Based on these theoretical and empirical findings, a simple student model for *Diagnosis* is proposed in Chapter 6. Due to the complexity of the domain and media handling, the degree of personalisation is rather limited compare to some very good Intelligent Tutoring Systems. Further, the proposed model is created with common sense reasoning and should be proved to be efficient. *Scaffolding* is another techniques used in Intelligent Tutoring Systems to support learning. In this method, the student in provided with help in the beginning and as the learning progresses if the student perform well, the amount of help will be reduced. This idea is employed by the student model.

5.4.2 Feedback

Feedback is a message that is presented to a learner after s/he gives a response. According to behavioral psychologists the function of reinforcement is to increase the probability of a correct response and to decrease the probability of an incorrect one. Effective feedback is feedback that is specific to the nature of the response and understood by students. (Friend and Milojkovic, 1984; Hartley and Lovell, 1984; Steinberg, 1980; Stevens, Collins and Goldin, 1982). Feedback is most helpful when it is explicit. It should say exactly what you mean. This does not mean counter examples cannot be given as feedback. But they should not be too subtle for the student to perceive the intended meaning.

Feedback serves two functions - to inform and to motivate (Steinberg, 1984). Feedback can motivate students by encouraging them when the learning is difficult for them. The task of the instructor is to decide how much and what kind of information to provide. The nature of feedback can vary on a continuum from simple to complex.

Morrison et. al. (1995) mentions four types of feedback. Knowledge of response - the program simple tells the learner if their answer is right or wrong. Knowledge of correct response - learner typically receives a message saying 'No, the correct answer is'. Answer until Correct - telling the learner the response is incorrect and try again. Anticipated Wrong Answer - telling that the response was incorrect and provides information about the nature of the error.

Feedback may include information such as the number of allowed attempts at the question, the correct answer to the question, the nearly correct or common incorrect answer with the same further options, the last attempt provided the correct answer to the student and also specified a branch (Hedberg, 1985). Feedback about how well they are performing toward mastery and advice about how many examples they should look at can be very valuable in helping them achieve their goal (Tennyson, 1981).

The value of feedback may be related to a student's confidence in her response. If the student is quite sure about the correctness of the response, s/he probably gives

only cursory attention to feedback (Kulhavy, 1977). If the student is not sure about the correctness, feedback may be most valuable as corrective information.

The nature of the feedback should be commensurate with a learner's competence. Feedback should commensurate with developmental level (Steinberg, 1980). In some study, the feedback given to first grade students was ineffective because they did not understand how to use this kind of information. However, there is no guarantee that the student will accept the feedback.

Last but not least, the philosophy of 'decreasing intervention' (Liddle et al., 1996) should be employed. Feedback should gradually be phased out, lest the student become dependent on it and fail to develop her own strategies. The student should be able to apply her knowledge correctly, without feedback, in real life situations. Otherwise, feedback may become a crutch for getting the correct answers rather than a tool for reaching the instructional goals.

A tutor who understands why a student makes an error can do better job of remediating than one who knows only the error itself. Feedback given at the surface level may correct the response but not the reason for it. The underlying cause of error should be detected. Ideally, feedback should respond at a deeper level. This is yet to be achieved by most Intelligent Tutoring Systems. Most of them cannot help the student overcome basic misconceptions.

Accuracy is important in a tutoring system; wrong feedback can have a negative impact on learning. A human tutor will certainly be more accurate than any machine based tutor (Feifer, 1983) because, they understand the context in which the student is making the error and would probably be able reason about the error more precisely. While identifying the reason behind the bug is such a novel and effective idea, it cannot be efficiently achieved in *Diagnosis* as there is no knowledge representation. However, the guided authoring systems compensate for this by providing for the author to differentiate between types of errors while constructing the problem.

5.4.2.1 Timing of feedback

Should the feedback in multimedia instruction be provided immediately or delayed? The answer is not clear and probably it depends on the mentality of the student. But, it seems delayed feedback may be more effective for higher cognitive tasks than immediate feedback. It is possible that delaying feedback allows additional time for reflection, which may in turn facilitate learning challenging material.

Results from feedback studies suggest that immediate feedback is more effective than delayed feedback (Kulik and Kulik, 1988) and that most forms of feedback are more effective than no feedback (Smith, 1988). Generally, an instructional designer views feedback as an opportunity to reinforce, elaborate clarify, or even magnify learning (Park and Hannafin, 1993). On her studies on feedback Mory (1992) observed feedback acts to correct errors and that the effect is more powerful if the learner feel confident that the incorrect response is correct. So, a strategy is to have the learner declare their level of confidence at key points during instruction and then tailor feedback to acknowledge the professed level of confidence.

Some students prefer delayed feedback. Immediate feedback does not uniformly give rise to enhance learning (Anderson, Boyle, Corbett and Lewis, 1989; Kulik and Kulik, 1988 in Corbett and Anderson, 1989). Human tutors do not always intervene immediately when a student makes a mistake (Fox, 1988; Lepper and Chaby, 1989 in Corbett and Anderson, 1989). There is no definite answer for the question of timing of error correction.

A key principle in the behavioral approach to instruction is that feedback must be immediate. It is reasonable if feedback is equated with reinforcement. If it is not the role of feedback to strengthen responses why should it be immediate? R.C.Anderson et al. (1972), demonstrated that feedback is essential and if it is poor it may decrease learning (Kulhavy and Anderson, 1972). They note that a test strengthens response tendencies. After a delay, a student forgets the responses s/he gave on the initial test. Error tendencies interfere with his/her learning when feedback is immediate. Consequently there is more error prevention in immediate feedback. Also, students who got delayed feedback takes more time than those who got immediate feedback.

J.R Anderson quotes Skinner 'the importance of immediate feedback to skill acquisition is one of the best documented facts in psychology'. He says when feedback is delayed student may learn incorrect procedure. In a complex problem solving task students spend more time trying to solve problems when feedback is delayed than when it is immediate. Also, immediate feedback avoids frustration. With delayed feedback, some students get demoralised.

Even though both of them argue that immediate feedback is the best, their explanations are different. It is because of the difference in the subject matter and difference in the nature of the task. The current version of *Diagnosis* has a delayed feedback of knowledge of correct response type. Regardless of the type, amount and depth of mistakes student make, it gives the same typical feedback already built by the problem builder. Considering the above arguments, may be it is good to have delayed feedback, but personalised feedback would make it a more effective system.

5.4.3 Learner control

An intuitively appealing argument for learner control in instruction is that the students will be more motivated if allowed to control their own learning. Learner control could alleviate boredom, frustration and anxiety because it enables students to skip over materials they not prepared to study. Learner control will maintain attention longer, involving students more deeply and perhaps gives students greater insights.

Before 1977, in most cases, students failed to employ adequate review strategies and they did not know how to manage their time. The motivational benefits of learner control were not accompanied by better performances. Since 1977, its been proposed that the psychological processes in learning and individual differences in learning skills and strategies (such as lack of clearly defined learning objectives (Rimiszowski, 1981), use of naive or erroneous learning strategies (Steinberg, 1977), lack of metacognitive skills (Rigney, 1978; Allen and Merrill, 1985) or lack of information they needed about their learning progress to make meaningful

decisions (Tennyson and Rothen, 1979) should be taken into account to prove learner control is advantageous.

In multimedia instruction, control refers to the selection of content and sequence, but may also include the full range of learner preferences, strategies, and processes used by the learner. Giving the learner control may increase motivation to learn (Santiago and Okey, 1990; Steinberg, 1977), but it does not necessarily increase achievement and may increase time spent learning (Santiago and Okey, 1990).

Learners who are generally high achievers or who are knowledgeable about an area of study can benefit from a high degree of learner control (Borsook, 1991; Gay, 1986; Hannafin and Colamaio, 1987), or the naive or uninformed learners require structure, interaction, and feedback to perform optimally (Barsook, 1991; Carrier and Jonassen, 1988; Cartwright, 1998).

High effectiveness is achieved by maintaining strong control over the students' behavior. It is apparent that it is not possible to create a system that allows students more control and yet still gets them through the exercises quickly. Corbett and Anderson (1989) call it the *Dilemma of Freedom and Educational Efficiency.*

Learner control with advisement is an effort to take advantage of the best features of both adaptive computer control and learner control, using a mathematical or statistical formula to diagnose a learner's understanding and to prescribe the sequence of instruction. Adaptive advisement and Branching adaptive are two such methods.

Management decisions like the amount of time the learner is allowed to view each display, the difficulty level of exercises, the number of exercises, the sequence of topics, conditions for advancing through lessons can be designed under computer control.

The motivational aspects of learner control are not necessarily accompanied by better performance. Learner characteristics such as prior knowledge, motivation, metacognitive skills, and individual differences play an important role in influencing the effectiveness of learner control. Learner control may have been unsuccessful for some subjects because they did not have clearly formed objectives

(Romiszowski, 1981) or because they employed naive or erroneous learning strategies (Steinberg, 1977), or they may have lacked metacognitive skills (Allen and Merrill, 1985; Rigney, 1978) or lacked the information they needed about their learning progress to make meaningful decisions about how to manage learning (Tennyson and Rothen, 1979). The key issue for Merrill (1984) is not whether to provide learner control but how to maximise the student's ability to use the learner control that is available.

Allen and Merrill (1985) suggest that students with metacognitive skills should be given freedom to control their own learning. And students who have some knowledge but do not know how to apply it, should be guided to select an appropriate strategy. And for students who are totally deficient in this skills, the system should embed the strategies.

It appears that shared control of computer tools would be better than total control by either the learner or the computer alone. In *Diagnosis,* learner control is limited, in the sense that a student has a number of menu options to select from. But student explorations are not necessarily systematic. S/he has the freedom to choose from where to start. For example, s/he could look at the leaves first and then go on to flowers or vice versa. This amount of learner control seems to be appropriate for the type of instruction *Diagnosis* provides.

5.5 Appropriate elements for *Diagnosis*

Based on the theoretical work discussed in the previous sections, the following elements for alterations have been proposed to enhance *Diagnosis*. It just describes 'what' changes are appropriate. How it is going to be implemented is discussed in detail in Chapter 6.

The major deficiencies in *Diagnosis* are lack of individualisation and lack of context sensitive immediate feedback. The proposed changes are centered around these issues. For these purposes, a student model is created and depending on the student's performance, the model is updated. The presentation of problem is based on the skills of the student assessed by the student model.

According to Anderson et al. (1989), from their work on *cognitive tutors*, it is believed that presenting problems, starting from the least difficult ones and then increasing the level of difficulty is the best method of presentation for a novice user.

In plant pathology, to be exact in all the subjects, the level of difficulty is slightly subjective to the student, if not fully subjective. S/he may be thoroughly familiar with a particular plant or the effects of particular pathogen in several plants. However, it is not impossible to divide the problems into a hierarchy with crop type depending on level of complexity, in a reasonable way. Actually, what it means by 'complexity' here is how fast the student could figure out the clue to move towards the right diagnostic path. The clue may be obviously stated in the problem, it may be a direct implication of a statement of the problem or it may be a subtle one which needs some thought and linking bits of information together.

Another feature that could direct the student to think in the right direction is providing expert consultation. *Diagnosis* already has options such as 'Ask Grower' and 'Ask Neighbour'. Both these options are very helpful in exploring the problem space. There is no help to explore the solution space, unless otherwise the student direct himself/herself into the solution space by correctly identifying the clues.

This 'Ask Expert' option could provide a generalised expert advice on the problem presented. It would be helpful for the 'poor' students in moving forward in the right direction, relatively quickly. It will be available only in the practice version, not in the assessment version. When this option is selected, there should be a multimedia or single media hint displayed on the screen, explaining the possible pathogens, prevention methods and exceptions from an expert's point of view.

The current version of *Diagnosis* only has delayed feedback. This feedback gives the correct diagnosis for the problem, which is already prescribed by the problem builder. It does not consider the type of error, number of errors made or the depth of error. There is zero degree of personalisation. The presentation of the problem or the feedback does not adapt itself to the student. Even though it is argued that delayed feedback gives time for the students to reflect on their errors, it is

suggested that personalised immediate feedback would be helpful for the novice user.

In the proposed model, in addition to the delayed feedback, an immediate feedback is also suggested. This feedback is personalised in the sense that it will depend on the performance of the student. The better the performance the lesser the help and the poorer the performance the more specific the help. Immediate feedback ranges from a general statement, through an expert's opinion to a specific hint.

The proposed changes do not include context sensitive help. But, an attempt is made to achieve it by handling different types of errors in different ways. The more serious conceptual errors and the less serious mistakes are treated differently and given different feedback.

Most of the above changes are incorporated in the authoring system or Builder module. The 'Diagnosis' module only has to be programmed to accommodate these changes. Since the system is not represented as rules, the feedback for each case has to be built by the problem constructor. A guided authoring system is suggested to make it easier for the builder to think about all the possibilities and effectively construct the problem.

5.6 Discussion

The process of learning involves construction of meaning by perception and taking final responsibility for accepting or rejecting the subject. Florin (1994) uses *virtual towns* and *intellectual amusement parks* to visualise several dimensions of the learning environment with a single metaphor.

We feel that we have understood something or learnt something, when we hear this 'logical click' within ourselves. From that particular moment, we have our own visualisation of the concept we were learning. It may differ from someone else's visualisation of the same concept. But, that doesn't matter. All that matters is if we had understood it correctly. Continuous adoption of content and representation, aid provided in visualising the information presented are

important factors in visualisation skill development or making this 'logical click' happen sooner.

To improve knowledge acquisition, tools such as hypothesis formation and problem solving techniques which are supported by 'quick feedback' should be provided. It was not computer or learner control of the tool but rather the tool in conjunction with feedback that positively affected performance.

With the technology and theories available right now, computers cannot answer just any question posed by a learner. There is a limit for the area that a computer can handle at a time. We cannot assume it could model all the students. The only thing that could be done is providing relatively better systems, that could model the majority of the students and answer most of the questions.

Wang (1996) pictures virtual classrooms as large number of heterogeneous, intelligent educational agents distributed over large computer and telecommunication networks. Agent refers to a software-based computer system that enjoys:

- autonomy (agents operate without direct intervention of humans)
- social ability (communicate with other agents)
- reactivity (agents perceive their environment and respond in a timely fashion to changes that occur in it)
- pro-activity (agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviors by taking the initiative).

A computer-based environment should have a balance between 'discovery learning', 'personal exploration' and 'systematic instruction and guidance'. Content and form and interactivity go together in important learning experiences. Reflection becomes the glue that takes advantage of the experiences we can provide. A compelling goal for multimedia designers is to find ways to combine action and reflection in an effective learning cycle.

It is worth mentioning the questions posed by Bednar et al. (1995). Answers to these questions may be used as guidelines for designing a better system. Is critical thinking the goal of all learning? Do the contexts in which learning is to be applied relate to the nature of the learning experience? Are there contexts in which it is appropriate to apply traditional instructional development models and others in which it is not? Does a distinction exists between training and education such that a training environment is more appropriate than a school for instruction based on traditional instruction design principles? At what level of schooling is critical thinking a reasonable goal? Is it reasonable to differentiate levels of learning?

Chapter 6

Diagnosis : From Game to ITS

6.1 Introduction

Diagnosis for crop protection is a puzzle type multimedia game for diagnosing plant diseases. As it is described, 'successful diagnosis of a crop problem can be as much as an art as a science, relying on the correct interpretation of subtle clues' (Diagnosis Manual, 1997).

It has two modules called Builder and Player. Builder module is to build and modify problems and Player module is to present the problems to the player. The author has to construct the problem, by describing the situation and explaining the defects in the plant parts, field observations, the lab procedures to be performed in order to identify the cause of the problem, cost involved in performing a lab test, and a delayed feedback message which consist of the points that should have been noted in the process of diagnosing the problem and the optimal path in solving the problem. Text, video, picture or audio can be added to describe the situation. The player is presented with the description of a situation. S/he has to search for the clues and make deductions in order to identify the problem.

The player has to interrogate the system to diagnose the problem. S/he can closely look at the plant parts, soil, insects available and/or ask the grower or the neighbour to fully understand the situation. S/he can collect items such as plant parts, insects, soil and take them to the lab for further studies in order to confirm the tentative conclusion or in order to eliminate possibilities. Some of the lab procedures are not free. The player will be informed of the cost involved before proceeding with the procedure. Accumulated cost will be shown.

The player has to make an initial diagnosis and recommendation before s/he goes to the lab. After more exploration and lab visits the student is supposed to type the detailed diagnosis, justification and remedial action, which will be assessed by the human tutor sometime later. Once the lab procedures are over, the player has to make the final diagnosis in order to see the feedback. The feedback consist of the correct answer, the important clues that should have been noted and an optimal path to the solution. Then the player can select another problem to solve or exit. They can also quit in the middle of a session and save it to be continued later.

Educational aims of Diagnosis are claimed to be

- To reinforce the processes undertaken in diagnosing a crop problem
- To experience and practise diagnostic situations
- To appreciate the pre-disposing factors which may lead to disease or pest problems
- To gain a greater understanding of the importance of pathogens and pests in crop management

6.2 Shortcomings in *Diagnosis*

The shortcomings of *Diagnosis* are:

- It does not help most of the players in achieving their educational goals, even though it has a good set of goals. Because it does not provide any guidance or help during the diagnosis session, it only help a small range of students who are good at the subject.
- Diagnosis does not adapt its style of presentation to the standard of the player. It has a standard way of presentation, regardless of the player knowledge in the area.
- There is no time limitation (i.e. limitation in number of tries) in interrogating the problem. A player can go on for ever looking at the problem, if they are misled in the wrong direction. It does not provide any assistance for those who are lost in the solution space, in order to get them back in the correct solution path. It does not have a reference book, which is available in a practical situation, for the player to consult when they need to clarify something.
- It does not provide any guidance in constructing a problem. A better understanding of the builder would lead to better construction and so better presentation of the problem.

To educationally enhance Diagnosis, these problems should be addressed.

6.3 Proposed alterations to enhance Diagnosis

According to Reder and Klatzky (1994), situated learning is based on the following claims (Anderson et al., 1996) :

- Action is grounded in the concrete situation in which it occurs
- Knowledge does not transfer between tasks
- Training by abstraction is of little use, real learning occurs in authentic situations
- Instruction needs to be done in a highly sociable environment.

Diagnosis is in a well suited area to be developed as a situated learning game. In that way, the problem will be more similar to the real environment and it will be easier for the player to apply the knowledge to real the environment without much difficulty. Besides, if it is implemented like other situated learning packages such as *Investigating Lake Illuka* (University of Wollongong, 1993), where the physical environment could be explored without going through menu, it will be easier for the player to explore the situation just by clicking in the area to be observed closely [For more details see Chapter 7].

However, considering the resources available and simplicity of the current system -especially building problems- it was decided that it is better to make use of multimedia elements rather than changing it into a totally multimedia package.

6.4 Detailed Design Suggestions

The following sections describe the proposed alterations to Diagnosis. An algorithmic description of the changes are given in Appendix F. A few screen designs are given in Appendix G.

6.4.1 Error types

A player can make two types of errors. One is *ErrorType1*, which has no justification to be the correct thing to do in that particular situation described to the player. And the other is *ErrorType2* which can be justified to be correct in certain circumstances even though it is not correct in that particular scenario. In other words, errors of *ErrorType1* are more serious errors than those of *ErrorType2*.

For example, in a problem caused by *cause1*, the symptoms are similar to the symptoms caused by *cause2*. However, there is a subtle clue *clue1* which eliminates the possibility of *cause2*. If the player overlooks this clue and indicates that the problem is due to *cause2*, it is an error of type *ErrorType2*. But, in a problem caused by *cause1*, if the player identifies that the problem is due to *cause2*, where *cause2* could not at all be mistaken for *cause1* then it is of *ErrorType1*.

Players who make a error of type 2 (*ErrorType2*) should be reminded of the clue they are overlooking and players who make an error of *ErrorType1* should be presented with more direct and precise information about the situation - they have to be directed to the correct path.

The possible errors of *ErrorType1* and *ErrorType2* in a problem are marked when the problem is created. They are also counted while a player is presented with a problem, to make the assessment about the player. Errors of *ErrorType1* and *ErrorType2* are treated differently and different feedback is provided in each case.

Anticipating all the possible errors in advance a player could make is very difficult. But, the proposed Builder Module helps the author by modularising the construction process. It prompt the author to reflect on one portion of the problem at a time and make it easier for them to consider the possible errors and mistakes as soon as they describe that portion of the problem.

For example, assume the author enters the following initial scenario information:

> *" You have been called in to advise on a home garden problem. You find yourself in the garden of Mrs Christine Warner, local body politician and keen camellia grower. It's the middle of the spring flowering season. Over 100 varieties of varying shapes and sizes adorn the property in what should be a kaleidoscope of attractive blooms.*
>
> *However, not all is well in this paradise. At least ninety percent of all the flowers present are showing dark brown splotches on the tissue with many blooms having turned completely brown!*

Mrs Warner accompanies you on your investigation, eager to learn what the problem may be. She prides herself on knowing camellia culture and is very puzzled by the symptoms.

As you walk around the garden you are impressed by how beautiful it is. Camellias are obviously the most popular plant and there are many varieties present that you have never seen before. You are struck by the way the colours are arranged, and how other flowering and foliage plants are used to soften the edges of the plantings and vary the visual pattern. "

At this point the author will be asked to mark any important clues in the text. For which the author highlights the sentence *"At least ninety percent of all the flowers present are showing dark brown splotches on the tissue with many blooms having turned completely brown"*. Now, the Builder Module will display a window saying "What would you say to a player who doesn't understand the problem?". The reply to this question will be given as feedback whenever the player misses this clue and eventually submits a wrong diagnosis. Also they are asked to type in expert's opinion on such cases. The reply *"The symptoms seems to be like that of flower blight or Botrytis cinerea, wind damage, cold injury or even natural senescence. Flower blight symptoms may appear first in the centre or base of a petal as opposed to the edges. Botrytis cinerea infection can be common in older spent blooms"* is used as feedback during presentation of the problem. The same questions are repeated for misleading clues.

6.4.2 Feedback

In the proposed Diagnosis, there will be three types of feedback, namely *immediate direct feedback, immediate indirect feedback* and *delayed feedback.*

Immediate direct feedback is feedback which directly indicates the problem and what should be done to avoid it. For example, consider a scenario with a crop that is affected by some insects in the soil, but the crop looks unhealthy. If the player is only looking at the parts of the crop and not looking at the soil (probably guessing that something is not right with the crop itself), and not looking at the soil for a number of tries, then they will be given a message saying "Why don't you look at

the soil?". It directly ask them to do the right thing. This type of feedback is provided for the players in the category *poor* when they are given a guided presentation.

Immediate indirect feedback is an expert's opinion on the current state of the problem. For example, in a scenario with a crop that has early falling fruits, the real reason for that may be *cause1*. Say the player tries to establish that the reason is something else which is possibly true under some other condition. A feedback message ' *In my experience I've seen this type of problems could be caused by cause2, and in some other times due to the presence of object1 caused by cause3*' would serve as indirect immediate feedback. With this message, the student has a few possible causes, but has to eliminate some using the clues given in order to arrive at the correct diagnosis. This type of feedback is provided for the players in the category *average* when they are given a guided presentation.

Delayed feedback is given after the player types in the final diagnosis, remedy and justification. This consists of a list of things the player should have looked at or done in order to go in the right direction, if they haven't done so and also the optimal path in arriving at the solution, in case the player has not taken the optimal path.

For example, if the optimal path for a solution is looking at the root and taking it to the lab for a nutrient procedure, and if the student had looked at the leaves, branch, fruit and soil before looking at the root and has checked the root for pathogens, s/he will be given a delayed feedback message which is personalised from the list constructed by the builder. A typical message for this case might be ' You don't need to look at leaf, branch, fruit or soil for *reason1*. You should have looked at the root first for *reason2*. And the pathogen test is unnecessary. Instead you should have run a nutrient check'. For these purposes a trace of the student solution path has to be maintained.

6.4.3 Player Model

The Player Model has to assess the player to be able to present the problems accordingly. One way of getting an impression about the student is to determine what they know. For this purpose a list of subtopics in Plant Science, which are necessary for the player to solve the problems should be displayed. From the list the player can tick on the familiar subjects. Depending on the number of familiar subjects, the knowledge level of the student will be determined.

However, quantifying such issues is not always easy and the result will not be perfect for everyone. Firstly, the student may claim that s/he knows something when s/he does not really know it and vice versa. Secondly, not all the subtopics are equally important in solving a problem. For a particular problem the players need to know some topics but not others. For example, say we are listing the subjects *subject1, subject2, subject3,* in the initial list and the player checks on each subject which means s/he is familiar with all three subjects. The player is presented with a problem caused by *cause1,* which only needs knowledge in *subject1.* Subjects such as *subject2, subject3* are not helpful in this situation. Thirdly, people who are good at theory are not necessarily good at applying the theory. Likewise someone who has no idea of the subject may have very good practical experience and hence may be able to solve the problem without theoretical knowledge. In cases like this, counting the number of known subjects is not a good way to judge the student.

Hence it is proposed to test the student with a few multiple choice questions, which cover the subject areas to be understood in order to deal with the problems. And depending on the range of correct answer the standard of the student could be determined. For the standard test in the proposed Diagnosis there will be 10 questions. Even though a larger number of questions would greatly help in clear classification of student type, it may be too time consuming. If we reduce the number of questions too much, it will be difficult to test the player on various subject areas. That is why 10 questions are taken to be about the right number. A plant pathology teacher helped in deciding the initial range for each category of player as follows:

8-10	good
5-7	average
0-4	poor

This method does sound slightly better than the previous one because it eliminates the first problem in the previous method.

There will be a standard test, which could be modified by the author. The builder of the problems who has the clear idea about the types of problems to be solved, should be able to create the test. To achieve this the Builder module should be modified to accommodate the creation, modification and alteration of the test. And also the number of correct answers in determining the player's standard should be decided by the builder.

If the player wants to skip the test, they should be able to do so. Then they can select the problem type, for example, a tough problem of the *strawberry* crop type caused by environment, and will be presented with the same (if available). It is also possible to turn the test session off, in case the tutor wants to assess the student by presenting some standard problems.

6.4.4 Previous session details

When the player finishes a session, the session details such as the number of errors s/he made and the problems attempted are saved. From the number of errors (both *ErrorType1* and *ErrorType2*) the standard of the player could be decided, and comparing it to the initial standard would give the progress of the player, which could be used with the feedback. The problems attempted are marked to prevent presenting the same ones again.

6.4.5 Classification of Problems

When creating a problem, it is classified in three ways. These are *level of difficulty, crop type* and *problem cause.*

The difficulty classification,

- easy
- medium
- hard

is adapted from the Equation Solving Tutor (Pilletier et al, 1993). The aim is to select the problems to present in the best way to educate the player. For example, it is good to present a poor player with an easy problem in the first place, so that they can solve it and get a psychological boost about continuing the diagnosis. And when they are familiar with solving *easy* problems they could be presented with *medium* or *hard* problems to advance their learning.

The crop type classification allows the player to choose a particular crop, if they are interested in various diseases in that crop, and also for the technical reason of presenting scenarios according to the type of crop. For example, a banana tree cannot have branches and so the menu should be different from that of a peach tree. The following is the classification provided in *Diagnosis* Version 2.1.

- Apples
- Bananas
- Berry Crops
- Brassicas
- Bulb Crops
- Cereal
- Cotton
- Curcubit
- Flowers
- Legume
- Lettuce
- Peanuts
- Pineapples
- Potatoes
- Root Crops
- Sugar Cane
- Strawberries
- Tree Crops
- General Crop

The cause of problem classification is a list of broad classification of reasons for the problems. For example,

- Problems caused by air pathogens
- Problems caused by root pathogens
- Nutrient problems
- Environmental problems
- Other problems

Even though is looks like the student could guess the diagnosis from the classification selected, it is not that easy. The initial diagnosis will be more refined and detailed so that the player has to understand the situation and decide the cause or must be lucky enough to make a correct wild guess.

This classification is to select similar problems to present in case the player did not make a correct diagnosis for a particular problem. It also will be helpful to have a further level of classification within a cause, to be more precise in selecting a similar problem.

6.4.6 Style of Presentation

Depending on the standard of the player determined, the presentation will be varied. For a poor player the presentation will be a guided one with direct help. For an average student the presentation will be guided with indirect help such as expert opinion on the situation, which is more general and needs some transformation to be applied to the situation in hand. For a good student the problem is presented and they are allowed to explore and diagnose the situation without any help.

Basically, the approach is to start presenting those who are not good at the subjects with easy problems and if they do well, move onto harder problems. And for those who are good at the subject start presenting with hard problems and move to harder problems. In case they don't do well, easier problems should be presented. It should be noted that this is a reasonable generalisation, for there are numerous possibilities in between, such as a good student doing well in the beginning and becoming worse later or a poor student who progresses well and then drops back to the previous level.

6.4.6.1 Presentation for poor player

The poor player is presented with easy problems in the beginning. The important clues in the text will be highlighted with a beep sound if they don't take the right action. Also, whenever they do something that is not even remotely appropriate in finding the problem or eliminating the possibilities, then they will be given a warning or reminder saying 'Do you really think this is necessary?'.

If they manage to correctly identify the cause from the initial diagnosis list in the first attempt, and continue do the same for a number of problems then they are progressed to attempt harder problems. If they don't progress well, they could be presented with medium problems. In each case the number of errors of *Error Type1* and *ErrorType2* will be counted to determine the progress rate. A detailed presentation is given in the algorithm in Appendix F.

6.4.6.2 Presentation for average player

Average students are presented the problem with indirect feedback. If they get the initial diagnosis correct with a limited number of operations, they are allowed to try medium problems with indirect help and then to easy problems without help and hard problems with indirect help and finally to hard problems without help. Otherwise they will be going through step by step progress.

If they tend to make many errors, after a certain stage they have to be provided with direct help.

6.4.6.3 Presentation for good player

Good players are presented with easy problems without help in the first place and then medium without help and finally hard problems without help. However, the number of errors they make are counted and after a certain number of errors of *Error Type1* they are asked if they want help and, if so, the presentation will be guided with direct or indirect feedback.

6.4.7 Initial diagnosis and remedy

The initial diagnosis consists of a limited list of possibilities. Even though it is better to have multiple levels of sub options to lead towards a refined, specific diagnosis, it is very difficult to classify the causes that way, because some problems have a combination of causes. So, it was decided to go only up to two levels, preferably.

The initial diagnosis list includes a number of possible causes together with some random causes. This is created as a standard list and additions and deletions could be made when building the problems. A typical initial diagnosis list is given below and would probably include one more level of options for each item in the list.

- Nutrient Problem
- Environmental Problem
- Fungal Disease
- Insect or mite Problem
- Virus Problem
- Spray Damage
- Other

Initial remedy is also a list of possible remedies for the problem. The player has to select from this list. The author will be constructing the list according to the nature of the problem.

- Do Nothing
- Apply Fungicides
- Apply Insecticides
- Apply Herbicides
- Remove Plants
- Improve Drainage
- Improve Irrigation
- Buy healthier seed/propagating material
- Improve Fertiliser Regime
- Use Biological Control
- Other

An immediate feedback message will be given for both initial diagnosis and remedy. For those who don't get it right, a few more chances are given.

6.4.8 Reference Book

Just as in a real situation the player is given access to a reference book, which consists of information about plants, insects, soil etc. The access can be denied if the tutor sets the reference book access to *off*. The author should be able to make additions, deletions and modifications to entries.

6.4.9 Guided Builder

As mentioned earlier, it will be easier for the author of the problem if s/he is guided through the module in an organised manner. S/he could be given a menu hierarchy that gives an overview of the things happening when building a problem, similar to that available in *Investigating Lake Illuka*. Preferably, the modules that have already been completed should be highlighted to avoid repetition.

The new Builder will have eight modules. Information about each module is given in Table 6.1. The author will be encouraged to follow the prescribed order. It is also possible to display only the menu for the module being build and until it is finished or the author wants to stop, s/he cannot move onto the next module.

Module	Name	Purposes
1	*Strategy and Player model*	The author is asked if s/he want to create/modify/turn off the test. The strategy behind the presentation is displayed and s/he is allowed to create/modify/turn off the test. Then s/he is asked to enter/alter the number of correct answers for the test in order to determine the standard of the player. And also s/he has to enter the number of errors of *Error Type 1 and Error Type 2* to decide the progress of the player.
2	*Problem description*	General introduction to the problem. Initial scenario description. Add/modify/delete audio/video/picture/text. Mark the clues in the text. Feedback messages for the clue - both direct help and expert opinion.
3	*Field Observations*	Observations to be made in the field. Mark plant parts to be observed. Mark important clues. Feedback messages for the clue - both direct help and expert opinion. Mark misleading clues. Feedback messages for the clue - both direct help and expert opinion - if they have overlooked.
4	*Initial Diagnosis*	A list of possible diagnoses - including the correct one and a few look alikes. Feedback messages. Same for the initial remedial action.
5	*Lab procedures*	Important procedures to be carried out and the feedback messages.
6	*Debriefing Session*	Points that should have noted while progressing with problem solving. Reasons why they are important. Each node in the path should be described separately in order to tailor personalised debriefing. Optimal solution path.
7	*Reference Book*	Creation /addition/deletion/ modification of reference book. Turning on/off the access to reference book.
8	*Menu Structure*	It is the display of the menu structure in detail. The modules that have been completed in constructing a problem are displayed in different colour for the purpose of giving instant visualisation of the construction process to the author.

Figure 6.1 Components of Builder module

6.5 Do the proposed changes make *Diagnosis* intelligent?

Intelligence is not about having the typical components of ITSs. It lies in the package's ability to adapt its presentation according to the player's learning pattern, to motivate her and make her actively involve in learning. The proposed *Diagnosis* offer a higher degree of flexibility compared to the older version. A player can opt to try it out without any help if s/he doesn't prefer immediate feedback. At the same time s/he can try it with immediate feedback if s/he wants to.

Even though *Diagnosis* is a menu driven game, the menu selection is totally the choice of the player. So, it provides a feeling of learner control to the player, while maintaining a balance in the control by limiting the options available to the learner.

Begg and Hogg(1989) list the following as the major functions performed by an Intelligent Authoring System.

- Create and modify a knowledge network consisting of either facts/concepts or procedures/functions.
- Specify and modify error/diagnostic rules and teaching rules linked to the information in the knowledge network.
- Specify and modify the format of user input, program output and screen displays.
- Identify inconsistencies and incompleteness in the knowledge network, error/diagnostic rules, and teaching rules.

Diagnosis Builder Module, which is the authoring system for *Diagnosis*, does not have all the above mentioned functions. It does not create or modify knowledge network for reasons specified in section 6.6. Rather it creates or modifies knowledge about a particular problem.

Builder Module does create rules for teaching and error detection. It could alter the quantitative values behind the strategy in order to cater for a particular curriculum, but it cannot figure out the inconsistencies between different problems, because it does not share a general knowledge network. It has several small scenarios stored and does not compare them for consistency. Since knowledge is not represented as rules or expressions, comparing the scenarios for consistency is far beyond reach for the moment.

From this view, Builder Module is not an Intelligent Authoring System. But, it can be considered as a well organised authoring system. After all one of the big plus points about Diagnosis is being able to create new problems and Builder Module serves its purpose.

A Prototypical ITS can be considered to engage in three distinct although interrelated instructional processes (Duchastel,1988):

- Diagnosing the state of the student's knowledge
- Providing instruction
- Motivating the student

If we consider the proposed Diagnosis, it does have all the above processes. It diagnoses the player's knowledge by giving her a test. The Player Type (Good, Average, Poor) is determined from the results of this test.

The teaching strategy is to present the student with a predetermined number of scenarios (called a series) and to evaluate each of them. Depending on how well s/he had performed in the series, the next series of questions will be selected.

Learning pattern of a player can be any one of the 3*16! curves. For example, consider an *Average player* trying out Diagnosis. Let the number of questions in a series be 5. Let the learning pattern be the one in Figure 6.2

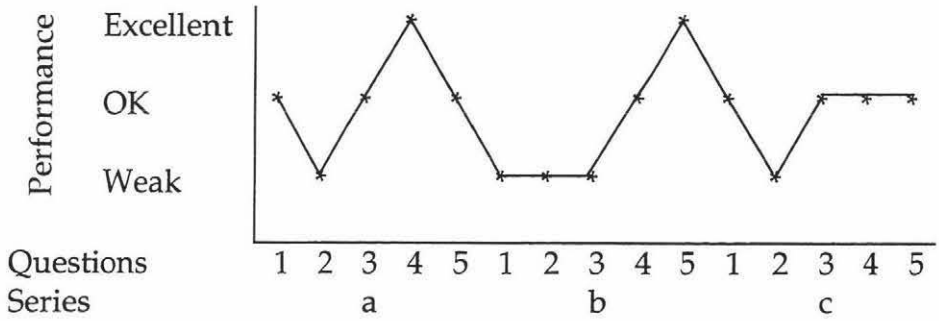


Figure 6.2 Pattern of user Performance in each problem

Figure 6.3 shows the pattern of overall performance in the series. According to the strategy, results of series a, b are used to select the type of questions for series c. And results of series b, c are used to select the type of questions for series d and so on.

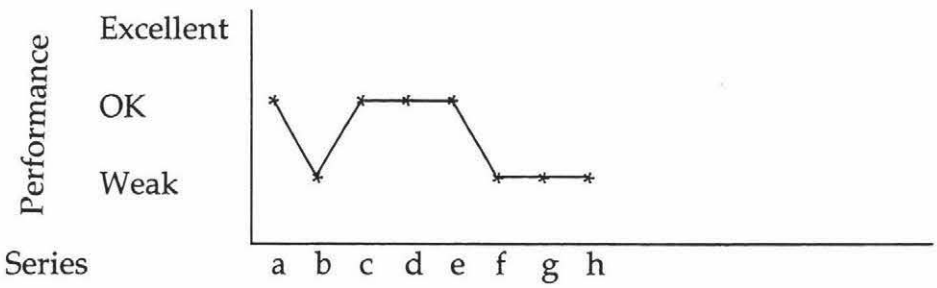


Figure 6.3 Pattern of user Performance in each series of problems

It will be good to have a trace of the whole learning curve of a particular player, for a the purpose of selection of type of problems in the next series. However, for the sake of simplicity of the player model, only the pattern of previous two series are considered to select the next series.

Below tables show the selection questions for each type of Player depending on the results of previous series.

Initial Values for *Poor Player*

Problem Type	Easy
Problem Cause	1
Help Level	1

Series 1	Series 2	Problem Type	Problem Cause	Help Level
Good	Good	Medium	Different	2
Good	Average	Easy	Different	2
Good	Poor	Easy	Different	1
Average	Good	Medium	Same	2
Average	Average	Easy	Different	1
Average	Poor	Easy	Same	1
Poor	Good	Medium	Same	1
Poor	Average	Easy	Different	1
Poor	Poor	Easy	Same	1

Initial Values for *Average Player*

Problem Type	Medium
Problem Cause	1
Help Level	2

Series 1	Series 2	Problem Type	Problem Cause	Help Level
Good	Good	Hard	Different	3
Good	Average	Medium	Different	2
Good	Poor	Medium	Same	2
Average	Good	Hard	Different	2
Average	Average	Medium	Different	2
Average	Poor	Medium	Same	1
Poor	Good	Medium	Different	2
Poor	Average	Medium	Same	1
Poor	Poor	Easy	Different	1

Initial Values for *Good Player*

Problem Type	Hard
Problem Cause	1
Help Level	3

Series 1	Series 2	Problem Type	Problem Cause	Help Level
Good	Good	Hard	Different	3
Good	Average	Hard	Different	2
Good	Poor	Hard	Same	2
Average	Good	Medium	Different	3
Average	Average	Medium	Same	3
Average	Poor	Medium	Different	3
Poor	Good	Medium	Different	3
Poor	Average	Medium	Different	2
Poor	Poor	Medium	Different	1

Besides having a player model, Diagnosis provides instruction to the player by giving different types of feedback. It also motivate the player with its multimedia elements. So, according to the criteria of Duchastel (1988), Proposed Diagnosis could be justified as an Intelligent Game.

6.6 Discussion

The proposed changes make both the *Builder Module* and the *Player Module* slightly more complex to be programmed and to be maintained. However, the resulting presentation and guidance will be effective enough to justify this complexity. That is the reason for dividing *Builder Module* into sub-modules. In that way, the author will be guided through the construction of the problem. Further, it provides more concentration and better visualisation of the module the author is dealing with at any particular moment.

It is really hard to fully computerise the system, to eliminate the need for a human tutor in assessing the player. With the current technology, there is no way to

effectively process the final diagnosis, to include remedial action and justification typed in natural language.

The subject area of diagnosis is so vast and mostly non-procedural that it is almost impossible to represent the knowledge as rules or expressions or any other similar form. The knowledge network of plant pathology is so interconnected and complicated that clearly defining the domain is extremely difficult. Besides, it is a growing subject. Everyday we face new problems caused by new reasons or a combination of new and existing reasons.

The issue of having a time limitation or limited number of attempts before initial diagnosis to prevent wasting a lot time wandering around the field, is very critical. Some may want to go to the initial scenario once in a while to make sure they are on the right track. So, it is left as it is and the player can check this information as many times as s/he wants.

For the moment, selection of multimedia element for presentation is left up to the builder. It is difficult to provide help on this matter, as every problem is different and the selection depends on the type of problem. An alternative is to have a list of nature of problem, let the builder decide the nature of the problem to be built, and then provide standard help. For example, the problems that can be identified by looking at the plant should have more pictures or video of the plant and the ones that have to be figured out by examination should have more microscopic figures. This cannot be done simply, because the multimedia usage to spread out through the presentation to have a balance and to be subtle enough not to give obvious clues.

Another critical decision is the selection of problem type in presentation. One practical reason for this selection is there is not enough problems. And another reason is how to handle rare situations like running out of problems or a student who could not even handle a single easy problem with help. Decisions have been made 'democratically' even though it may be not helpful for all the students. It could not be better for even human teachers cannot adapt their teaching style to every student in the class.

Chapter 7

Conclusion and Further Work

7.1 Introduction

This chapter summarises the research and points out its findings. Further it analyses the possibilities for future work on *Diagnosis*.

7.2 About the research.....

This research is about finding an appropriate way to synthesise intelligent tutoring and multimedia for good teaching and learning environments. To apply the theoretical work, a multimedia tutoring system *Diagnosis* was selected and some changes were proposed to make it intelligent and effective.

A review of literature on AI and multimedia for educational systems revealed the potential of intelligent multimedia systems as an efficient and effective alternative for human tutoring. However, there is no rigid rules to combine these and the efficiency depends on the optimum systhesis of strengths of both AI and multimedia.

Software review is essential for developing and improving the system. The framework for evaluation developed during this research was found to be very useful in highlighting the strengths and weaknesses of the pieces of software that were evaluated.

Multimedia systems are technologically rich and have diverse types of media compare to conventional media. While this is an advantage of multimedia over single media, another critical problem is how do we allocate media in multimedia system. In single media systems we do not need to worry about this issue.

There are different opinions in addressing this media allocation problem. Some argue that tried and tested methods of conventional system design development can be applied to these new media whereas others identify new design methodologies are needed to develop multimedia systems. These ideas are hypothetically valid. But, linking theory into practice seems difficult. Most of the practically implemented multimedia systems have their own development methodology depending on the nature of the system. From the analysis we could

reach the conclusion that even though we can have a general methodology for multimedia system development, we do have to deal with the problems - especially media allocation problems - according to the type of domain and target population.

Educational systems have two equally important faces namely technical issues and pedagogical issues. Improvement of a piece of software is not only technologically broadening the software, it is also pedagogically making it more usable. Learning involves construction of meaning by perception and taking final responsibility for accepting or rejecting the material being presented. Continuous adaptation of content and representation is an important factor in improving knowledge acquisition.

There are several intelligent features and human factors that have to be considered in developing a good multimedia learning system. It should have a balance between 'discovery learning' and 'personal exploration'. Content and form and interactivity should go together making learning experiences interesting. A compelling goal for multimedia designers is to find ways to combine action and reflection in an effective learning cycle. Listing the objectives of the system and finding appropriate features may help in achieving these educational objectives.

The proposed changes to *Diagnosis* are based on theories and common sense reasoning. However, that is not enough to prove that these changes will make *Diagnosis* more learnable. Some empirical evidence is also needed for that purpose. Since the changes have not been fully implemented it is impossible to test the changes and arrive at some statistical figures to justify that it is better than the previous version.

Being able to construct problems is one of the important features of *Diagnosis*. Building a problem is not just about describing the scenario of the problem and its solution. It is a creative activity. Problems have to be constructed properly in order to reflect reality and to achieve the goal of being able to apply the knowledge in the real world.

There are different things to focus on while constructing a problem. If a problem is constructed without much multimedia explanation, that is without any images or

video audio clips, it does not differ essentially from a text book problem. A problem should be built in a way so that it would take advantage of the multimedia capabilities of *Diagnosis* and at the same time not to overdo it by including an excessive amount of images, audio and video.

From personal experience with *Diagnosis*, if the scenario is not built subtly, it is easier for a good problem solver to figure out the problem, even if s/he has very little knowledge in the subject. For example, if a problem is built without any red-herrings and it only contains images for the optimal route, then the diagnosis of the problem is very obvious. Especially with the proposed change of selecting from a list of preliminary list of reasons, it is easier for a smarter student to make an intelligent guess. While it is good to be able to make educated guesses in *Diagnosis*, in the real world application, there is no assurance that there will only be problems without any misdirecting clues. So, it is important that the problem, especially the harder problems should have enough red-herring to be close imitations of real world problems.

For the above mentioned purposes, *Diagnosis* needs more author support documents and a guided intelligent authoring system. As a result, the authoring system needs more changes than the presentation system. An alternative method to construct problems is to have a team of software developers and educationalist who are specialised in this area and let them construct the problems, rather than letting the tutor build the problem. *Life Learn Series I* employs this approach.

7.3 Further Work: What else could be done to *Diagnosis*?

Diagnosis is basically a game, which serves its educational purpose with the aid of multimedia. But considering the similarity to guided discovery systems for Ecology such as Investigating Lake Illuka, in subjects, appropriateness, effectiveness and success, it appears that *Diagnosis* would be more effective if it could shift to situated learning while holding the plus points of gaming.

The current version of *Diagnosis* is simple and cost effective. In case it has to be designed in situated learning paradigm, the cost and the complexity of program structure should be justified by the educational achievements.

7.3.1 Situated Learning

Situated Learning (Greeno, 1989; Brown, Collins and Duguid, 1989) is a stance holding that enquiries into learning and cognition must take serious account of social interaction and physical activity (from Rochelle, 1994). It is the notion of learning knowledge and skills in contexts that reflect the way the knowledge will be useful in real life. It can incorporate situations from everyday life to the most theoretical endeavors (Collins, 1991).

Breach between learning and use may be a product of the structure and practice of our educational system. It assume a separation between knowing and doing, treating knowledge as an integral, self-sufficient substance theoretically independent of the situations in which it is learned and used. Recent investigations of learning challenge this separating of what is learned from how it is learned and used. The activity in which the knowledge is developed and deployed is an integral part of what is learned.

Miller and Gildea's work on vocabulary teaching has shown that the constituent parts of knowledge, index the world and so are inextricably a product of the activity and situations in which they are produced. So a concept will continually evolve with new occasion of use, because new situations, negotiations, and activities inevitably recast it in a new, more densely textured form. This would also appear to be true of well-defined abstract concepts (from Brown et al., 1995).

Cognitive Science, Artificial Intelligence, and Intelligent Tutoring Systems tend to assume too readily that knowledge, the goal of learning, resides in the head in explicit concepts and reified, abstractions (Lave, 1988). But, it is becoming increasingly clear that abstract knowledge is not only very difficult to learn, it is also not particularly transferable. What is learned in the classroom too often cannot be carried beyond the classroom door. On the other hand, less abstract, situation-specific knowledge appears to be much more easily acquired.

Knowledge is a product of the activity, context, and culture in which it is developed and used, and must be evaluated as such. Therefore, this knowledge must be learned in the actual work setting (Brown et al., 1989) or a highly realistic or virtual surrogate of the actual work environment (McLennan, 1991) or an anchoring context such as a video program (Bradford et al., 1988; Bradford et al., 1990) (from CTGV, 1993).

According to Brown et al.(1989), the components of situated learning include:

- apprenticeship
- coaching
- repeated practice
- articulation
- reflection
- collaboration
- technology
- stories.

Collins et al. (1991) presented a modified version of this model which includes four types of features (See Figure 7.1)

- content: domain knowledge, heuristic strategies, control strategies and learning strategies.
- methods: modeling, coaching, scaffolding and fading, articulating, reflection, and exploration.
- sequence: increasing complexity, increasing diversity, and global before local skills.
- sociology: situated learning, the culture of expert practice, intrinsic motivation, exploiting cooperation, and exploiting competition.

Figure 7.1 Cognitive Apprenticeship Model (Collins et al., 1991)

A unifying concept emerging from situated learning research is "communities of practice" - the idea that learning constituted through the sharing of purposeful, patterned activity (Lave and Wegner, 1989). This idea stresses "practice" and "community" equally. Knowledge is seen as practical capability for doing and making. Meaning is seen as a construction of social unit that shares stake in a common situation. As a consequence, learning is seen as a capability for increased participation in communally experienced situations - a dual affair of constructing identity and constructing understanding (Wegner, 1990, from Rochelle, 1994).

The important feature of situated learning that would be useful for *Diagnosis* is learning situation-specific knowledge which leads to higher transferability of this knowledge.

7.3.2 Why Situated learning ?

From a futuristic view, it seems that *Diagnosis* could be developed as a interactive multimedia learning environment. However, it cannot be accepted without a reasonable ground to base this view on. Questions such as why Situated Learning? why not any other learning paradigm? needs to be answered in order to fortify this argument.

Situated Learning and Constructivism are based on different ideas. Situated learning emphasises that knowledge is maintained in the external social world. Constructivism argues that knowledge resides in an individual's internal state, perhaps unknowable to anyone else. However both share the general philosophical position that knowledge cannot be decomposed or *'decontectualised'* for purposes of either research or instruction (Anderson J.R., 1996).

Situated learning claims	*Constructivism Claims*
Action is grounded in the concrete situation in which it occurs	Knowledge cannot be instructed (transmitted) by a teacher, it can only be constructed by the learner.
Knowledge does not transfer between tasks	Knowledge cannot be represented symbolically
Training by abstraction is of little use; real learning occurs in 'authentic situations	Knowledge can only be communicated in complex learning situations
Instruction need to be done in highly social environment	It is not possible to apply standard evaluations to assess learning

From constructivist point of view, Anderson et al. argue that while cognition is partly context-dependent, it is also context-independent; while concrete instruction helps, abstract instruction also helps, while some performances benefit from training in a social context, others do not. The indirect implication here is that cognition also depend on the nature of the subject matter being learned. For a very practical subject area like plant pathology, context-dependent environments with concrete instructions would be more helpful. A very interesting critique about the

above mentioned claims on situated learning by Anderson et al. is written by Greeno (1997). Each and every claim has been proved to be incorrect with the aid of counter examples.

It is good to base the development of educational software on some profound learning theory. But, it is not necessary to be bounded to a single theory. What we need is a better learning system. It is not incorrect to synthesise ideas from different learning theories, even though they are from non-identical schools, as far as they do not contradict each other.

Achieving the educational goals is more important than what theory is behind the system. It is appropriate to mention the five educational goals for technology supported environments, derived from a Chinese adage.

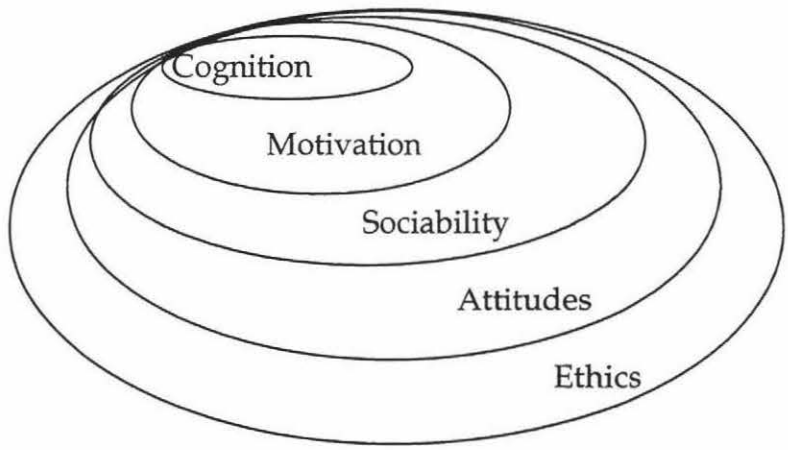


Figure 7.2 Five Educational Goals (Chan, 1996)

From a goal oriented point of view, increasing cognition, motivation, and sociability, and developing attitudes and ethics are the aims of the learning environment.

The idea of further refining *Diagnosis* is not just about shifting the learning theory from constructivism to situated learning. It is about incorporating the valuable characteristics of situated leaning while retaining the values of constructivism. The idea is to lean more towards situated cognition than it is now, to make it an improved learning system. We cannot just present a student with authentic situations and expect them to perform well, just because the context is real. Providing help, which is not really available all the time in the real life situations, is important for better learning for novice students.

If a pedagogy based upon ideas about situated cognition is to succeed it must not only be based upon true ideas. Indeed, it is nearly irrelevant whether those ideas are true or false. Situated pedagogy must actually be better than traditional instruction, and not just under ideal conditions. It must survive the wear and tear of everyday use and be better under those conditions especially (Tripp, 1993).

The benefits of situated learning observed by Collins (1991) are as follows.

- Students learn conditions for applying knowledge
- Situations foster inventions
- Students see the implication of knowledge.
- Context structure knowledge appropriate to its uses.

All the above benefits of situated learning could be achieved for a non-procedural, less- abstract domain like plant pathology.

7.3.4 Combining Situated learning and gaming

Computer games are one of the most attractive and entertaining among all the types of software. Its interactivity, motivation and creativity are very important factors in its success. Employing intelligence to games is one of the major research areas in computers. Since Garry Kasparov was defeated by Deep Blue in May 1997, smart games have gained more popularity.

By nature, manually diagnosing a disease itself is a game, which involves careful observations, interpretation of clues and red-herrings and determining the actual cause of the problem. Also, prescribing something to prevent the problem involves a similar process.

Picture *Diagnosis* as a system that present the student with a authentic looking situations. A system that let the student zoom in and out, with just a click to examine plant parts, iconic menu system with hierarchy of menu for guidance may seem imaginative. But it is achievable. Similar idea is featured in *Life Learn Series I* (See Appendix A for details).

The challenge is to create interesting, realistic contexts that encouraged the active construction of knowledge by learners. It is not only about creating real situations. But also about providing help and in some cases providing immediate feedback, or expert advise, reference to relevant books are all part of this system.

7.3.5 Problems to be addressed

The biggest problem facing this dream version of *Diagnosis* is authoring system for constructing problems. How many multimedia experts are out there with considerable amount of theoretical knowledge on education? This version has to be sophisticated and easy to author at the same time.

Guiding the author to construct an intelligent game of diagnosis is the greatest challenge. Teachers report that one of the biggest challenges in knowing when students really need guidance versus when students are struggling in a constructive way with a problem or issue (CTGV, 1993). Providing help at the right time should be considered while constructing the problem.

One of the major concerns about situating instruction in specific contexts is that students' understanding and application of these concepts will stay welded to the context. What the situated cognition people have done is to emphasise the crucial difference between a representation of a situation and the situation itself (McLennon, 1993).

Finding some way to create an intelligent authoring system that can guide the author to focus and reflect on different issues as the problem being built would be a very interesting area to be explored.

References

Adam, J. A. (1993). *Applications, implications*. IEEE Spectrum. Vol. 30, No. 3. March 1993.

Adams, E., Carswell, L., Ellis, E., Hall, P., Kumar, A., Meyer, J. and Motil, J. (1996). *Interactive Multimedia Pedagogies*. Report of the working group on Interactive Multimedia Pedagogy.

Alessi, S. M. and Trollip, S. R. (1991). *Computer-Based Instruction - Methods and Development*. Prentice Hall Inc.

Alexander, S. and Hedberg, J. (1994). *Evaluating technology-based learning: Which model?*. In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Ambron, S. (1988). *Interactive Multimedia*. In S. Ambron and K. Hooper (Eds.) Interactive multimedia: visions of multimedia for developers, editors, and information providers. Microsoft Press.

Anderson J. R. (1983). *Acquisition of Proof Skills in Geometry*. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Machine Learning - An AI Approach. Tioga Publishing Company.

Anderson J. R., Reder, L. M. and Simon, H. A. (1996). *Applications and Misapplications of Cognitive Psychology to Mathematics Education I.*
http://www.sands.psy.cmu.edu/personal/ja/misapplied.html

Anderson, T. G. (1988). *The idea of a Learning Laboratory*. In S. Ambron and K. Hooper (Eds.), Interactive multimedia: visions of multimedia for developers, editors, and information providers. Microsoft Press.

Andrews, D. H. and Goodson, L. A. (1995). *A Comparative Analysis of Models of Instructional Design*. In G. A. Anglin (Eds.), Instructional Technology - Past, Present and Future. Libraries Unlimited Inc, 1995.

ANGLE. (1989). University of Pittsburg, PA, USA.

Atkinson, M. A. and Burton, J. S. (1991). *Measuring the Effectiveness of a Microcomputer Simulation*. Journal of Computer-Based Instruction, 1991, 18(2), 63-65.

Baldwin, D. (1996). *Three years' experience with gateway labs*. Conference proceeding. In Proceedings of Integrating Technology into Computer Science Education, Barcelona, Spain.

Barker, P. (1994). *Designing Interactive Learning*. In T. De Jong and L. Sarti (Eds.), Design and Production of multimedia and simulation-based learning material. Dordrecht, Netherlands.

Barker, P. (1995). *Evaluating a Model of Learning Design.* World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Barker, P. (1996). *Electronic Books and their Potential for Interactive Learning.* In M. Brouwer-Jonse and T. L. Harrington (Eds.), Human-Machine Communications for Educational Systems Design. NATO Series, Springer-Verlag.

Bar-On, E. and Or-Bach, R. (1988). *Explanation-Based Learning in Intelligent Tutoring Systems.* In P. Ercolli and R. Lewis (Eds.), Artificial Intelligence Tools in Education. Elsvier Science Publishers B.V. (North-Holland).

Beattie, K. (1994). *How to avoid inadequate evaluation of software for learning.* In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Bednar, A.K., Cunningham, D., Duffy, T.M. and Perry, J.D. (1995). *Theory into Practice, How do we link?.* In G. A. Anglin (Eds.), Instructional Technology -Past, Present and Future. Libraries Unlimited Inc, 1995.

Begg, I.M. and Hogg, I. (1989). *Authoring Systems for ICAI.* In G. Kearsley (Ed.) Artificial Intelligence and Instruction - Applications and Methods. Addison-Wesley Publishing Company.

Benamou, N. and Celentano, A. (1995). *Production of Interactive Multimedia Courseware with Mathesis.* In T. deJong and L. Sarti (Eds.), Design and Production of multimedia and simulation-based learning material. Dordrecht, Netherlands.

Berryman, S.E. (1996). Designing Effective Learning Environments: Cognitive Apprenticeship Models.
Http://www.ilt.columbia.edu/k12/livetext/docs/berry1.html

Bielenberg, D. R. (1995). *Designing Interactive Stories for Learning.* World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Booch, G. (1991). *Object Oriented Design with applications.* The Benjamin/Cummings Publishing Company, Inc.

Braswell, R. (1994). *The Top Five Reasons Not to Use Multimedia.* In (Eds.) Reed, W.M., Burton, J.K. and Liu M.) Multimedia and Megachange- New Roles for Educational Computing. The Haworth Press, Inc.

Brown, J.S. (1990). *Toward a New Epistemology for Learning.* In Intelligent Tutoring Systems: At the crossroad of Artificial Intelligence and Education. (Eds) Claude Frasson, Gilles Gauthier Ablex Publishing Corporation.

Brown, J.S., Collins, A. and Duguid, P. (1995). *Situated Cognition and the Culture of Learning*. Institute for Learning Technologies, 1995
http://www.ilt.columbia.edu/ilt/papers/JohnBrown.html

Brown, P. and Hickey, M. (1990). *Validation - Cost Effective External Evaluation*. Australian Journal of Educational Technology, 1990, 6(2), 92-98.

Carbonell, J. R. (1970). *AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction*. IEEE Transactions on Man-Machine Systems, Vol 11, No 4. December 1970.

Carswell,L. and D. B. (1996). *An adventure game approach to multimedia distance education*. Conference proceeding. Integrating Technology into Computer Science Education Barcelona, Spain.

Certificate in Remote Area Servicing *(RAS)*. (1996). Department of Employment, Education, Training and Youth Affairs, Australia.

Chan, T. (1996). *Learning Companion Systems*. Journal of Artificial Intelligence in Education, Vol. 7, No. 2.

Collins, A. (1991). *Cognitive Apprenticeship and Instructional Technology*. Educational Values and Cognitive Instruction: Implications for reform (Eds Lorna Idol, Beau Fly Jones). Lawrence Erlbaum Associates, Publishers, 1991.

Constantine, L.L. (1995). Essential Modelling: use cases for user interface. ACM Interactions,. Vol.2, No.2. April 1995.

Corbett, A.T. and Anderson, J.R. (1989). *Feedback Timing and Student Control in the Lisp Intelligent Tutoring System.* In (Eds.) Bierman D., Breuker, J. And Sandberg, J.) Artificial Intelligence and Education - Synthesis and Reflection. 4th International Conference on AI and Education.

Corderoy, R.M., Harper, B.M. and Hedberg, J.G. (1993). *Simulating algal bloom in a lake: An interactive multimedia implementation*. Australian Journal of Educational Technology 1993.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet9/su93p115.html

Crockford, D. (1988). Stand by for Fun. In Ambron, S and Hooper, K (eds) *Interactive multimedia: visions of multimedia for developers, editors, and information providers*. Microsoft Press.

CTGV (Cognition and Technology Group at Vanderbilt). (1993*). Anchored Instruction and Situated Cognition Revisited*. Educational Technology. March 1993.

de Jong, T. and Sarti, L. (1994*). Trends in the Design and Production of Computer Based Learning Material*. In T. deJong and L. Sarti (Eds.), Design and Production of multimedia and simulation-based learning material. Dordrecht, Netherlands.

Diagnosis Manual. (1997). http://www.diagnosis.co.nz/

diSessa, A. A. (1987). *Artificial Worlds and Real Experience.* Artificial Intelligence and Education Volume One. M. Y. Robert W. Lawler. Norwood, NJ, Ablex Publishing.

Doulton, A. (1986). *What Will An Interactive Video Look Like?.* Artificial Intelligence for Society. K. S. Gill, John Wiley & Sons Ltd.

Duchastel, P.C. (1988). *Models for AI in Education and Training.* In P. Ercolli and R. Lewis (Eds.), Artificial Intelligence Tools in Education. Elsvier Science Publishers B.V. (North-Holland).

Dymalla A., Kent, B. and Russel, A. (1994?). *Towards Successful Complete Based Learning - the complete process.*

Espinosa, R. And Baggett, P. A. (1996). *Design for Accessing Multimedia Information Using Cohesion.* In S. Frasson, C. Gautheir and A. Lesgold (Eds.), Intelligent Tutoring Systems, ITS'96 Proceedings. Spinger-Verlag.

Exploring the Nardoo. (1995). Interactive Multimedia Unit, Faculty of Education, University of Wollongong in collaboration with the NSW Department of Land and Water Conservation, Australia.

Feifer, R.G. (1983). *Sherlock: An Intelligent Tutoring System for Teaching People How to Learn.* In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Machine Learning - An AI Approach. Tioga Publishing Company.

Focus. (1997). *For better or for worse - The Multimedia Reign.* In Gagne, S. (Ed.), Focus, April 1997, Montreal, Canada.

Foxon, M. (1989). *Evaluation of training and development programs: A review of literature,* Australian Journal of Educational Technology, 1989, 5(2), 89-104.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet5/su89p89.html

Foxon, M. (1992). *In defense of post-course evaluations: Going beyond the smile sheet.* Australian Journal of Educational Technology, 1992, 8(1), 1-12.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet8/wi92p1.html

Friedhoff, R. M. and Benson, W. (1983). The Second Computer Revolution VISUALIZATION. W.H.Freeman and Company, New York.

Fritze, P. (1994). *A Visual mapping approach to evaluation of multimedia learning materials.* In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Furht, B. (1994). *Multimedia Systems: An Overview.* IEEE Multimedia Spring 1994.

Gaver, W.W. (1995). *Class, You're not Making Enough Noise! The case for Sound-Effects in Educational Software.* In M. Brouwer-Jonse and T. L. Harrington (Eds.), Human-Machine Communications for Educational Systems Design. NATO Series, Springer-Verlag.

Gibaud, O. (1995). *An Example of Learning by Doing in a Systemic Context with Extended Micro-World Concept*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Gilnert, E.P. and Blattner M.M. (1996). *Multimodal Interaction*. Guest Editor's Introduction, IEEE, Spring 1996.

Godfrey, R. B. (1995). *New Wine in Old Bottles: Multimedia Design Methodology*. ASCILITE '95 - Learning With Technology.

Goodman, B.A. (1995). *Multimedia Explanations for Intelligent Training Systems*. Intelligent Multimedia Interfaces. AAAI Press/ The MIT Press.

Greeno. (1997). *A Critique of Situated Cognition*. In Selden, A. and Selden, J. (Eds.), Research Sampler.
Http://www.maa.org/t_andl/sampler/rs_2add.html

Harper, B. M. and Hedberg, J.G. (1995). *Exploration Information Landscapes Through Networks*. AUC95.
http://130.130.88.100/conferences/95I01.html

Harper, B.M., Hedberg, J. G., Wright, R.J. and Corderoy, R.M. (1995a). *Multimedia Reporting in Science Problem Solving*. Australian Journal of Educational Technology 11(2): 23-37.

Harper, B.M., Hedberg, J. G., Wright, R.J. and Corderoy, R.M. (1995b).
Multimedia Reporting in Science Problem Solving. Australian Journal of Educational Technology 1995.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet11/su95p23.html

Harper, B.M., Hedberg, J. G., Wright, R.J. and Corderoy, R.M. (1996). *Using Cognitive Tools in Interactive Multimedia*.
http://www.itu.arts.su.edu.au/AUC%C4/Harper.html

Harvey, B. (1994). *Reflections from a slick surface: Design issues in interactive multimedia*. In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Hedberg, J.G. (1985). *Designing Interactive Videodisc Learning Materials*. Australian Journal of Educational Technology 1985.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet1/sum85p24.html

Hedberg, J.G., Harper, B., Brown, C. And Corderoy, R. (1994). Ed. (1994). *Exploring user interfaces to improve learning outcomes*. In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Hedberg, J.G. and Harper, B. (1995). *Exploring Interactive Multimedia Information Landscapes*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95 Graz, Austria, Association for the advancement of Computing in Education (AACE).

Henderson, J.V. (1992). *Interactive Multimedia for Experiential Learning: A Consideration of Dewey's Educational Philosophy*. ITTE 1992. Proceedings of the 2nd International Conference on IT for Training and Education, University of Queensland, Australia.

Henderson, L. (1993). *Interactive Multimedia and culturally appropriate ways of learning*. In C. Latchem, J. Williamson and L. Henderson-Lancett (Eds.), Interactive Multimedia Practice & Promises. Kogan Page Ltd.

Henderson, M. and Patching, B. (1995). *Multimedia Interactivity: Relating Cognitive Processing to Click-drag Activities*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95 Graz, Austria, Association for the advancement of Computing in Education (AACE).

Hodges, M.E. and Sansett, R.M. (1993a). *Navigation: Design for a Visual Learning Environment*. In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Hodges, M.E. and Sansett, R.M. (1993b). *Designing for Quality*. In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Hodges, M.E. and Sansett, R.M. (1993c). *Design for the new Medium*. In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Hodges, M.E. and Sansett, R.M. (1993d). *A New Literacy*. In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Hodges, M.E. and Sansett, R.M. (1993e). *New Views of Education*. In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Hoogeveen, M. (1995). *Towards a New Multimedia Paradigm: is Multimedia Assisted Instruction Really Effective?*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA'95 Graz, Austria, Association for the advancement of Computing in Education (AACE).

Hooper, K. (1988). *Multimedia in Education*. In Ambron, S and Hooper, K (eds) Interactive multimedia: visions of multimedia for developers, editors, and information providers. Microsoft Press.

Hoppe, H.U., Okamoto, T., Petrushin, V. A. and Leopold, R. (1995). *Distributed Multimedia Learning Environments - a Challenge for Computer Science and Educational Technology*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Investigating Lake Illuka. (1993). Interactive Multimedia Unit, Faculty of Education, University of Wollongong, Australia.

Jerrams-Smith, J. (1995). *Combining Multimedia, Hypermedia and Artificial Intelligence to Support Four Aspects of Learning*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Kaplan, R. and Rock, D. (1995). *New Directions for Intelligent Tutoring*. AI Expert. February 1995.

Kass, R. and Finn, T. (1991). *General User Modeling: A Facility to Support Intelligent Interaction*. In J. W. Sullivan and W. Tyler (Eds.), Intelligent User Interfaces. ACM Press Frontier Series.

Kemp, R. H. (1995). *Ph.D Project: Designing Interactive Learning Environments*. Computer Science Department, Massey University, Palmerston North.

Kifer, E. (1995). *Evaluation A general view*. In G. A. Anglin (Eds.), Instructional Technology -Past, Present and Future. Libraries Unlimited Inc, 1995.

Kopec, D., Brody, M., Shi, C.C. and Wood, C. (1983). *Towards an Intelligent Tutoring System with Application to Sexually Transmitted Diseases*. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Machine Learning - An AI Approach. Tioga Publishing Company.

Lanza, A. and Roselli, T. (1989). *An Evaluation of a CBI System for Computer Programming Language*, Journal of Computer-Based Instruction, 1989, 16(4), 126-128.

Latchem, C., Williamson, J. and Henderson-Lancett, L. (1993). *IMM: An Overview. Interactive Multimedia*. In C. Latchem, J. Williamson and L. Henderson-Lancett (Eds.), Interactive Multimedia Practice & Promises. Kogan Page Ltd.

Laurillard, D. (1994). *The role of formative evaluation in the progress of multimedia*. In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Lawler, R. W. (1987). *Learning Environments: Now, Then, and Someday*. Artificial Intelligence and Education Volume One. M. Y. Robert W. Lawler. Norwood, NJ, Ablex Publishing.
Legree P.J. and Gillis, P.D. (1991). *Product Effectiveness Criteria for Intelligent Tutoring Systems*. Journal of Computer-Based Instruction, 18(2),57-62.

Liddle, J., Leitch, R. and Brown, K. (1996). *Generic Approaches in Developing Practical Intelligent Industrial Training Systems.* In S. Frasson, C. Gautheir and A. Lesgold (Eds.), Intelligent Tutoring Systems, ITS'96 Proceedings. Spinger-Verlag.

Litchfield, A. (1994). *Interface Communication Management: A user centered multimedia design model.* Second Interactive Multimedia Symposium, Perth.

Looms, P. O. (1993). *Interactive Multimedia in Education.* In C. Latchem, J. Williamson and L. Henderson-Lancett (Eds.), Interactive Multimedia Practice & Promises. Kogan Page Ltd.

Maddux, C.D., Johnson, D.L. and Willis, J.W. (1992). *Educational Computing, learning with tomorrow's technologies.* 1992.

Marchionini, G. (1995). *The cost of Educational Technology: A framework for Assessing Change.* World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95 Graz, Austria, Association for the advancement of Computing in Education (AACE).

Maybury, M.T. (1993). Intelligent Multimedia Interfaces. AAAI Press/ The MIT Press.

McKendree, J. (1989). *Effective feedback content for tutoring complex skills.* MCC Technical Report Number ACT-HI-259-89.

McLennan, H. (1993). *Evaluation in a Situated Learning Environment.* Educational Technology, March 1993.

McNamara, S.E. (1994). *Multimedia But.... (Boffins using Thingummybobs or Beginners Understanding Technology).*

McNaught, C., Whithear, K. and Browning, G. (1994). *The role of evaluation in curriculum design and innovation: A case study of a computer-based approach to teaching veterinary systematic bacteriology and mycology.* In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Morrison, G.R., Ross, S. M. and O'Dell, J. K. (1995). *Applications of Research to the Design of Computer-Based Instruction.* In G. A. Anglin (Eds.), Instructional Technology -Past, Present and Future. Libraries Unlimited Inc, 1995.

Neal, J.G. and Shapiro, S.C. (1994). *Knowledge-based Multimedia Systems.* In (Eds.) Bugord, J.F.K. Multimedia Systems. ACM Press.

Nouwens, F. and Robinson, P. (1991). *Evaluation and the development of quality learning materials.* Australian Journal of Educational Technology, 1991, 7(2).
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet7/su91p93.html/

Ohlsson, S. (1987). *Some Principles of Intelligent Tutoring.* Artificial Intelligence and Education Volume One. M. Y. Robert W. Lawler. Norwood, NJ, Ablex Publishing. 1.

OzSoils. (1996). University of New England, NSW, Australia.

Papert, S. (1984). *Tomorrow's Classrooms*. New Horizons in Educational Computing. M. Yazdani, Ellis Horwood Series AI.

Papert, S. (1987). *MICROWORLDS: TRANSFORMING EDUCATION*. Artificial Intelligence and Education Volume One. M. Y. Robert W. Lawler. Norwood, NJ, Ablex Publishing. 1.

Pellone, G. (1995). *Educational software design: A literature review*. Australian Journal of Educational Technology 1995.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet11/su95p68.html

Pérez, T.A., Lopistéguy, P., Gutiérrez, J. and Usandizaga, I. (1995). *HyperTutor: From Hypermedia to Intelligent Adaptive Hypermedia*. World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Quinn, C. N., Ed. (1994). *Designing Educational Computer Games*. In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Rees, K. (1995). *Design issues in computer-based education*. Australian Journal of Educational Technology 1995.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet11/wi95p28.html

Reeves, T.C. (1992). *Effective Dimensions of Interactive Learning Systems*. ITTE'92. Proceedings of the 2nd International Conference on IT for Training and Education, University of Queensland, Australia.

Reil, M.M., Levin, J. A. And Miller-Souviney, B. (1987). *LEARNING WITH INTERACTIVE MEDIA: DYNAMIC SUPPORT FOR STUDENTS AND TEACHERS*. Artificial Intelligence and Education Volume One. M. Y. Robert W. Lawler. Norwood, NJ, Ablex Publishing. 1.

Reynolds, L. and Ehrlich, D. (1992). *Multimedia in Industry and Education*
A Decision model for Design. IMMS 1992.

Romiszowski, A. (1987). *Artificial Intelligence and Expert Systems in Education: Progress, Promise and Problems*. Australian Journal of Educational Technology 1987.
http://cleo.murdoch.edu.au/gen/aset/ajet/ajet3/wi87p6.html

Rochelle, J. (1994). *What should Collaborative Technology Be? A Perspective From Dewey and Situated Learning*. Institute for Research on Learning

Rosmalen, P. V. (1994). *SAM, Simulation and Multimedia*. In T. deJong and L. Sarti (Eds.), Design and Production of multimedia and simulation-based learning material. Dordrecht, Netherlands.

Schank, R. C. (1994). *Active Learning through Multimedia.* IEEE Multimedia 1(1): 69-78.

Schank, R. C. (1995). *Engines for Education.* Hillsdale, New Jersy, Lawrence Erlbaum Associates.

Schiffman, S. S. (1995). *Instructional Systems Design - Five Views of the Field.* In G. A. Anglin (Eds.), Instructional Technology -Past, Present and Future. Libraries Unlimited Inc, 1995.

Schlusselberg, E. (1993). *Dans le Quartier St.Gervais.* In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Schute, V. J. and Glaser, R. (1990). *A Large-Scale Evaluation of an Intelligent Discovery World: Smithtown.* Interactive Learning Environments. Vol. 1, No. 1. March 1990.

Schweir, R. A. (1995). *Issues in Emerging Interactive Technologies.* In G. A. Anglin (Eds.), Instructional Technology -Past, Present and Future. Libraries Unlimited Inc, 1995.

Self, J. (1988). *Student Models: What use are they?.* In P. Ercolli and R. Lewis (Eds.), Artificial Intelligence Tools in Education. Elsvier Science Publishers B.V. (North-Holland).

Shapira, D. (1995). *Learning to Read with Joy by Use of Computer-based Graphic Representation.* World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95 Graz, Austria, Association for the advancement of Computing in Education (AACE).

Singh, R. (1986). *Interactive Video in Education and Training.* Artificial Intelligence for Society. K. S. Gill, John Wiley & Sons Ltd.

Sleeman, D. H. (1983). Inferring Models for Intelligent Computer-Aided Instruction. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Machine Learning - An AI Approach. Tioga Publishing Company.

Smith, D. (1986). *IT, AI and the Electronic Sabre-Tooth.* Artificial Intelligence for Society. K. S. Gill, John Wiley & Sons Ltd.

Steinberg, E.R. (1989). *Cognition and Learner Control: A Literature Review, 1977-1988.* Journal of Computer-Based Instruction Autumn 1989, Vol. 16, No. 4, 117-121.

Steinberg, E.R. (1991). *Computer-Assisted Instruction, A Synthesis or theory, Practice and Technology.* Lawrence Erlbaum Associates, Publishers.

Stern, M., Beck, J. and Woolf, B.P. (1996). *Adaptation of Problem Presentation and Feedback in an Intelligent Mathematics tutor.* In S. Frasson, C. Gautheir and A. Lesgold (Eds.), Intelligent Tutoring Systems, ITS'96 Proceedings. Spinger-Verlag.

Streibel, M. J. (1995). *Instructional Plans and Situated Learning - The Challenge of Suchman's Theory of Situated Action for Instructional Designers and Instructional Systems.* In G. A. Anglin (Eds.), Instructional Technology -Past, Present and Future. Libraries Unlimited Inc, 1995.

Takeuchi, A. and Otsuki, S. (1988). *A study of Student models and learner-machine interaction.* In P. Ercolli and R. Lewis (Eds.), Artificial Intelligence Tools in Education. Elsvier Science Publishers B.V. (North-Holland).

The Equation Solving Tutor. (1993). Steven Ritter, CMU, Pittsburg, PA, USA.

Tripp, S.D. (1993). *Theories, Traditions, and Situated Learning.* Educational Technology, March 1993.

Wang, H. (1996). *Learn Media: A Co-operative Intelligent Tutoring System for Learning Mulitimedia.* In S. Frasson, C. Gautheir and A. Lesgold (Eds.), Intelligent Tutoring Systems, ITS'96 Proceedings. Spinger-Verlag.

Webster, M. (1993). *Electronic Books.* In M. E. Hodges and R. M. Sansett (Eds.), Multimedia computing: case studies from MIT Project Athena/. Addison-Wesley Publishing Company.

Wills, S. and Swart, R. (1994). *The book is dead, long live the book: Designing interactive publications.* In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Woolsey, K.H. (1994). *Multimedia Opportunities.* In K. Beattie, C. McNaught and S. Wills (Eds.), Interactive Multimedia in University Education: Designing for Change in Teaching and Learning. Elsevier Science B.V.

Winship, J.A. (1989). *Information Technologies in Education The Quest for Quality Software.* Center for Educational Research and Innovation, OECD, Paris.

Woolf, B.P. and Hall, W. (1995). *Multimedia Pedagogues, Interactive Systems for teaching learning.* IEEE, May 1995.

Yazdani, M., (1986). *Artificial Intelligence and Education: A Critical Review.* In Gill, K.S. (eds.) AI for Society, John Willey & Sons Ltd.

Yazdani, M. (1987). *INTELLIGENT TUTORING SYSTEMS: AN OVERVIEW.* Artificial Intelligence and Education Volume One. M. Y. Robert W. Lawler. Norwood, NJ, Ablex Publishing.

Zhao, Y. (1995). *From the Technology of Teaching to the Technology of Learning: Creating a Virtual Community of Learners On The Internet.* World Conference on Educational Multimedia and Hypermedia ED-MEDIA 95, Graz, Austria, Association for the advancement of Computing in Education (AACE).

Appendix A

System Description of the packages mentioned

Package	*Description*	*Features*	*Developed At*	*Hardware Requirements*
ANGLE 3.3j (1989)	Cognitive tutor for geometry theorem proving	Include step-by-step feedback, learner-centred feedback and skill meter to indicate the progress	Carnegie-Mellon University, Pittsburgh, USA by Kenneth R. Koedinger, John R. Anderson, Doug Roesh and Jeremy Resnick	Macintosh II-series computer with minimum 8MB RAM , 6.0.5 or later , black & white monitor and windows environment with mouse
The Equation Solving Tutor (1993)	Cognitive tutor for solving linear equations	Individualised help, knowledge tracing, generality and flexibility	Carnegie-Mellon University, Pittsburgh, USA by Steven Ritter using Tutor Development kit developed by Ray Pelletier	Macintosh II-series computer with minimum 8MB RAM , 6.0.5 or later , black & white monitor and windows environment with mouse
ACT Programming Tutor (APT) Pascal Tutor (1993)	Cognitive tutor for learning introductory Pascal programming	Individualised help, customised exercise sequence, skill meter	Carnegie-Mellon University, Pittsburgh, USA by Albert Corbett, Anil Kulshrestha and John Anderson using Tutor Development kit	Macintosh II-series computer with minimum 8MB RAM , 6.0.5 or later , black & white monitor and windows environment with mouse
LifeLearn Companion Animal Dermatology (1997)	Case studies in animal dermatology for practice and assessment	Interactive multimedia, immediate feedback, scaffolding	Lifelearn Inc University of Guelph, Canada by E.J.Rosser and J. Littlewood	Windows 3.1 or above, double speed CD ROM drive, hard disk, 256 colours, sound card, QuickTime 2 and 8 MB RAM.

Package	*Description*	*Features*	*Developed At*	*Hardware Requirements*
The Algebra Word Problem Tutor (1993)	Cognitive tutor to learn algebraic skills and use them in the context of real-life problem situations	Individualised help, customised exercise sequence	Carnegie-Mellon University, Pittsburgh, USA by Steven Ritter using Tutor Development kit	Macintosh II-series computer with minimum 8MB RAM , 6.0.5 or later , black & white monitor and windows environment with mouse
OzSoils (1996)	Interactive multimedia courseware in Soil Science	Non-linear menu structure, questionnaire for revision	University of New England, NSW, Australia	Macintosh Version: Any Macintosh computer (colour monitor) with system 6.0.8 or above and QuickTime 1.5. At least 2.5 MB of RAM must be available to the package to run PC Version: Windows 3.1 or above, double speed CD ROM drive, hard disk, 256 colours, sound card, QuickTime 2 and 8 MB RAM.

Package	*Description*	*Features*	*Developed At*	*Hardware Requirements*
Certificate in Remote Area Servicing (1996)	Interactive Multimedia package developed for Department of Employment, Education, Training and Youth Affairs, Australia, to train their busy and isolated staff	Access, accredited training, articulation, workplace competencies, action learning, learning in the workplace, relevance, workplace projects and recognition for prior learning	University of New England, NSW, Australia	Windows 3.1 or above, double speed CD ROM drive, hard disk, 256 colours, sound card, QuickTime 2 and 8 MB RAM.

Package	*Description*	*Features*	*Developed At*	*Hardware Requirements*
Diagnosis for Crop Protection (1997)	multimedia software package that aids in teaching the process of diagnosing crop problems	Multimedia, delayed Feedback, note pad.	A joint project between Massey University , New Zealand and the Co-operative Research Center for Tropical Pest Management, QLD, Australia	Windows Version: IBM compatible computer running Windows 3.1 or Windows95 with approximately 6 Megabytes of disk space and soundcard. Macintosh Version: Minimum a Macintosh LC2 with 512 KB VRAM and a 16 bit colour monitor, Hypercard Developer's Toolkit" to run the Builder and approximately 6 Megabytes of disk space. Dos Version: An IBM compatible computer with at least 640 K of memory. A SVGA graphics card and monitor. Video cannot be shown with the DOS version

Package	*Description*	*Features*	*Developed At*	*Hardware Requirements*
Investigating Lake Illuka (1993)	Simulation of an imaginary lake environment designed to support the teaching of ecology	Media room that contains news paper clippings, radio discussions, video clips and a reference book in the media room to be read/watched. Physical, chemical and biological tools to measure various characteristics of an ecosystem. A user notebook for collecting information directly from the package.	Interactive Multimedia Unit, Faculty of Education, University of Wollongong, Australia	Any Macintosh computer (colour monitor) with system 6.0.8 or above and QuickTime 1.5. At least 2.5 MB of RAM must be available to the package to run

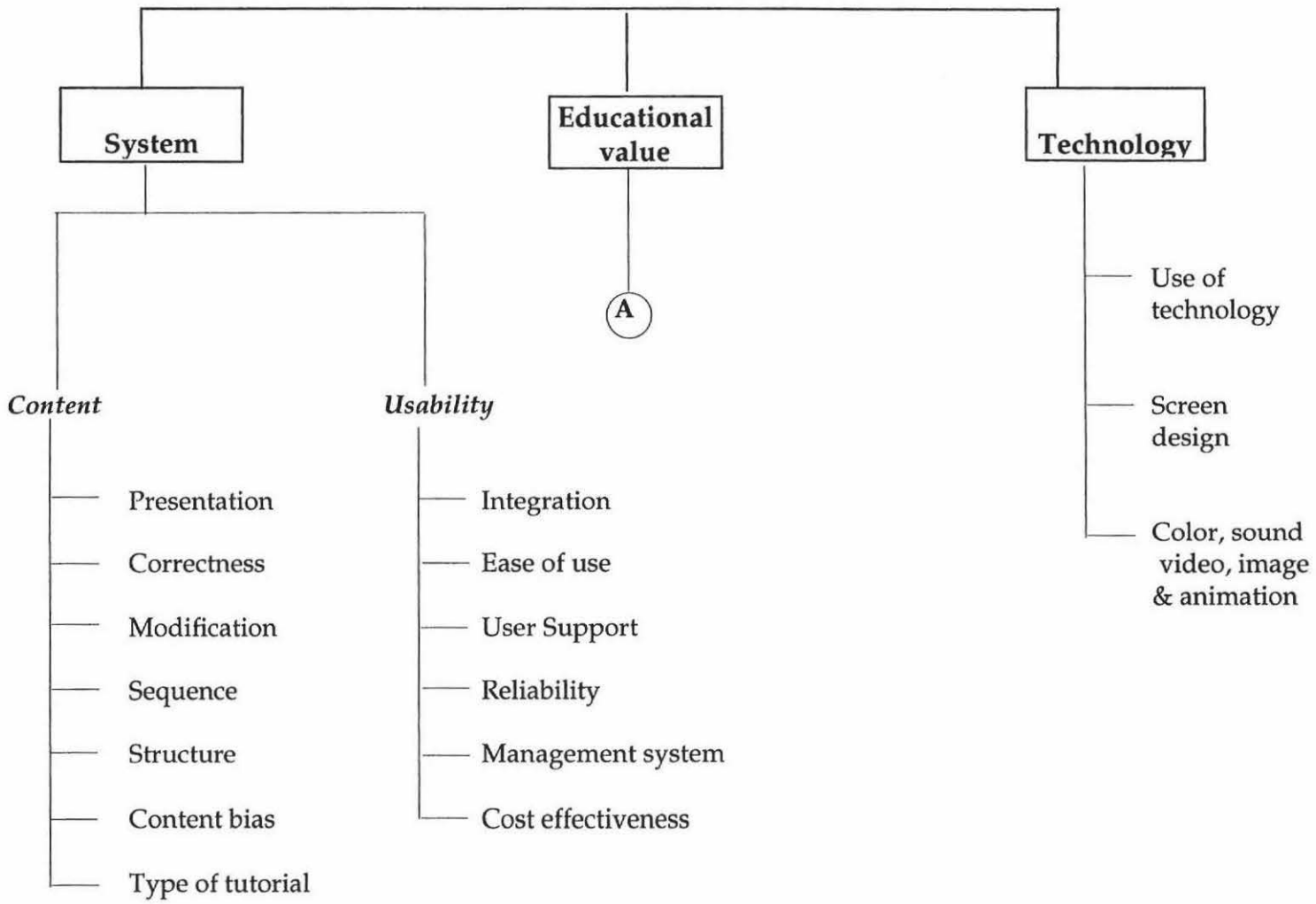
Package	*Description*	*Features*	*Developed At*	*Hardware Requirements*
Exploring the Nardoo (1996)	Simulation of a changing island river catchment, from late last century to a development affected environment today	Personal Digital Assistance (PDA), a multipurpose device which allows the student to navigate the river, take measurements, view and collect news items, make notes and get help. Water research center (WRC), which has a computer catalogue, a Plant and Animal Book, Film, television and radio clipboards, River investigation notice board, a Filling cabinet, a simulator and a clipboard folder with news paper clips.	Interactive Multimedia Unit, Faculty of Education, University of Wollongong in collaboration with the NSW Department of Land and Water Conservation, Australia	Macintosh: Colour capable Macintosh with System 7.1 or above, double speed CD ROM drive, hard disk, QuickTime 2 and an additional 2.5MB RAM. PC : Windows 3.1 or above, double speed CD ROM drive, hard disk, 256 colours, sound card, QuickTime 2 and 8 MB RAM.

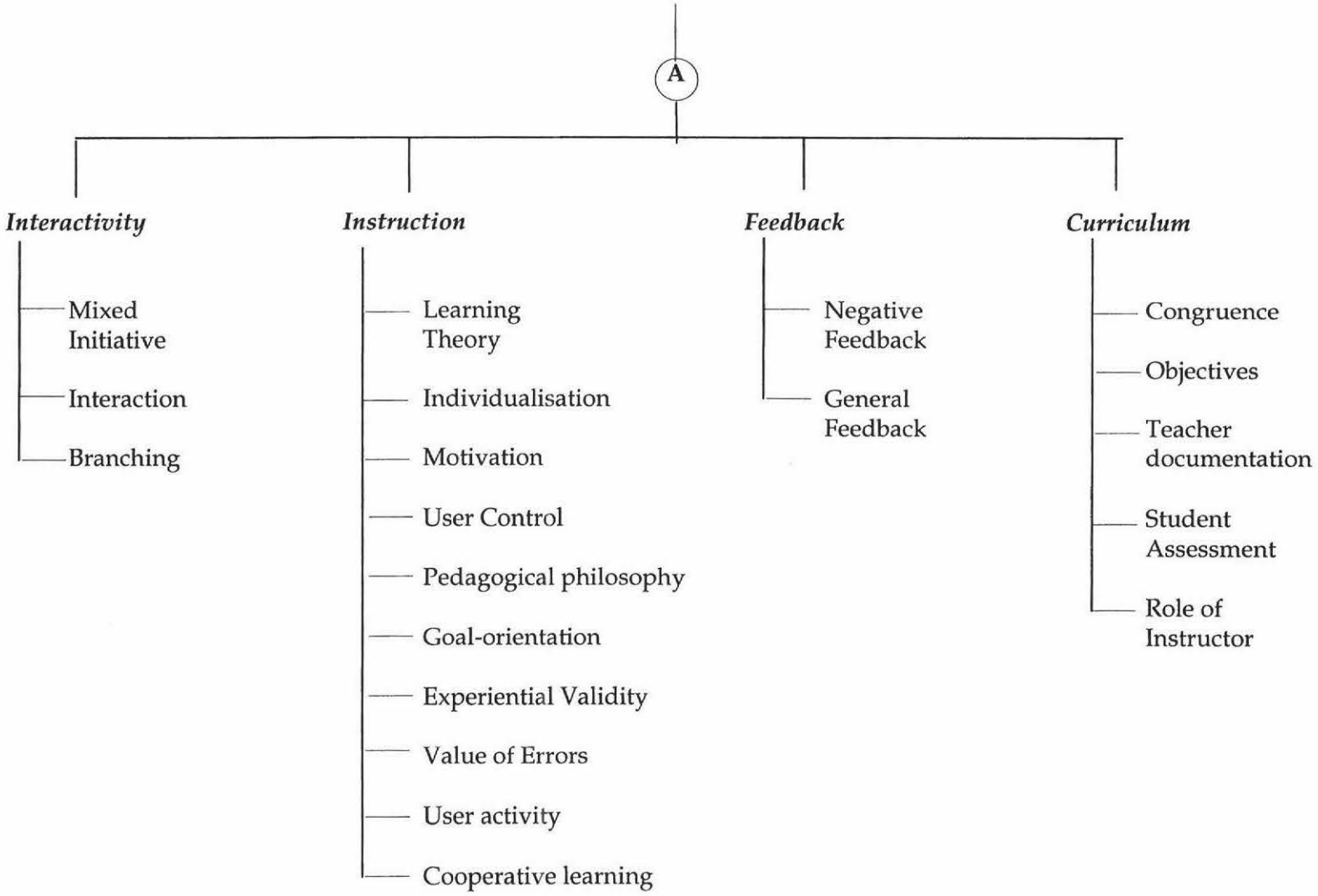
Appendix B

Characteristics of good Educational Software

This page is intentionally left blank.

Characteristics of good Educational Software




A
Interactivity
Mixed Initiative
Interaction
Branching
Instruction
Learning Theory
Individualisation
Motivation
User Control
Pedagogical philosophy
Goal-orientation
Experiential Validity
Value of Errors
User activity
Cooperative learning
Feedback
Negative Feedback
General Feedback
Curriculum
Congruence
Objectives
Teacher documentation
Student Assessment
Role of Instructor

Appendix C

Essential use-cases of Diagnosis

Use cases of Builder module

1. Start Diagnosis

User Action	System Response
Execute Diagnosis	
	Display Name Prompt
Enter name	
	Display Student Number prompt
Enter student number	
	Ask confirmation
Confirm entries	
	Start Diagnosis and display main menu

2. Load Scenario

User Action	System Response
	Display Main menu
Select File option	
	Display more options
Select 'Load Scenario'	
	Display available scenario types and their directories
Select and confirm options	
	Load initial scenario and display menu

3. Create Scenario

User Action	System Response
	Display Main menu
Select File option	
	Display more options
Select 'New Scenario' or 'Load Scenario'	
	Display available scenario types (Apples, Bananas, Berry Crops etc.)
Select scenario type and confirm	
	Display initial scenario information window
Enter/delete/modify text, audio, video, image entries , mark if it is an important clue and confirm	
	Save changes, close window and go back to main window

4. Build Initial Scenario

User Action	System Response
Select Build Scenario option	Display Main menu
Select 'Initial scenario information'	Display more options
Enter/delete/modify text, audio, video, image entries , mark if it is an important clue and confirm	Display initial scenario information window Save changes, close window and go back to main window

5. Build Lab Information

User Action	System Response
Select Build Scenario option	Display Main menu
Select 'Initial lab information'	Display more options
Enter/delete/modify text, audio, video, image entries, mark if it is an important clue and confirm	Display initial lab information window Save changes, close window and go back to main window

6. Build Weather Information

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select 'Research weather information'	Display more options
Enter/delete/modify text, audio, video, image entries, mark if it is an important clue and confirm	Display research weather information window Save changes, close window and go back to main window

7. Build Solution to the problem

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Problem solution'	
	Display problem solution window
Enter/delete/modify text entry and confirm	
	Save changes, close window and go back to main window

8. Build Control Action

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Recommended control action'	
	Display recommended control action window
Enter/delete/modify text entry and confirm	
	Save changes, close window and go back to main window

9. Build Debriefing introduction

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Debriefing introduction'	
	Display 'Debriefing introduction window
Enter/delete/modify text entry and confirm	
	Save changes, close window and go back to main window

10. Build Optimal Route

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select 'Optimal route'	Display more options
Enter/delete/modify text entry and confirm	Display Optimal route window
	Save changes, close window and go back to main window

11. Set Show debriefing option

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select 'Choose debriefing'	Display more options
Tick on button and confirm	Display show debriefing window
	Save changes, close window and go back to main window

12. Add Background Image

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select Add Background Image	Display more options
	Display Add Background Image window
Select image file type	Display Image file type, directories
	Display files of selected type and their directories.
Select View	Display image
Confirm entry	Save changes, close window and go back to main window

13. Add Lab Image

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select Add Lab Image	
	Display Add Lab Image window
	Display Image file type, directories
Select image file type	
	Display files of selected type and their directories.
Select View	
	Display image
Confirm entry	
	Save changes, close window and go back to main window

14. Add Scenario credits

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select Add Scenario credits	
	Display Add scenario credits window
Enter scenario writer, acknowledgement information and confirm	
	Save changes, close window and go back to main window

15. Build Examine Plant part (Leaves/Stems/Seeds/Flowers/Fruits/Roots) information

User Action	System Response
	Display Main menu
Select *Build Actions* option	
	Display more options
Select 'Examine part'	
	Display submenu (Look/Cut into/ Collect/ Insects)
Select Look	
	Display Look window
Enter/delete/modify text, audio, video, image entries, mark if it is an important clue and confirm	
	Save changes and close window and go back to main menu

16. Build Actions

User Action	System Response
	Display Main menu
Select File option	
	Display more options
Select 'New Scenario' or 'Load Scenario'	
	Display available scenario types (Apples, Bananas, Berry Crops etc.)
Select scenario type and confirm	
	Display initial scenario information window
Enter/delete/modify text, audio, video, image entries , mark if it is an important clue and confirm	
	Save changes, close window and go back to main window
Select Build actions	
	Display Build actions submenu depending on the crop type

17. Build Cut into Plant part

User Action	System Response
	Display Main menu
Select *Build Actions* option	
	Display more options
Select 'Examine part'	
	Display submenu (Look/Cut into/ Collect/ Insects). If Look has been selected already display other options
Select Cut into	
	Display Cut Into window
Enter/delete/modify text, audio, video, image entries, mark if it is an important clue and confirm	
	Save changes and close window and go back to main menu

18. Build Collect Plant part/Soil/Weed

User Action	System Response
	Display Main menu
Select *Build Actions* option	
	Display more options
Select 'Examine part'	
	Display submenu (Look/Cut into/ Collect/ Insects). If Look has been selected already display other options
Select Collect	
	Display Collect window
Select Add lab procedures or Referrals	
	Go to lab procedure window or Referral window

19. Add lab procedures (Microscope/Agar plate/ Humidity Chambers/ Nematode Extraction)

User Action	System Response
	Display Lab procedure window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Enter cost for the procedure if any. Confirm entry.	
	Save changes and close window and go back to Collect window

20. Add Referrals (Virus Testing/ Bacterial Test/ Plant Nutrient Analysis/ Pesticide Residue)

User Action	System Response
	Display Referral window
Enter virus name or select from the existing list. Enter response for the test Mark if it is an important clue. Enter cost for the procedure if any. Confirm entry.	
	Display response and ask if it is an important clue
Mark if it is an important clue. Confirm entry.	
	Save changes and close window and go back to Collect window

21. Build Insects in Plant part information

User Action	System Response
	Display Main menu
Select *Build Actions* option	
	Display more options
Select 'Examine part'	
	Display submenu (Look/Cut into/ Collect/ Insects). If Look has been selected already display other options
Select Insects	
	Display Add Insects window
Select Next Insect	
	Display next insect number
Select previous insect	
	Display previous insect number
Select Examine insects or Collect Insect or Sample insects	
	Go to Examine insects or Collect Insect or Sample insects window respectively

22. Examine Insects in Plant part

User Action	System Response
	Display Add Insects window
Select Examine insects	
	Display Examine insects window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Go back to Add insects window

23. Add resistance testing for collected Insects from Plant part

User Action	System Response
	Display Resistance information window
Select or enter name Enter response for testing. Enter cost involved, if any. Confirm entry.	
	Display header information window
Enter header information Mark if it is an important clue Confirm entry	
	Goto Insect Procedures window

24. Collect Insects in Plant part

User Action	System Response
Select Collect Insects	Display Add Insects window
Select Add lab procedures (Microscope / Insect Rearing)	Display Insect Procedures window
Enter/delete/modify text, audio, video, image entries. Enter cost involved, if any. Mark if it is an important clue. Confirm entry.	Display Collect Insect - procedure window
Select Send away for Pest identification	Go back to Insect Procedures window
Enter/delete/modify text, audio, video, image entries. Enter cost involved, if any. Mark if it is an important clue. Confirm entry.	Display Collect Insect - procedure window
Select Send away for Resistance testing	Go back to Insect Procedures window Goto Resistance Information window

25. Sample Insects in Plant part

User Action	System Response
Select Sample Insects	Display Add Insects window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display Sample insects window
	Go back to Add insects window

26. Build Weed, Soil information

User Action	System Response
Select *Build Actions* option	Display Main menu
Select 'Examine weed' or Examine soil	Display more options
Select Look	Display submenu (Look/Collect/ Insects).
Enter/delete/modify text, audio, video, image entries, mark if it is an important clue and confirm	Display Look at Soil/Weed window Go back to main menu

27. Build Collect Soil/Weed

User Action	System Response
Select *Build Actions* option	Display Main menu
Select 'Examine soil' or Examine weed	Display more options
Select Collect	Display submenu (Look/ Collect/ Insects). If Look has been selected already display other options
Select Add lab procedures or Referrals	Display Collect window Goto lab procedure window or Referral window

28. Add lab procedures for Soil (Examine/ Nematode Extraction/Baiting)

User Action	System Response
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Enter cost for the procedure if any. Confirm entry.	Display Lab procedure window Save changes and close window and go back to Collect window

29. Add lab procedures for Weed (Microscope/Agar plate/ Humidity Chambers/ Nematode Extraction)

User Action	System Response
	Display Lab procedure window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Enter cost for the procedure if any. Confirm entry.	
	Save changes and close window and go back to Collect window

30. Add Referrals for Soil (Nutrient pH Analysis/ Pesticide Residue)

User Action	System Response
	Display Referral window
Enter virus name or select from the existing list. Enter response for the test Mark if it is an important clue. Enter cost for the procedure if any. Confirm entry.	
	Display response and ask if it is an important clue
Mark if it is an important clue. Confirm entry.	
	Save changes and close window and go back to Collect window

31. Add Referrals for Weed (Virus Testing/ Bacterial Test/ Plant Nutrient Analysis/ Pesticide Residue)

User Action	System Response
	Display Referral window
Enter virus name or select from the existing list. Enter response for the test Mark if it is an important clue. Enter cost for the procedure if any. Confirm entry.	
	Display response and ask if it is an important clue
Mark if it is an important clue. Confirm entry.	
	Save changes and close window and go back to Collect window

32. Build Insects in Soil/Weed information

User Action	System Response
	Display Main menu
Select *Build Actions* option	
	Display more options
Select 'Examine Soil' or Examine Weed	
	Display submenu (Look/ Collect/ Insects). If Look has been selected already display other options
Select Insects	
	Display Add Insects window
Select Next Insect	
	Display next insect number
Select previous insect	
	Display previous insect number
Select Examine insects or Collect Insect or Sample insects	
	Goto Examine insects or Collect Insect or Sample insects window respectively

33. Examine Insects in Soil/Weed

User Action	System Response
	Display Add Insects window
Select Examine insects	
	Display Examine insects window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Go back to Add insects window

34. Add resistance testing for collected Insects from Soil/Weed

User Action	System Response
	Display Resistance information window
Select or enter name Enter response for testing. Enter cost involved, if any. Confirm entry.	
	Display header information window
Enter header information Mark if it is an important clue Confirm entry	
	Goto Insect Procedures window

35. Collect Insects in Soil/Weed

User Action	System Response
Select Collect Insects	Display Add Insects window
Select Add lab procedures (Microscope / Insect Rearing)	Display Insect Procedures window
Enter/delete/modify text, audio, video, image entries. Enter cost involved, if any. Mark if it is an important clue. Confirm entry.	Display Collect Insect - procedure window
Select Send away for Pest identification	Go back to Insect Procedures window
Enter/delete/modify text, audio, video, image entries. Enter cost involved, if any. Mark if it is an important clue. Confirm entry.	Display Collect Insect - procedure window
Select Send away for Resistance testing	Go back to Insect Procedures window Goto Resistance Information window

36. Sample Insects in Soil/Weed

User Action	System Response
Select Sample Insects	Display Add Insects window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display Sample insects window
	Go back to Add insects window

37. Add Symptom distribution

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select 'Other observations'	Display more options
Select Symptom Distribution	Display submenu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display symptom distribution window Save changes, close window and go back to main window

38. Add Insect Trap details

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select 'Other observations'	Display more options
Select Insect Trap	Display submenu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display Insect Trap window Save changes, close window and go back to main window

39. Add Sprayer usage details

User Action	System Response
Select *Build Scenario* option	Display Main menu
Select 'Other observations'	Display more options
Select Sprayer	Display submenu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display Sprayer Trap window Save changes, close window and go back to main window

40. Add Fertilizer Applicator details

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Other observations'	
	Display submenu
Select Fertilizer Applicator	
	Display Fertilizer Applicator window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

41. Add Plant damage assessment information

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Quantitative Assessment'	
	Display submenu
Select Plant Damage	
	Display Plant Damage window
Enter/delete/modify text, audio, video, image entries. Enter cost, if any. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

42. Add Ask grower about past weather information

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Past weather window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

43. Add Weed assessment information

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Quantitative Assessment'	
	Display submenu
Select Weed	
	Display Plant Damage window
Enter/delete/modify text, audio, video, image entries. Enter cost, if any. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

44. Add nematode assessment information

User Action	System Response
	Display Main menu
Select *Build Scenario* option	
	Display more options
Select 'Quantitative Assessment'	
	Display submenu
Select Nematode	
	Display Plant Damage window
Enter/delete/modify text, audio, video, image entries. Enter cost, if any. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

45. Add Ask grower about disease management

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Disease management window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

46. Add Ask grower about weed management

User Action	System Response
Select *Build Ask* option	Display Main menu
Select Ask Grower about...	Display sub menu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display weed management
	Save changes, close window and go back to main window

47. Add Ask grower about insect management

User Action	System Response
Select *Build Ask* option	Display Main menu
Select Ask Grower about...	Display sub menu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display Insect Management window
	Save changes, close window and go back to main window

48. Add Ask grower about nematicides

User Action	System Response
Select *Build Ask* option	Display Main menu
Select Ask Grower about...	Display sub menu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display nematicides window
	Save changes, close window and go back to main window

49. Add Ask grower about drainage

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Drainage window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

50. Add Ask grower about Fertilizer

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Fertilizer window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

51. Add Ask grower about Irrigation

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Irrigation window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

52. Add Ask grower about variety/Cultivar/Source

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Variety/Cultivar/Source window
Enter/delete/modify text, audio, video, image entries.	
Mark if it is an important clue.	
Confirm entry.	Save changes, close window and go back to main window

53. Add Ask grower about Land use history

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Land use history window
Enter/delete/modify text, audio, video, image entries.	
Mark if it is an important clue.	
Confirm entry.	
	Save changes, close window and go back to main window

54. Add Ask grower about neighbour's crop

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Neighbour's crop window
Enter/delete/modify text, audio, video, image entries.	
Mark if it is an important clue.	
Confirm entry.	
	Save changes, close window and go back to main window

55. Add Ask grower about Grower Diagnosis

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Grower diagnosis window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

56. Add Ask grower about cultivation

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Cultivation window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

57. Add Ask grower about Changes in management

User Action	System Response
	Display Main menu
Select *Build Ask* option	
	Display sub menu
Select Ask Grower about...	
	Display Changes in Management window
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	
	Save changes, close window and go back to main window

58. Add Ask grower about History of problem

User Action	System Response
Select *Build Ask* option	Display Main menu
Select Ask Grower about...	Display sub menu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display History of Problem window
	Save changes, close window and go back to main window

59. Add Ask neighbour about the problem

User Action	System Response
Select *Build Ask* option	Display Main menu
Select Ask neighbour about...	Display sub menu
Enter/delete/modify text, audio, video, image entries. Mark if it is an important clue. Confirm entry.	Display Ask neighbour window
	Save changes, close window and go back to main window

60. Get Help

User Action	System Response
Select *Help* option	Display Main menu
Select Help	Display sub menu
Select topic	Display Help window
Select OK/Close	Display help information
	Close window and go back to main window

61. See License Conditions

User Action	System Response
Select *Help* option	Display Main menu
Select Licence Condition	Display sub menu
Select OK/Close	Display License condition window
	Close window and go back to main window

62. See details about Diagnosis Builder

User Action	System Response
Select *Help* option	Display Main menu
Select About Diagnosis	Display sub menu
	Display Programming and design details
Select OK/Close	Close window and go back to main window

Use cases of Diagnosis module

1. Start Diagnosis

User Action	System Response
Execute Diagnosis	Display Name Prompt
Enter name	Display Student Number prompt
Enter student number	Ask confirmation
Confirm entries	Start Diagnosis and display main menu

2. Load Scenario

User Action	System Response
Select File option	Display Main menu
Select 'Load Scenario'	Display more options
	Display available scenario types and their directories
Select and confirm options	Load initial scenario and display menu

3. Load User

User Action	System Response
Select File option	Display Main menu
Select 'Load User'	Display more options
	Display available users and their directories
Select and confirm options	Load saved scenario and display menu

4. Save user and exit in the middle of the Diagnosis

User Action	System Response
Select File option	Display Main menu
Select Save user and Exit	Display more options
Enter name, directory and confirm	Ask for name and directory
	Save changes, close window and exit diagnosis

5. Exit Diagnosis

User Action	System Response
Select File option	Display Main menu
Select Exit	Display more options
Confirm	Ask for confirmation
	Close window and exit diagnosis

6. Examine Plant part (Leaves/Stems/Seeds/Flowers/Fruits/Roots) / Weed/ Soil

User Action	System Response
Select *Action* option	Display Main menu
Select 'Examine part'	Display sub menu
	Display submenu (Look/Cut into/ Collect/ Insects) (if built).
Select Look	Display Text, Audio, Video, Image information (if built).
Select Help	Display help window
Select OK	Close window and go back to main menu

7. Cut Into Plant part (Leaves/Stems/Seeds/Flowers/Fruits/Roots) / Weed/ Soil

User Action	System Response
Select *Action* option	Display Main menu
Select Examine parts	Display sub menu
	Display submenu (Look/Cut into/ Collect/ Insects) (if built).
Select Cut Into	Display Text, Audio, Video, Image information (if built).
Select Help	Display help window
Select OK	Close window and go back to main menu

8. Collect Plant part (Leaves/Stems/Seeds/Flowers/Fruits/Roots) / Weed/ Soil

User Action	System Response
	Display Main menu
Select *Action* option	
	Display sub menu
Select Examine Parts	
	Display submenu (Look/Cut into/ Collect/ Insects) (if built).
Select Collect	
	Display Collected items window
Select Close	
	Close window and go back to main menu

9. Check Plant part (Leaves/Stems/Seeds/Flowers/Fruits/Roots) / Weed/ Soil for insects

User Action	System Response
	Display Main menu
Select *Action* option	
	Display sub menu
Select Examine Parts	
	Display submenu (Look/Cut into/ Collect/ Insects) (if built).
Select Insects	
	Go to Insects window

10. Look for Insects in Plant part / Weed/ Soil

User Action	System Response
	Display Insects Window
Select Next Insect	
	Display next insect number
Select previous insect	
	Display previous insect number
Select Examine insects or Collect Insect or Sample insects	
	Goto Examine insects or Collect Insect or Sample insects window respectively

11. Examine Insects in Plant part/ Weed/ Soil

User Action	System Response
	Display Insects window
Select Examine insects	
	Display text, audio, video, image information
Close window.	
	Go back to Insects window

12. Collect Insects in Plant part/ Weed/ Soil

User Action	System Response
Select Collect Insects	Display Insects window(optional)
Close window	Display Item collected
	Go back to Insects window

13. Sample Insects in Plant part/ Weed/ Soil

User Action	System Response
Select Sample Insects	Display Insects window
Close window	Display text, audio, video, image information
	Go back to Insects window

14. See symptom distribution

User Action	System Response
Select Action	Display Main menu
Select Other Observation	Display submenu(Optional)
Select Symptom Distribution	Display submenu(Optional)
Close window	Display text, audio, video, image information
	Go back to main menu

15. See Insect traps information

User Action	System Response
Select Action	Display Main menu
Select Other Observation	Display submenu (Optional)
Select Insect Traps	Display submenu(Optional)
Close window	Display text, audio, video, image information
	Go back to main menu

16. See Fertilizer Applicator information

User Action	System Response
Select Action	Display Main menu
Select Other Observation	Display submenu (Optional)
Select Fertilizer Applicator	Display submenu (Optional)
Close window	Display text, audio, video, image information
	Go back to main menu

17. See Sprayer usage information

User Action	System Response
Select Action	Display Main menu
Select Other Observation	Display submenu (Optional)
Select Sprayer Usage	Display submenu (Optional)
Close window	Display text, audio, video, image information
	Go back to main menu

18. Assess Plant damage

User Action	System Response
Select Action	Display Main menu
Select Quantitative Assessment	Display submenu (Optional)
Select Plant Damage	Display submenu (Optional)
	Display text, audio, video, image information
Close window	Display Cost
	Go back to main menu

19. Assess Weed

User Action	System Response
Select Action	Display Main menu
Select Quantitative Assessment	Display submenu (Optional)
Select Weed	Display submenu (Optional)
	Display text, audio, video, image information Display Cost
Close window	Go back to main menu

20. Assess Nematodes

User Action	System Response
Select Action	Display Main menu
Select Quantitative Assessment	Display submenu (Optional)
Select Nematodes	Display submenu (Optional)
	Display text, audio, video, image information Display Cost
Close window	Go back to main menu

21. See Collected Items

User Action	System Response
Select Show option	Display Main menu
Select Collected Items	Display sub menu
	Display Collected items window (Objects and Insects)
Select Close	Close window and go back to main menu

22. See Initial scenario Information

User Action	System Response
Select Show option	Display Main menu
Select Initial Scenario	Display sub menu
Select Close	Display Initial Scenario
	Close window and go back to main menu

23. See Running Totals

User Action	System Response
Select Show option	Display Main menu
Select Running Totals	Display sub menu
Select Close	Display number of actions, total cost
	Close window and go back to main menu

24. Use Note Pad

User Action	System Response
Select Show option	Display Main menu
Select NotePad	Display sub menu
Enter/Delete/Edit own notes	Display Notepad window
Select Close	Close window and go back to main menu

25. Ask grower about (Past weather,

User Action	System Response
Select Ask	Display Main menu
Select Ask grower about..	Display submenu (Optional)
Select past weather etc.	Display submenu (Optional)
Close window	Display text, audio, video, image information
	Go back to main menu

26. Ask grower 's neighbour about the problem

User Action	System Response
Select Ask	Display Main menu
Select Neighbour..	Display submenu (Optional)
Close window	Display text, audio, video, image information
	Go back to main menu

27. Go to Lab

User Action	System Response
Select Lab	Display Main menu
Select Goto lab	Display submenu (Optional)
Enter diagnosis, justification and recommendation	Display Preliminary diagnosis window
Confirm Entry	
Select OK	Close window and display initial lab information. Text, video, audio, image.
	Close window and display Collected items window

28. Exit from Lab

User Action	System Response
Select Exit	Display Lab menu
Select Exit	Display submenu
Confirm	Ask for confirmation Close lab menu and go back to main menu

29. Exit from Diagnosis while in the lab

User Action	System Response
Select Exit	Display Lab menu
Select Save User and Exit	Display submenu
Enter user name, user id and confirm	Display user information window
	Close lab menu and go back to main menu

30. Getting help while in the lab

User Action	System Response
Select Help	Display Lab menu
Select Help	Display submenu
Select Topic	Display Help window
Select OK/Close	Display help information
	Close lab menu and go back to main menu

31. See running totals while in the lab

User Action	System Response
Select Show	Display Lab menu
Select Running Totals	Display submenu
Select Close	Display Total cost
	Close window

32. Ask grower about (Past weather etc.) from lab

User Action	System Response
Select Phone Grower	Display Lab menu
Select Ask about....	Display submenu (Optional)
Select past weather etc.	Display submenu (Optional)
Select Close	Display text, audio, video, image information
	Close window

33. Research past weather from lab

User Action	System Response
Select Research	Display Lab menu
Select Research weather	Display submenu (Optional)
Select Close	Display text, audio, video, image information
	Close window

34. Examine Collected objects

User Action	System Response
	Display Lab menu, collected items window
Select Object	
	Display procedures and referrals window
Select Microscope procedure	
	Display text, audio, video, image information, cost
Select Agar Plate procedure	
	Goto Show fungus window
Select Humidity chamber	
	Display text, audio, video, image information, cost
Select Nematode Extraction	
	Display text, audio, video, image information, cost
Select Referral	
	Go to Referral window

35. Perform Agar plate procedure

User Action	System Response
	Display Lab menu, collected items window
Select Object	
	Display procedures and referrals window
Select Agar Plate procedure	
	Goto Show fungus window
Select Examine fungus	
	Display text, audio, video, image information, cost
Select Pathogenicity	
	Display text, audio, video, image information, cost
Select Identification	
	Display text, audio, video, image information, cost
Select Resistance testing	
	Go to Resistance window

36. Perform Referrals (Virus Testing, Bacterial test, Plant Nutrient analysis, Pesticide Residue)

User Action	System Response
	Display procedures and referrals window
Select Virus Testing etc.	
	Display viruses
Select Viruses for test	
	Display cost
Confirm	
	Display text information on test results
Select Close	
	Close window and go back to procedures and referrals window

37. Make final Diagnosis

User Action	System Response
	Display Main menu
Select Diagnosis option	
	Display sub menu
Select Final Diagnosis	
	Display window
Enter final diagnosis, justification and recommendation	
	Get confirmation
Select OK/Close	
	Close window and go back to main menu

38. Entering Solution menu

User Action	System Response
	Display Main menu
Select Solution option	
	Display sub menu
Select Show solution	
	Get user file name
Enter file name and confirm	
	Get confirmation on entering solution menu
Confirm	Go to Solution menu

39. See Solution and general feedback

User Action	System Response
Select Debriefing option	Display Solution menu
Select Solution and feedback	Display sub menu
Select OK/Close	Display solution, general feedback, number of actions and total cost
	Close window and go to Solution menu

40. See Optimal route for Solution

User Action	System Response
Select Debriefing option	Display Solution menu
Select Optimal route	Display sub menu
Select OK/Close	Display optimal route in arriving at the solution
	Close window and go to Solution menu

41. See Recommended Action

User Action	System Response
Select Debriefing option	Display Solution menu
Select Recommended action	Display sub menu
Select OK/Close	Display recommended action for preventing the problem
	Close window and go to Solution menu

42. Getting help on feedback

User Action	System Response
Select Help	Display Solution menu
Select Feedback help	Display sub menu
Select Topic	Display help window with feedback topics
Select OK/Close	Display help contents
	Close window and go to Solution menu

43. Print Results

User Action	System Response
Select File option	Display Solution menu
Select Print results	Display sub menu
Select OK/Close	Print results
	Close window and go to Solution menu

44. Set printer

User Action	System Response
Select File option	Display Solution menu
Select Printer	Display sub menu
Select printer	Display printer window
Select OK/Close	Set printer
	Close window and go to Solution menu

45. Select another scenario

User Action	System Response
Select File option	Display Solution menu
Select Another Scenario	Display sub menu
	Save changes and go back to main menu

46. Exit from Diagnosis while in Solution menu

User Action	System Response
Select File option	Display Solution menu
Select Exit Diagnosis	Display sub menu
Confirm option	Get confirmation
	Save changes, close windows and exit

47. Get Help

User Action	System Response
Select *Help* option	Display Main menu
Select Help	Display sub menu
Select topic	Display Help window
Select OK/Close	Display help information
	Close window and go back to main window

48. See License Conditions

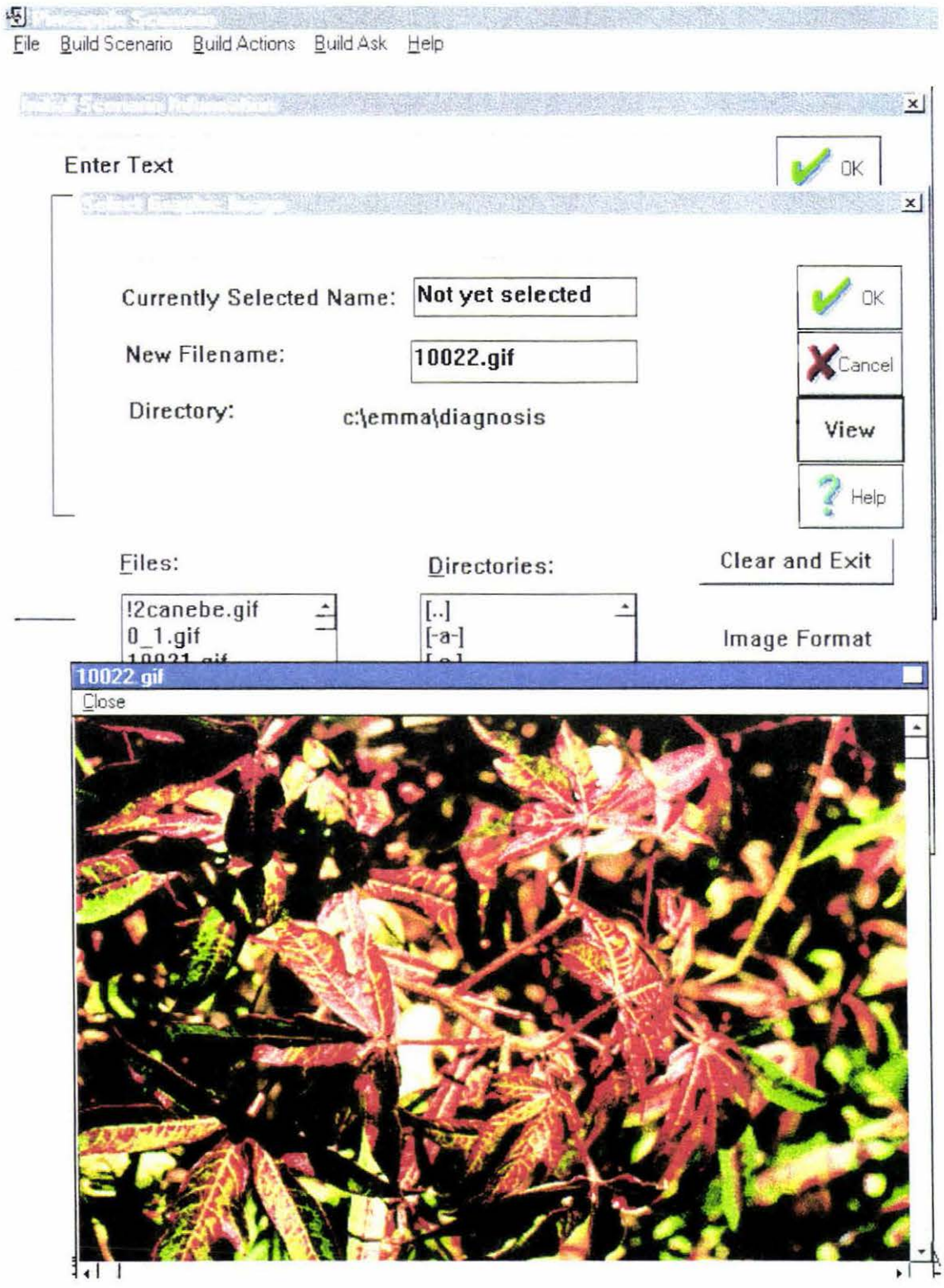
User Action	System Response
Select *Help* option	Display Main menu
Select Licence Condition	Display sub menu
Select OK/Close	Display License condition window
	Close window and go back to main window

49. See details about Diagnosis Builder

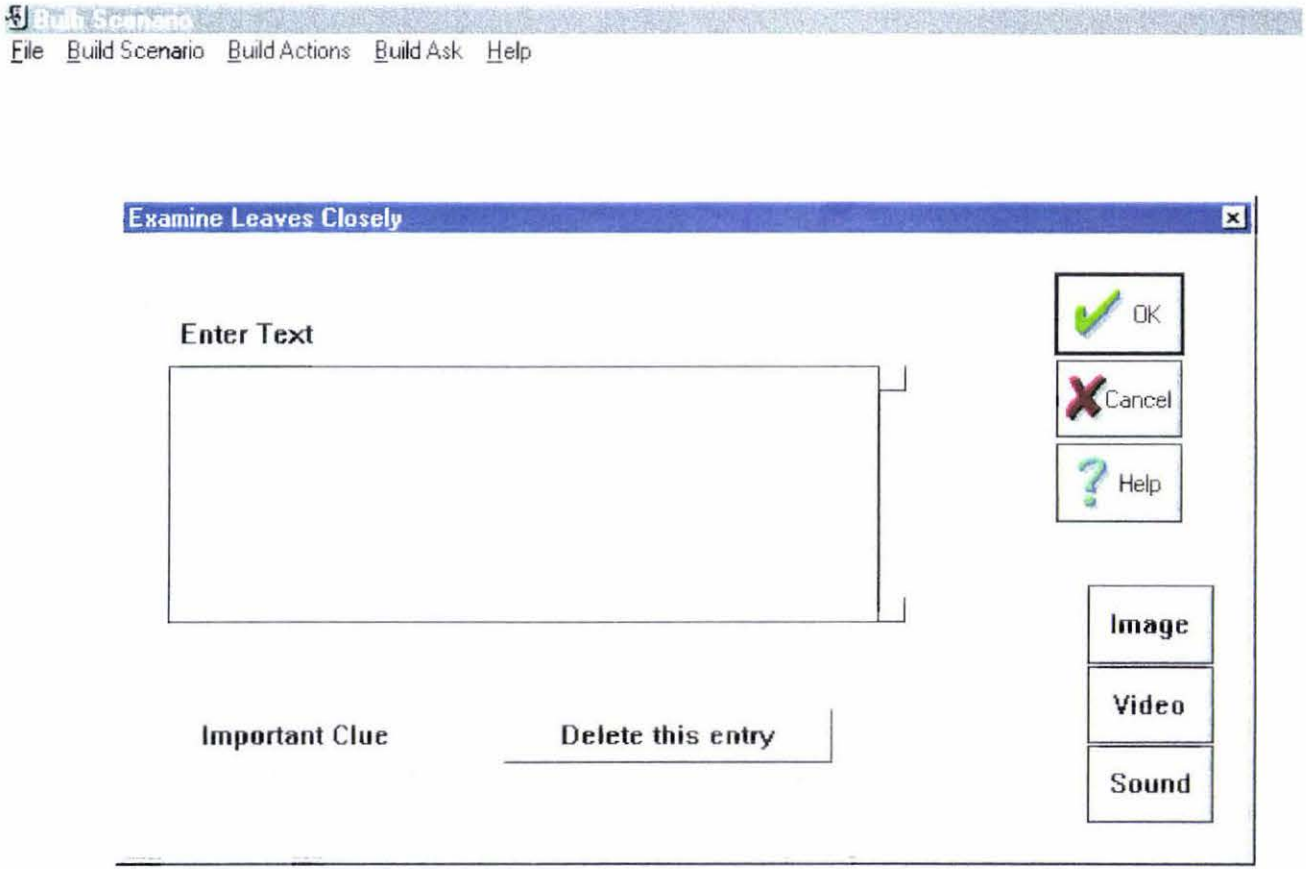
User Action	System Response
Select *Help* option	Display Main menu
Select About Diagnosis	Display sub menu
	Display Programming and design details
Select OK/Close	Close window and go back to main window

Appendix D

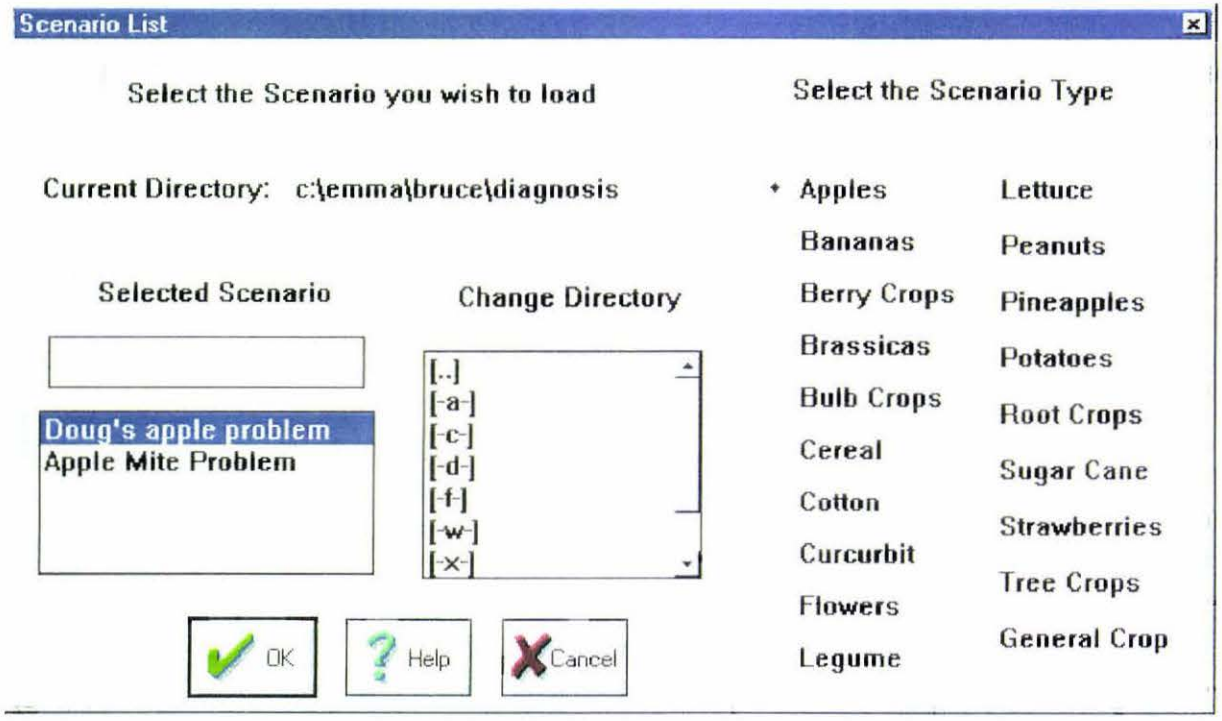
Screens from Diagnosis 2.1



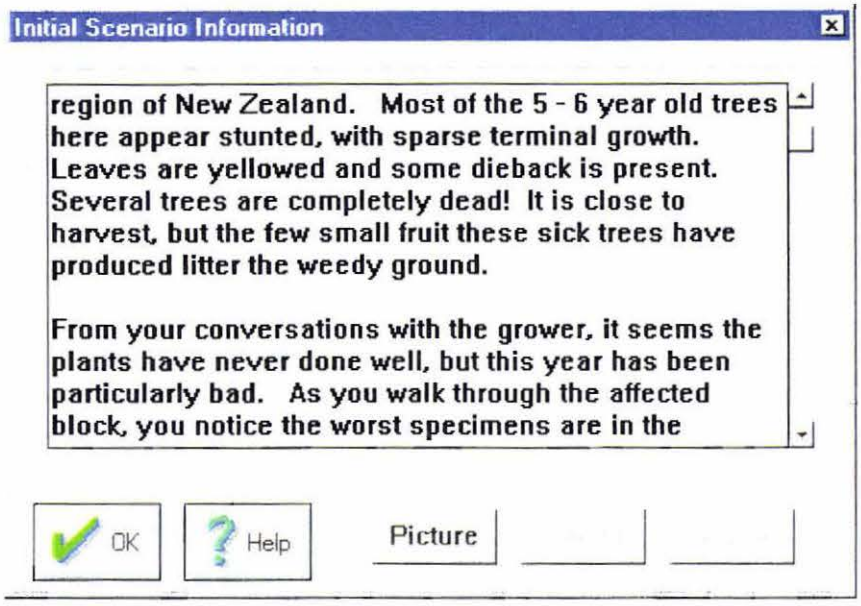
Viewing image while building initial scenario information

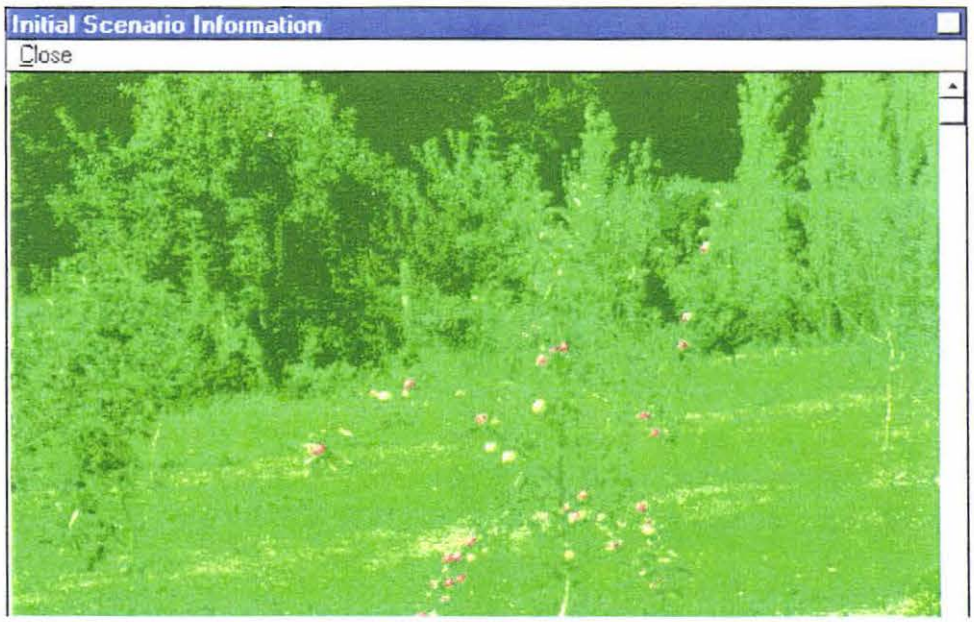


Entering information about leaves while building problem

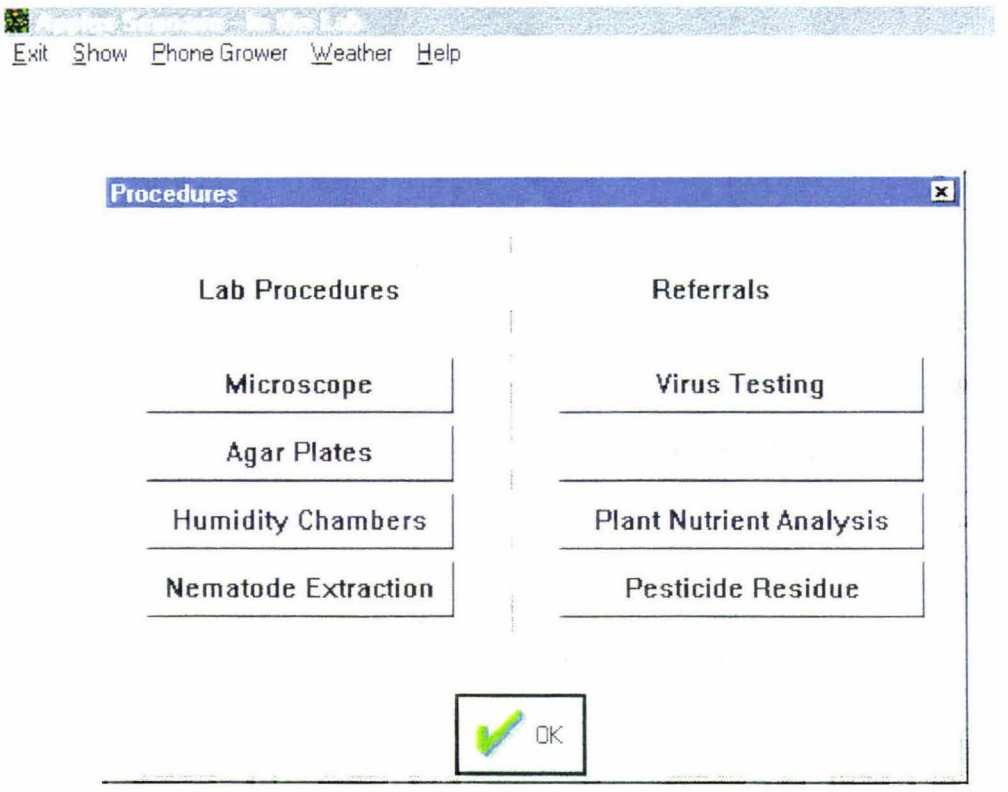


Selecting a problem for playing

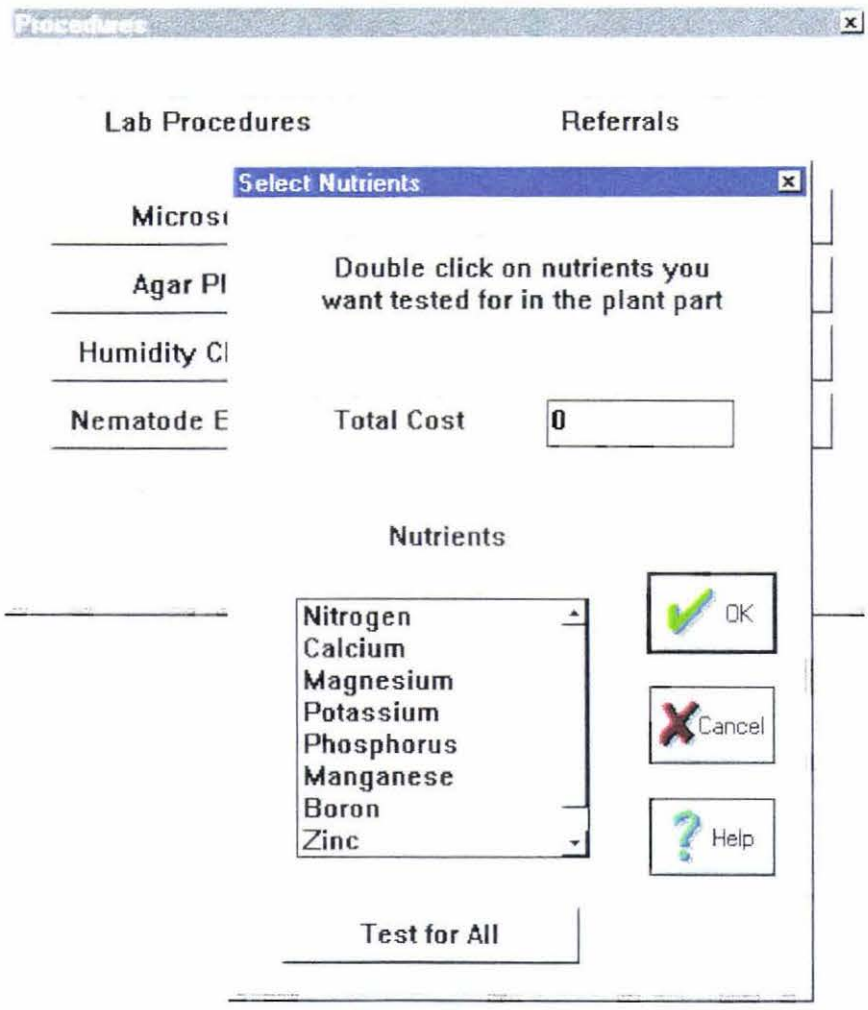




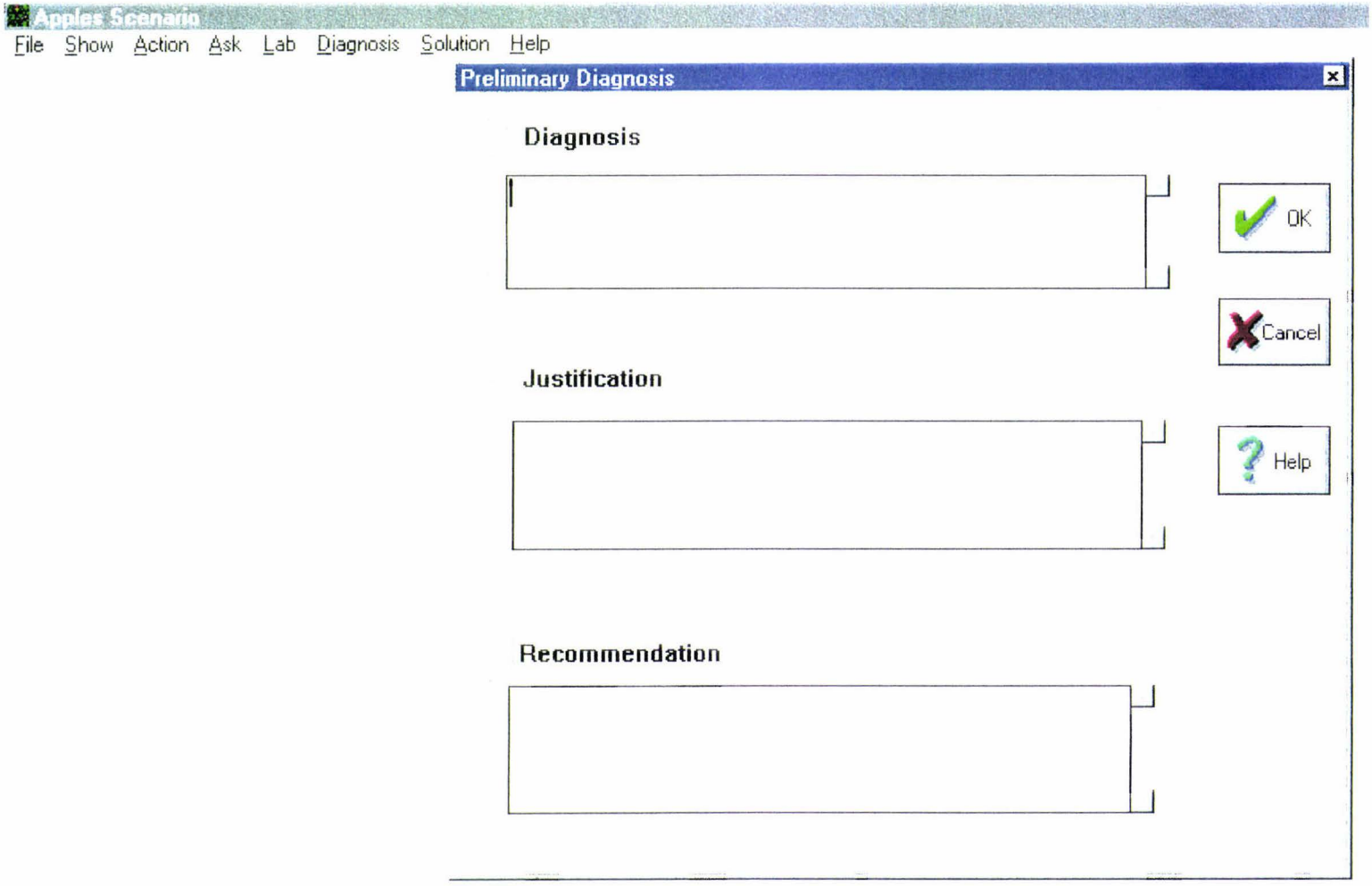
Initial scenario information presented by the system while playing



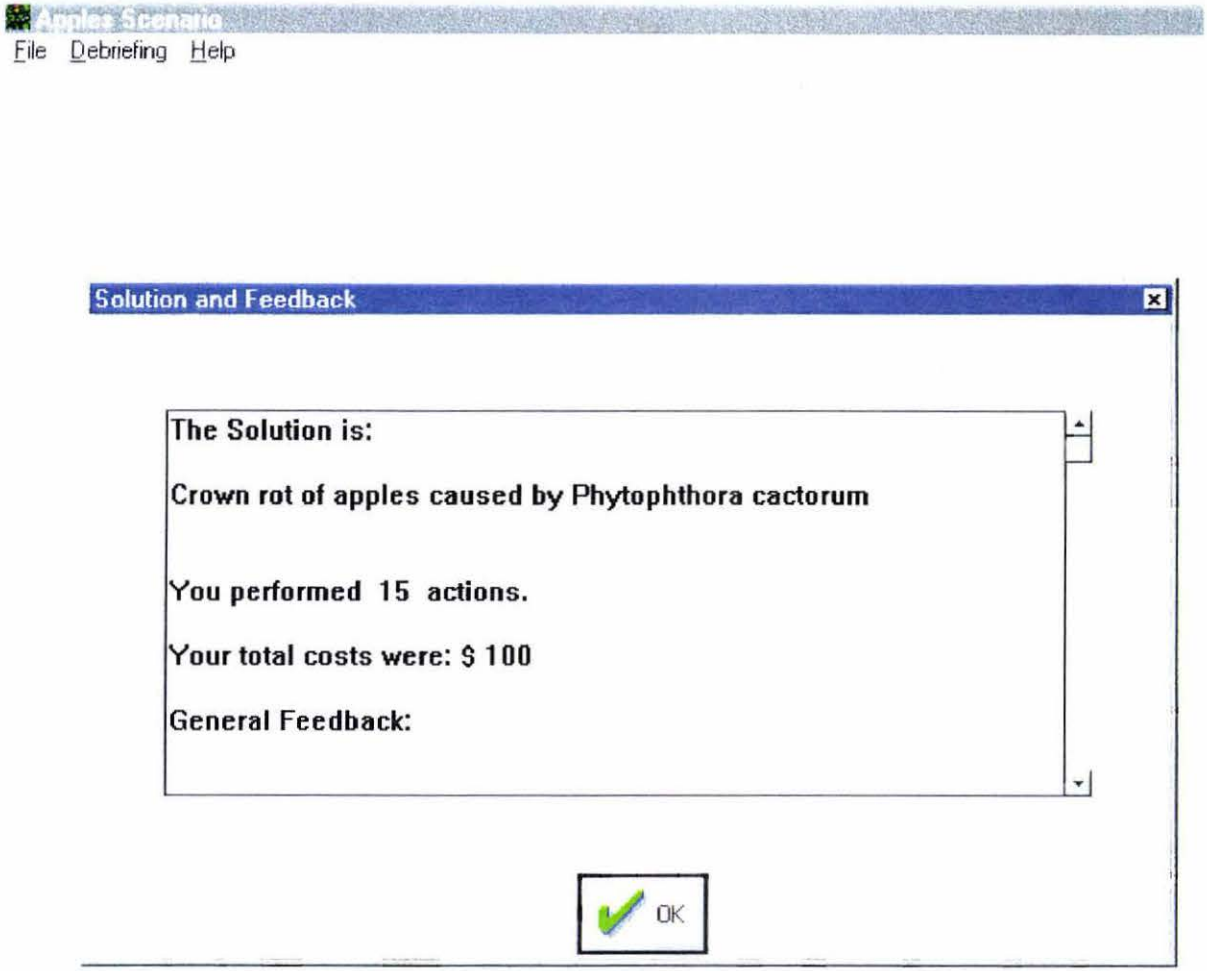
Lab procedures and referrals available while playing



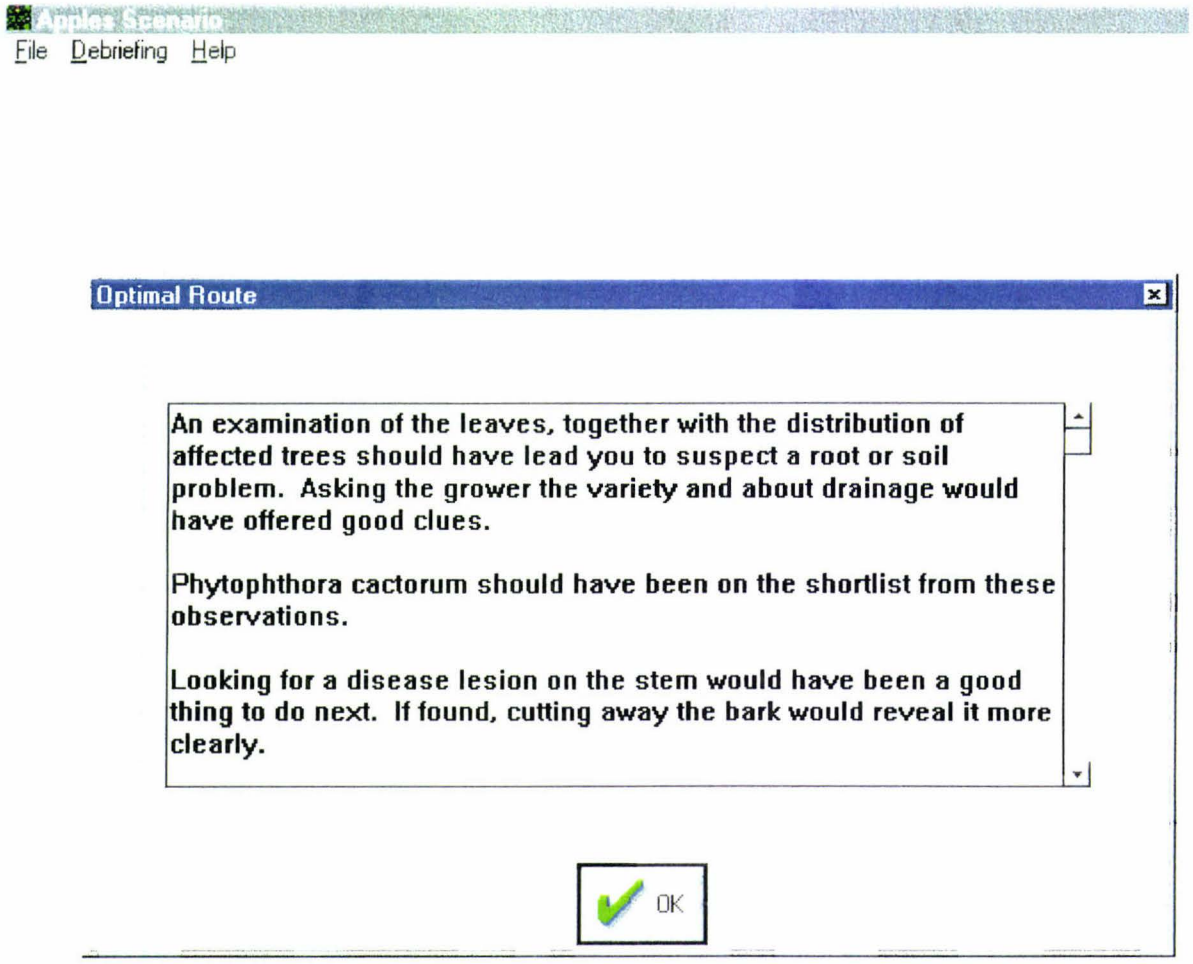
Testing for nutrients while playing



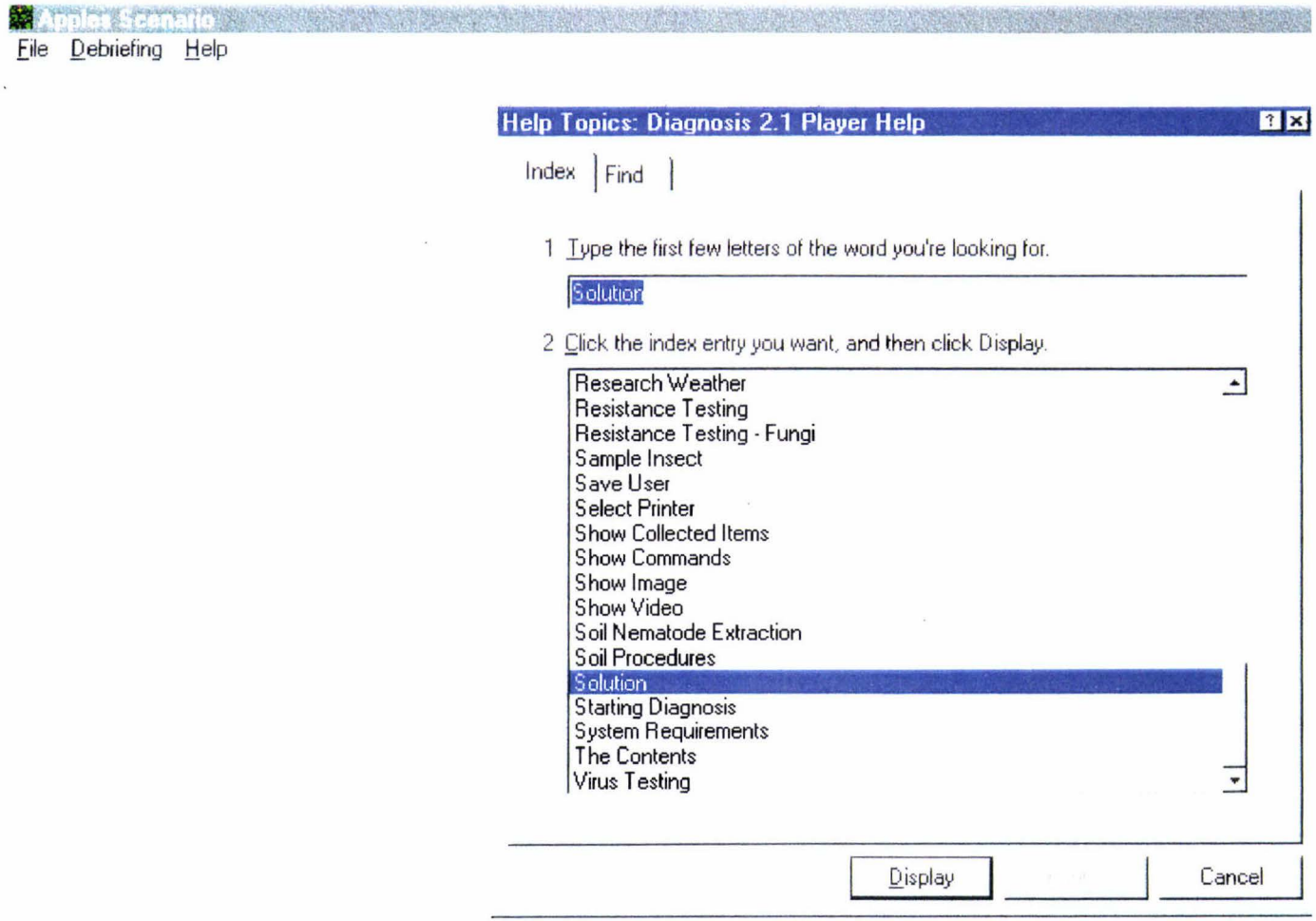
Entering initial diagnosis while playing



System displaying solution after player entering final diagnosis



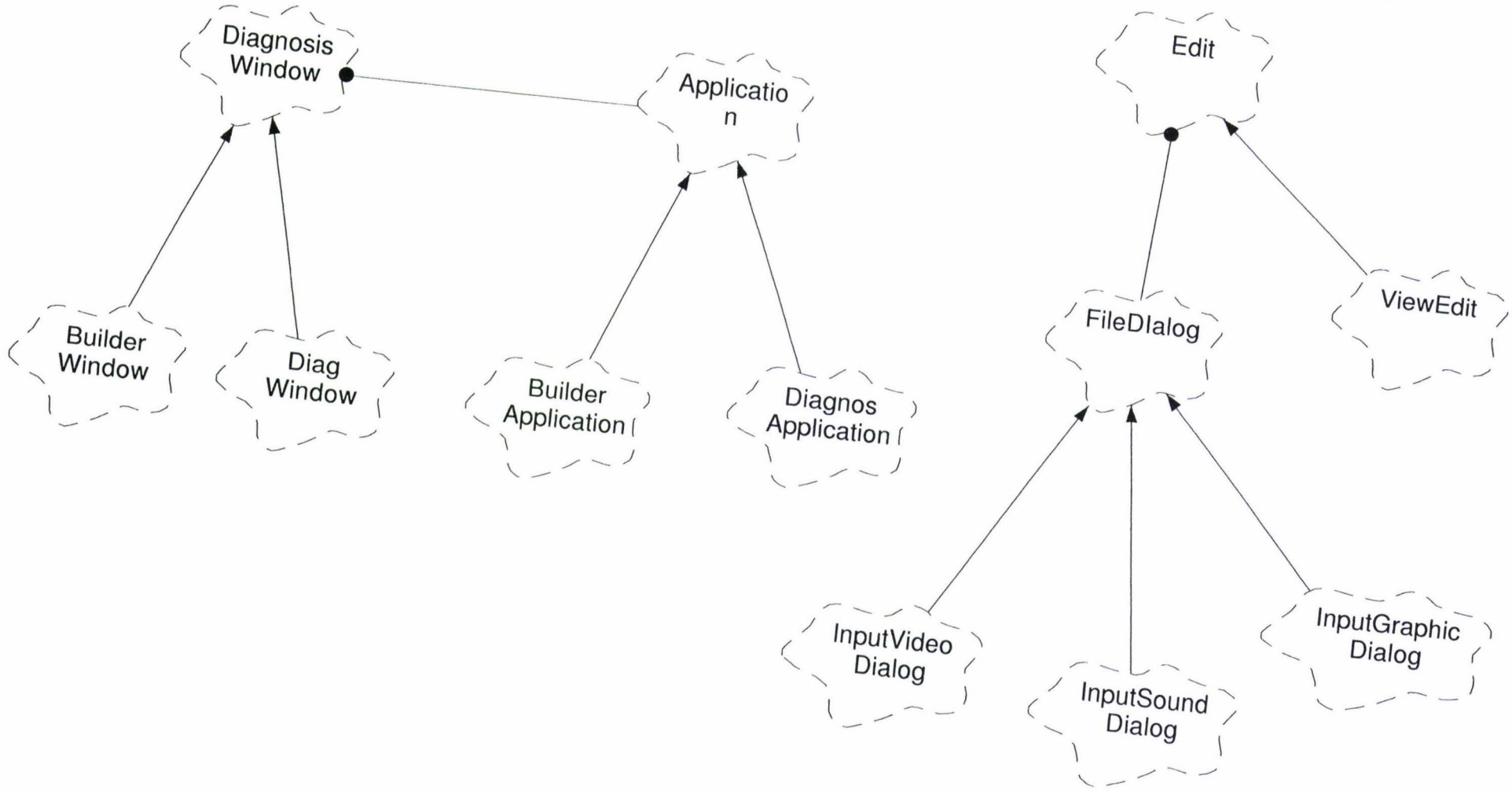
System displaying optimal path for solution after player entering final diagnosis

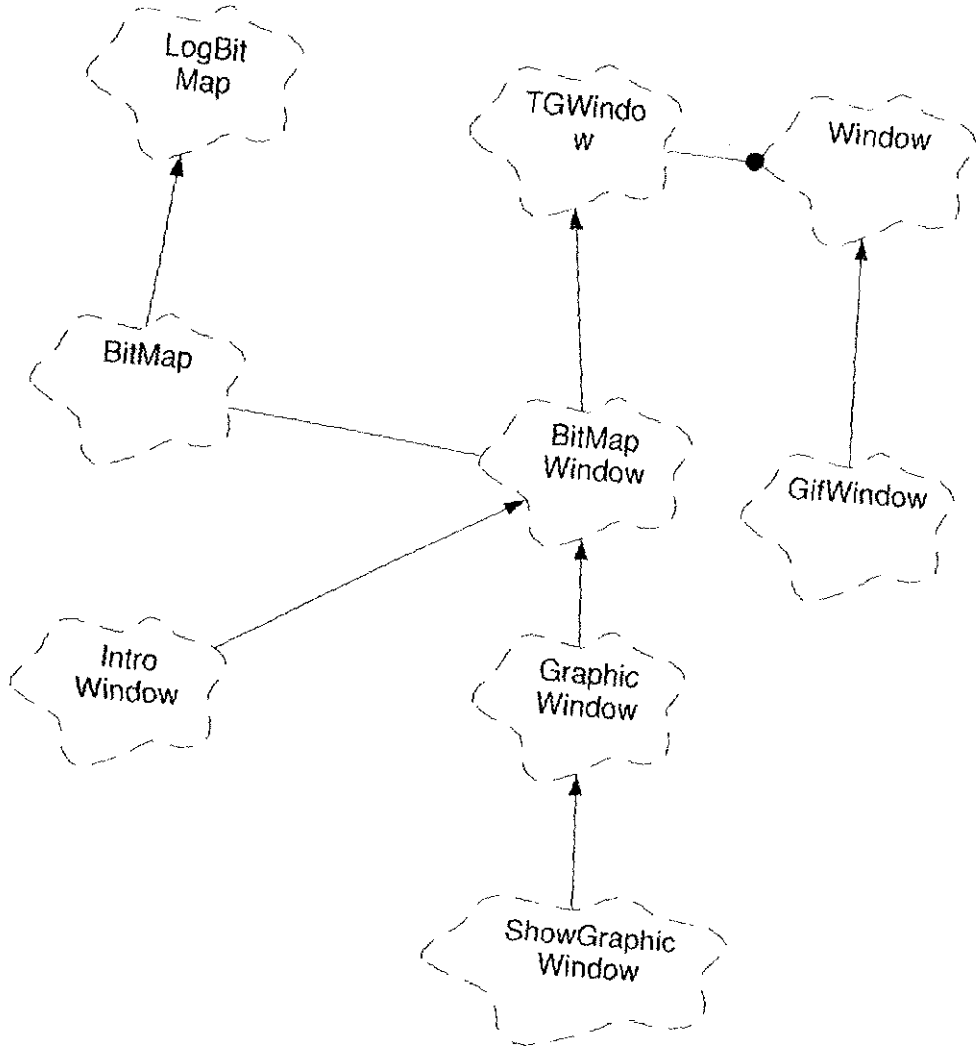


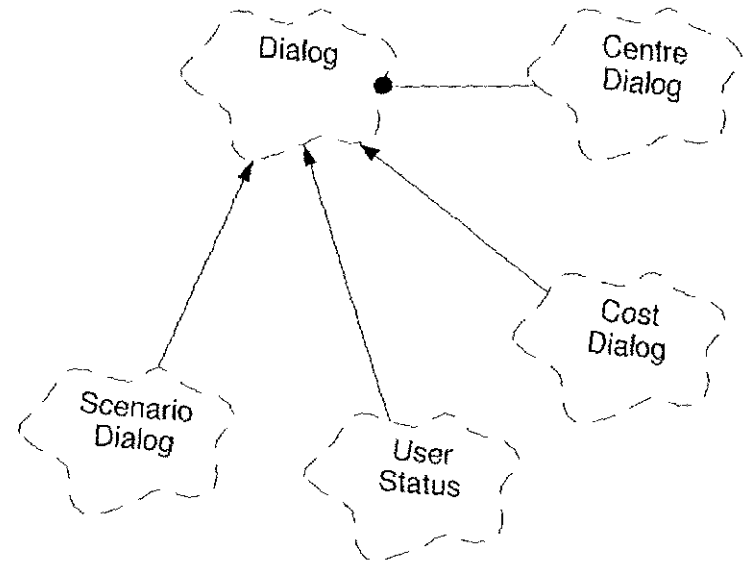
Diagnosis help

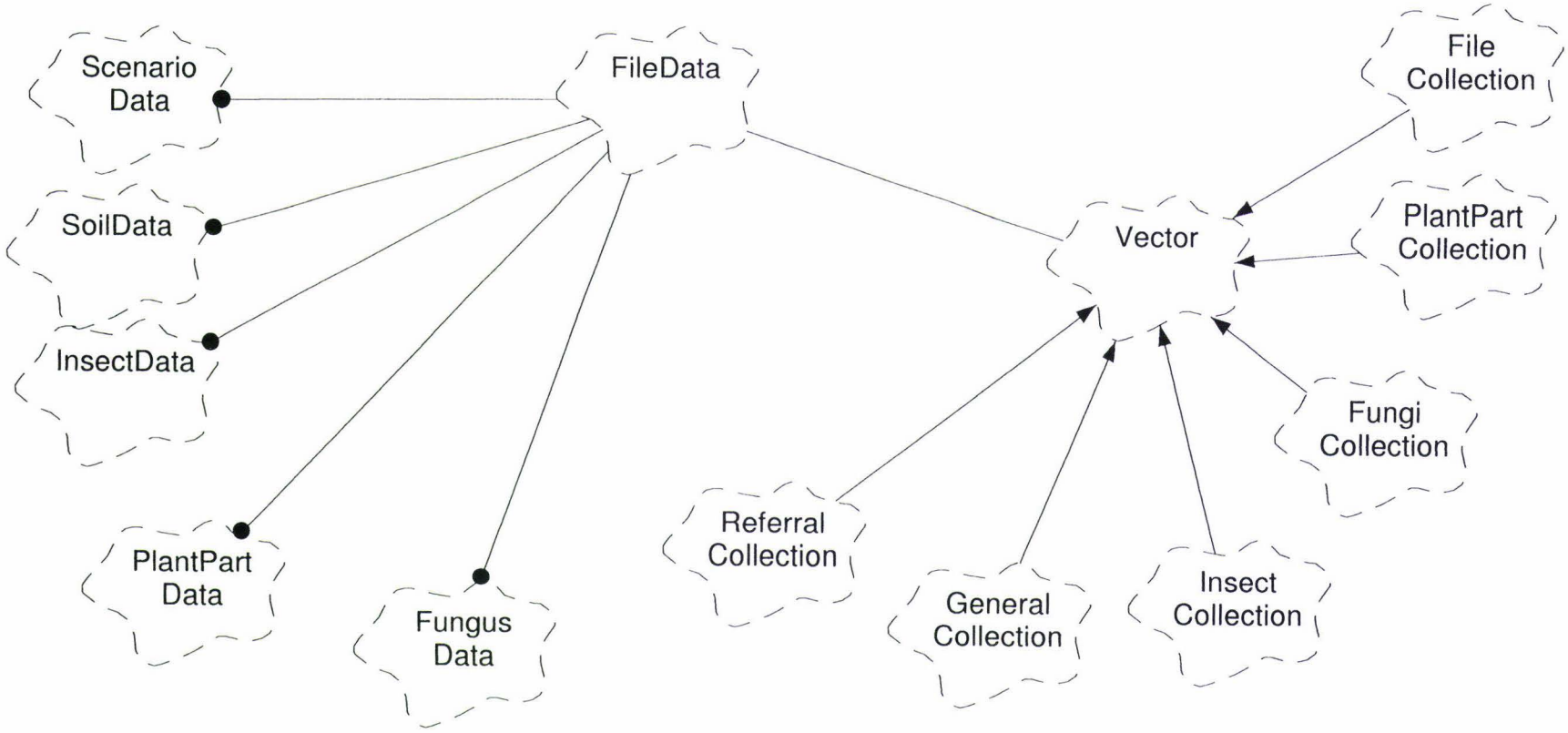
Appendix E

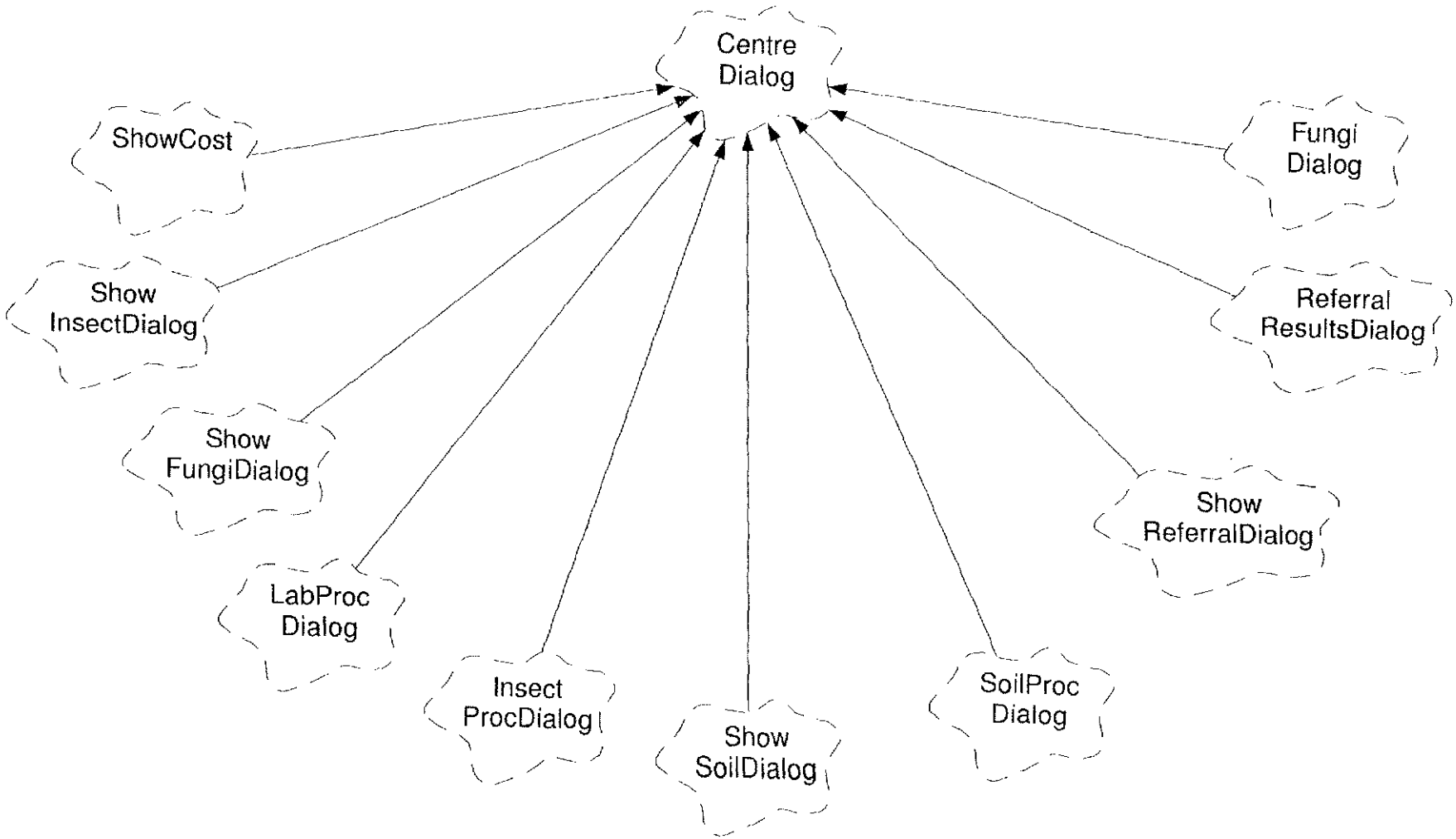
Object Diagram for Diagnosis

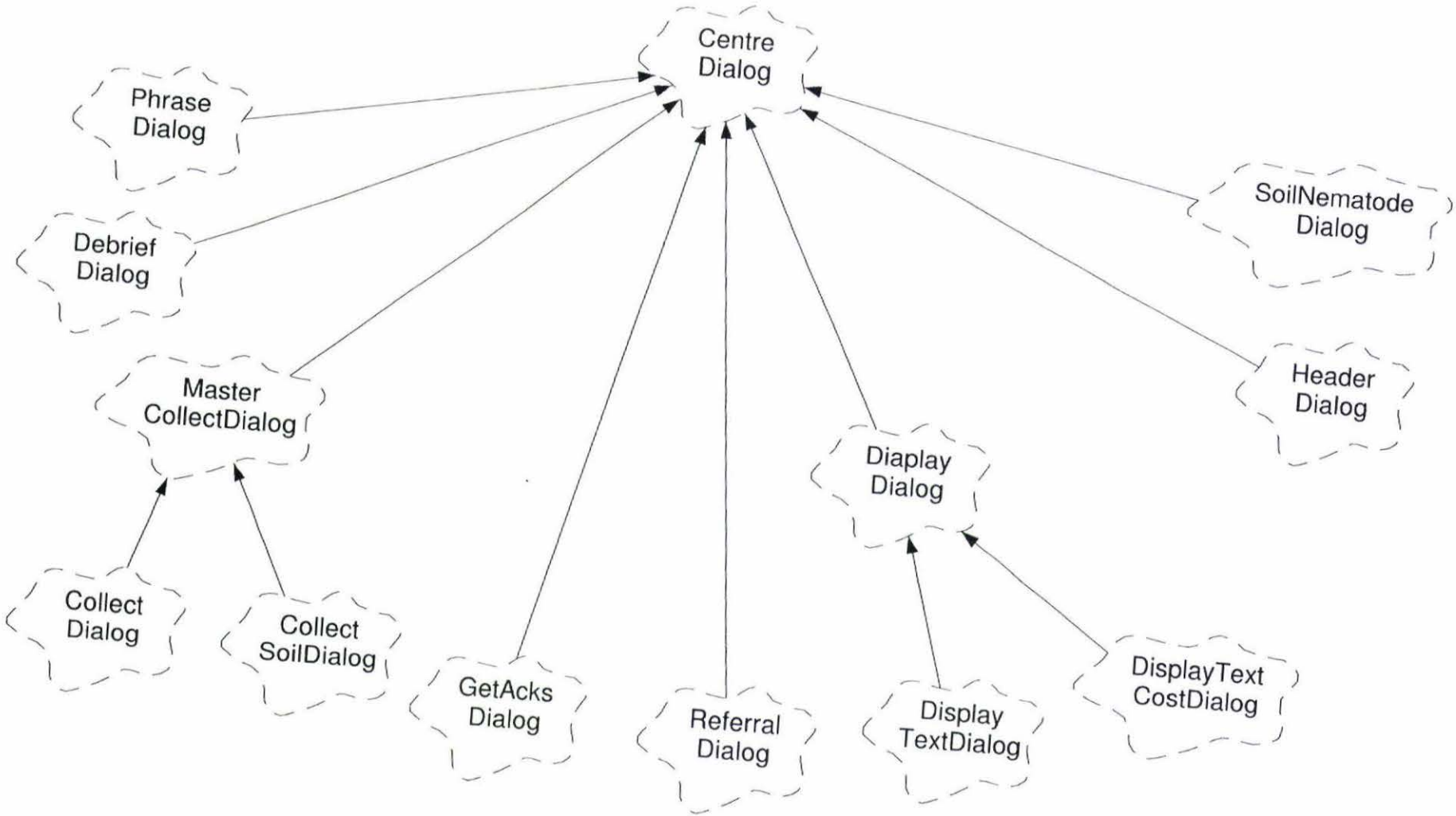

Diagnosis Window
Applicatio n
Builder Window
Diag Window
Builder Application
Diagnos Application
Edit
FileDIalog
ViewEdit
InputVideo Dialog
InputSound Dialog
InputGraphic Dialog

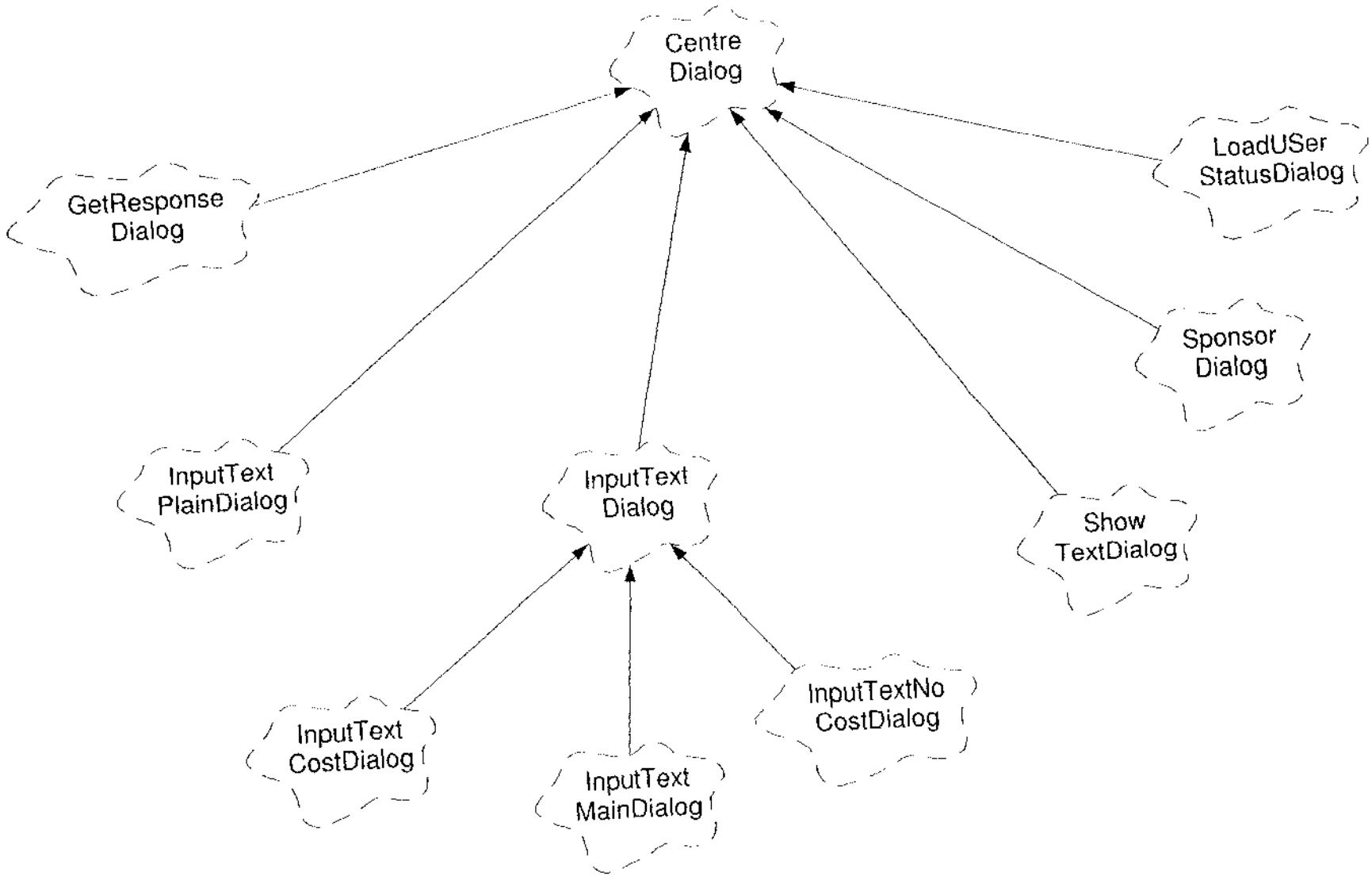

LogBit Map
TGWindo w
Window
BitMap
BitMap Window
GifWindow
Intro Window
Graphic Window
ShowGraphic Window


Dialog
Centre Dialog
Cost Dialog
Scenario Dialog
User Status


Scenario Data
SoilData
InsectData
PlantPart Data
Fungus Data
FileData
Referral Collection
General Collection
Insect Collection
Fungi Collection
PlantPart Collection
File Collection
Vector


Centre Dialog
ShowCost
Show InsectDialog
Show FungiDialog
LabProc Dialog
Insect ProcDialog
Show SoilDialog
SoilProc Dialog
Show ReferralDialog
Referral ResultsDialog
Fungi Dialog


Centre Dialog
Phrase Dialog
Debrief Dialog
Master CollectDialog
Collect Dialog
Collect SoilDialog
GetAcks Dialog
Referral Dialog
Diaplay Dialog
Display TextDialog
DisplayText CostDialog
SoilNematode Dialog
Header Dialog


Centre Dialog
GetResponse Dialog
LoadUSer StatusDialog
Sponsor Dialog
InputText PlainDialog
InputText Dialog
Show TextDialog
InputText CostDialog
InputText MainDialog
InputTextNo CostDialog

Name : **BitMap**

Fields :

Bitspixel;
DIBInfo;
DIBInforSize;
ShouldFree;
Handle;
Hbits;
OldBitMap;

Operations :

PvoidCopy;
CopyFrom;
AsBitmap;
AsDDB;
Draw;
ColorTableSize;
FastDraw;
FreeDDB;
GetBits;
GetPallette;
ParseDIBInfo;
ParseDDBInfo;
SetBrush;
SetDBIInof;
SetPen;
Size;
SaveBitmapFile;
LoadBitMapFile;
LoadBitmapResourse;
Write;
Read;
isA;
nameOf;
GetDDB;
SetMonoColors;
AttachPort;
DetachPort;
ConvertClipDDB;

Name : **BitmapWindow**

Fields :

Result;
GraphicH;
GraphicW;
FileName;

Operations :

BitmapExit;
SetupWindow;

Name : **BlankWindow**

Fields :

Operations :

WMTimer

Name : **BuilderApplication**

Fields :

Operations :

InitMainWindow;

Name : **BuilderWindow**

Fields :

FileName;
ExportFile;
FilenameSpecified;
ScenarioDirty;

Operations :

SetupWindow;
Insect;
Collect;
PlantpartDetails;
GeneralDetails;
DefCommandProc;
NewScenarioProc;
LoadScenarioProc;
SaveScenarioProc;
SaveScenarioAsProc;
ExportSceneAsTextProc;
ShowPhrase;
AddScenarioCredits;
SponsorDetails;
SelectBackground;
SelectLabBackground;
HelpAbout;
GetActionMenu;
CanClose;
GetClassname;
GetWindowClass;
SaveAs;
GetPhrase;
ChooseDebrief;
Cut;
Examine;
TickActionMenu;
TickSubActionMenu;
ExamineSoil;
CollectSoil;
CollectPLant;
CheckSubMenuCollect;
CheckSubMenu;
CheckMenu;
TickLevel;

Name : **CentreDialog**

Fields :

Operations :

SetupWindow;

Name : **CollectDialog**

Fields :

Plant;
MicroCheck;
AgarCheck;
HumidCheck;
NematodeCheck;
VirusCheck;
BackyCheck;
NutrientCheck;
PestCheck;

Operations :

Microscope;
Agar;
Humidity;
Nematode;
Nutrient;
Virus;
Pesticide;
Bacteria;

Name : **CollectionDialog**

Fields :

ObjectList;
InsectList;
TheUser;
Scenario;

Operations :

SetupWIndow;
Close;

Name : **CollectLabDialog**

Fields :

Operations :

HandleINsectListMsg;
HandleObjectListMsg;

Name : **CollectSoilDialog**

Fields :

CollectSoil;
BaitingCheck;
SoilNemoChack;
NutrientCheck;
PestCheck;

Operations :

SoilNematode;
ExamineLab;
Nutrient;
Pesticide;
Baiting;
SetupWindow;

Name : **DiagnosApplication**

Fields :

Operations :

InitMainWindow;

Name : **DiagWindow**

Fields :

Diagnoses;
Display;
DisplayNoCost;
Headers;
Caption;
TheUser;
Printer;
CollectionDisplay;
LabCollection;
FileName;
UserSateFile;
UserFIle;
Infile;
DisplayConnection;
InitialDisplay;
UserResultSaved;

Operations :

ExamineSoil;
CollectSoil;
ExamineCut;
SaveGeneralDetails;
Insect;
ShowCollection;
LoadScenario;
LoadUserState;
ChoiceGoLab;
NotepadOpen;
FileSelPrint;
FilePrint;
FileExit;
HelpAbout;
FinalDiagnosis;
GoSolutionMenu;
OptimumRoute;
SolutionFeedback;
Action;
SolutonExit;
LabExit;
LabHelp;
AnotherScenario;
MainHelp;
FeedbackHelp;
Save;
SaveAs;
ShowStatus;
AboutScenario;
GetActionMenu;
CanClose;
SetupWindow;
Collect;
GetClassName;
GetWindowClass;
MakeLab;
StartAgain;
ShowLabCollection;
SetBackground;
StoreMenuHeaders;
PlantpartDetails;
Initialise;
ItemsCollected;
DefCommandProc;
PurgeMenu;
CleanActionMenu;
PruneSubactionMenu;
CleanMainMenu;
EnableMenus;
EnableActionMenu;
EnableSubactionMenu;

Name : **DiagnosisWindow**

Fields :

SBuffer;
Scenario;
ScenarioLoaded;
Crop;
Exedir;

Operations :

FileExit;
HelpLicence;
CropType;
LoadCropMenu;
GetActionMenu;
CollectEmpty;
CollectSoilEmpty;
CleanGeneralMenu;
TickGeneralMenu;

Name : **DiaplyDialog**

Fields :

GraphicFile;
VideoFile;
SoundFile;
PictureCaption;

Operations :

SetupWindow;
View;
Video;
Sound;

Name : **DisplayFilePhraseDig**

Fields :

SelectedPhrase;
ThePhrases;
Apples;
Files;
FileMask;
PathName;

Operations :

GetPhrase;
HandlePhraseMsg;
HandleDList;
OK;
DefChildProc;
Help;
SetupWindow;
CanCLose;
FileName;
FilePhrase;
UpdateListBoxes;

Name : **DisplayTextCostDialog**

Fields :

Operations :

DisplayHelp;

Name : **FileCollection**

Fields :

Operations :

FreeAll;
FirstThat;

Name : **FileData**

Fields :

Operations :
FileName;
FilePhrase;

Name : **FileNameDialog**

Fields :
Name;
DgnPath;
PathLen;
InFile;

Operations :
OK;

Name : **FungiDialog**

Fields :
Fungi;
Counter;
Index;
ExamineCheck;
PathoCheck;
ResistCheck;
IndentCheck;

Operations :
Examine;
Pathogenicity;
Identify;
Resistance;
Previous;
Next;
Help;
DeleteAll;
OK;
SetTicks;
SetupWindow;

Name : **GetAcksDialog**

Fields :

Operations :
Help;

Name : **GetDataDialog**

Fields :
HelpCatg;
Diagnosis;
Justification;
Recommendation;

Operations :
Help;

Name : **GifWindow**

Fields :

Operations :
SetupWindow;

Name : **GraphicWindow**

Fields :

Operations :
SetupWindow;
Quit;

Name : **InputGraphicDialog**

Fields : TheFile

Operations :
View;
DefChildProc;
GraphicFileName;
SetupWindow;

Name : **InputGraphicWindow**

Fields :

Operations :
ViewGraphic;
Help;
ClearEntry;
DefChildProc;

Name : **InputSoundDialog**

Fields :

Operations :
PlaySound;
Help;

Name : **InputVideoDialog**

Fields :

Operations :
ViewVideo;
Help;

Name : **InsectCollectDialog**

Fields :
Insect;
MicroCheck;
RearingCkeck;
ResistCheck;
IdentCheck;

Operations :
MicroscpoeInspect;
Rearing;
Resistance;
Indentify;
SetupWIndow;

Name : **InsectDialog**

Fields :
Insects;
Counter;
InsectIndex;
ExamineCheck;
CollectCheck;
SampleCheck;

Operations :
Collect;
Examine;
Sample;
Previous;
Next;
Help;
DeleteAll;
OK;
SetupWIndow;
SetTicks;

Name : **InsectProcDialog**

Fields :

Insect;
TheUser;
Label;

Operations :

SetupWIndow;
Microscope;
Rearing;
Resistance;
PestID;

Name : **IntroWindow**

Fields :

Operations :

WTTimer;
WMLButtonDown;

Name : **LabProcDialog**

Fields :

Plant;
TheUser;
Label;

Operations :

SetupWindow;
Microscope;
Agar;
Humidity;
Nematode;
Nutrient;
Virus;
Pesticides;
Bacteria;

Name : **LogoWIndow**

Fields :

Operations :

WMLButtonDown;

Name : **NoteWindow**

Fields :

Operations :

NotepadExit;
NotepadHelp;
CanClose;
GetWIndowClass;
MenuString;
SetupWIndow;
EnableNotepad;
NeatenMenuItem;

Name : **PathogenDialog**

Fields :

Pathogens;
Counter;
Index;

Operations :

Examine;
Identify;
Resistance;
Previous;
Next;
Help;
SeleteAll;

Name : **ReferralDialog**

Fields :

Name;
Response;
Referrals;
NewRefs;
OriginalRefs;

Operations :

SetupWindow;
OK;
Help;
HandleListMsg;
FindReferral;
Delete;
Insert;
Clear;
ClearAll;
UpdateListBox;
DoubleClick;

Name : **ReferralResultDialog**

Fields :

TheReferrals;
Text;

Operations :

SetupWindow;
Help;

Name : **ShowCost**

Fields :

Operations :

Yes;
No;

Name : **ShowFungiDialog**

Fields :

User;
Name;
FungiCollected;
Fungi;
Counter;
Index;

Operations :

Examine;
Pathogenicity;
Identify;
Resistance;
Previous;
Next;
Help;
DeleteAll;
SetupWindow;

Name : **ShowInsectDialog**

Fields :

Insects;
Counter;
InsectIndex;
User;
Name;
InsectCollected;

Operations :

Collect;
Examine;
Sample;
Previous;
Next;
OK;
DeleteAll;
SetupWindow;

Name : **ShowReferralDialog**

Fields :
TheUser;
TotalCost;
ItemsList;
Items;
TickCount;

Operations :
TestAll;
Help;
OK;
DoubleClick;
HandleListMsg;
FindReferral;
UpdateListBox;

Name : **ShowSoilDialog**

Fields :
ShowSoil;

Operations :
Centrifuge;
Baermann;
SetupWindow;

Name : **SoilNematodeDialog**

Fields :
CentrifugeCheck;
BaermannCheck;
MySoil;

Operations :
Centrifuge;
Baermann;
SetupWindow;

Name : **SoilProcDialog**

Fields :
Soil;
TheUser;
Label;

Operations :
SetupWindow;
ExamineLab;
SoilNematode;
Nutrient;
Pesticide;
Baiting;

Name : **StartDialog**

Fields :
Name;
Number;

Operations :
OK;
Skip;

Name : **TextDisplay**

Fields :

Operations :
DisplayHelp;

Name : **UserStatus**

Fields :
TheUser;
Cost;
Actions;

Operations :
SetupWindow;

Appendix F

Algorithmic description of proposed changes to Diagnosis

Algorithm for proposed Builder module

Variable name	Meaning
PlayerType	Refers to Good/Average/Poor player
ProblemCause	Refers to Environmental Problems, Problems caused by pathogens, Nutrient Problems etc.
ProblemType	Refers to Hard, Medium or Easy.
CropType	Refers to crops classified in Chapter 6
HelpLevel	Refers to with help, with opinion, without help

START

Display Development Hierarchy
Get ***Option***
CASE ***Option***

// Module 1 : Player Model

1: **IF** Create Test **OR** Modify Test
{
Explain test strategy
Get test questions
Get Score range for each ***PlayerType***
Explain ErrorType1, ErrorType2
Get number of ***ErrorType2, ErrorType1*** to goto next level
From easy to medium
From medium to hard
Get number of problems to be solved to goto next level
Get number of tries for ***InitialDiagnosis*** and ***InitialRemedy***
}

// Module 2 : Initial Scenario

2: Display crop types
Get ***CropType***
Display problem types
Get ***ProblemType***
Display problem causes
Get ***ProblemCause***
Construct initial scenario

{
Add Text
{
Mark Clues in the text
Mark misleading-clues in the text
}
Add Audio
Add Video
Add Picture
}

// Module 3 : Field Observation

3: Decide parts depending on the ***CropType***
Construct field observations
REPEAT
Mark important clues for each part
Get Help Message if it is over looked
Get Expert Opinion on that particular situations
UNTIL (No more clues) **AND** (no more parts)

REPEAT
Mark misleading clues
Get Help Message why it is not the cause
Get Expert Opinion on the situations
UNTIL (No more mis-leading clues) **AND** (no more parts)

// Module 4 : Initial Diagnosis and Remedy

4: Display the list of ***InitialDiagnosis***
Get the necessary ones for the problem being built
Get the new ones if any
Get the correct ***InitialDiagnosis***
Get reason why it is right
Get expert opinion on the right choice

REPEAT
Get the misleading choice
Get help message for why it is not right
Get Expert opinion on why it cannot be right
UNTIL (no more misleading choice)

Display the list of ***InitialRemedy***
Get the necessary ones
Get the new ones if any
Get the correct ***InitialRemedy***
Get reason why it is right
Get expert opinion on the right choice

REPEAT
Get the misleading choice
Get help message for why it is not right
Get Expert opinion on why it cannot be right
UNTIL (no more misleading choice)

// Module 5 : Lab Procedures

5: Construct Lab Procedures

REPEAT
Mark important clues
Get Help Message if its over looked
Get Expert Opinion on the situation
Mark misleading clues
Get Help Message why its not the cause
Get Expert Opinion on such situations
Mark Optimal path node
UNTIL (No more procedures)

```
// Module 6 : Delayed Feedback

6:      Display widow for debriefing
        Get Debriefing - selected from the errors and ErrorType1 made.
        Create a list of all possible nodes (depending on crop type)
        Mark nodes in the optimal path(s)


// Module 7 : Reference Book

7:      About reference book
        Create Reference Book
        IF Add/Delete/Modify Entry
                {
                ADD/Delete/Modify entry
                }
        IF Hide reference book
                {
                Turn access OFF to Reference Book
                }


// Module 8 : Menu Structure

8:      Display Menu Hierarchy
        IF Done
                {
                        highlight menu option
                }

END CASE;

END.
```

Algorithm for proposed Diagnosis (Player) Module

Variable name	Meaning
Questions_in_Series	Number of questions in a series.
Min_CorrectlySolved_Questions-ToBeGood	Number of correctly solved problems in a series to be above average in the series
Questions_in_Series	Number of questions in a series
Min_CorrectlySolved_Questions-ToBeGood	Number of correctly solved problems in a series to be above average in the series
Min_CorrectlySolved_Questions_ToBe Avg	Number of correctly solved problems in a series to be average in the series
Min_CorrectlySolved_Questions_ToBe Poor	Number of correctly solved problems in a series to be below average in the series
Max_ErrorTYpe1_ToBeGood	Maximum number of ErrorType1 one can make to be a good player
Max_ErrorType2_ToBeGood	Maximum number of ErrorType2 one can make to be a good player
Max_ErrorType1_ToBeAvg	Maximum number of ErrorType1 one can make to be an average player
Max_ErrorType2_ToBeAvg	Maximum number of ErrorType2 one can make to be an average player
MaxTries	Maximum number of tries for Initial diagnosis and Initial Remedy

START

Display Session types (Assessment/ Practice)
Select ***SessionType***

IF ***SessionType*** = Practice
{
Give access to Notepad
Give access to ReferenceBook
}

Check Previous Session for User

IF Yes
{
Load Previous Session detail
Load Previous Session problem-list
Ask if the user want to continue (Yes/No)
IF Yes
{
Set initial values from Previous Session
Categorical Presentation
}
}
Want to take the test? (Yes/No)
IF Yes
{
Display test
Calculate score
Combine Previous session detail (optional)
Determine ***PlayerType***
Create solution path list
Categorical Presentation
}
ELSE
Standard Presentation

Save session details

END.

Standard Presentation

```
REPEAT
        Display Problem types (Easy/Medium/Hard)
        Choose ProblemType
        Display Problem causes
        Choose ProblemCause
        Select problem from problem-list
        Present Initial Scenario
        Let interrogate Initial Scenario
        Display Initial List of Diagnoses
        Get InitialDiagnosis
        IF Correct
                {

                Display Initial list of Remedy
                Get Initial Remedy
                IF Correct      // Both Diagnosis and Remedy are correct
                        {
                        Get final diagnosis
                        Show solution
                        }
                 ELSE           // Diagnosis correct and Remedy wrong

                        REPEAT
                                Display Warning Message
                                Get Initial Remedy
                        UNTIL (Correct) OR (Try  > MaxTries)

        ELSE                    // Diagnosis wrong
                {
                 Ask if they want help

                IF Yes
                        {

                        CASE

                        ProblemType  =  Easy :
                                        PlayerType  =  POOR
                                        HelpLevel = 1

                        ProblemType  =  Medium :
```

```
                                    PlayerType = AVERAGE
                                    HelpLevel = 2

            ProblemType = Hard :

                                    PlayerType = GOOD
                                    HelpLevel = 2
            END CASE;

            Categorical Presentation
            }
      ELSE
            {
            REPEAT
                  Display Warning Message
                  Get Initial Diagnosis
            UNTIL (Correct) OR (Try > MaxTries)

            REPEAT
                  Display Warning Message
                  Get Initial Remedy
            UNTIL (Correct) OR (Try > MaxTries)
            }

Select another problem

UNTIL (Exit) OR (No more problems)
```

Categorical Presentation

START

REPEAT

No_of_Questions = 0
Select InitialValues
Solved_Good= 0
Solved_Poor = 0
Solved_Average= 0
PreviousSeries_Good = 0
PreviousSeries_Avg = 0
PreviousSeries_Poor = 0
NextSeries_Good = 0
NextSeries_Avg = 0
NextSeries_Poor = 0
CurrentSeries= 0

REPEAT

No_of_Questions = No_of_Questions +1

Select problem of type ***ProblemType*** caused by ***ProblemCause***
Present with ***HelpLevel***
Present Problem

IF (ErrorType1< = Max_ErrorType1_ToBeGood) **AND**
(ErrorType2< = Max_ErrorType2_ToBeGood)
{
Solved_Good = 1
}
ELSE

IF (Max_ErrorType1_ToBeGood< ErrorType1<=Max_ErrorType1_ToBeAvg)
AND
(Max_ErrorType2_ToBeGood<ErrorType2<=Max_ErrorType2_ToBeAvg)
{
Solved_Average = 1
}
ELSE
Solved_Poor = 1

```
// Update series results

IF Solved_Good= 1
        {
        PreviousSeries_Good  = PreviousSeries_Good +1
        }

IF Solved_Average= 1
        {
        PreviousSeries_Avg  = PreviousSeries_Avg +1
        }

IF Solved_Poor= 1
        {
        PreviousSeries_Poor  = PreviousSeries_Poor +1
        }

UNTIL (Exit) OR (No_of_Questions>= Max_Questions_in_Series)

// At the end of series transfer values from next to previous

IF NextSeries_Good  = NextSeries_Avg  = NextSeries_Poor  = 0
// If not first series in the play

        {
          NextSeries_Good  = PreviousSeries_Good
          NextSeries_Avg  = PreviousSeries_Avg
          NextSeries_Poor  = PreviousSeries_Poor
        }

        Update PlayerType

        CASE PlayerType

                POOR: Select NextProblem_P
                AVERAGE: Select NextProblem_A
                GOOD:Select NextProblem_G
        END CASE;

        DO ErrorCheck

UNTIL (Exit) OR (No more questions)

END.
```

Error Count

START
Display help message for ***HelpLevel***
    IF unnecessary action
        {
        ErrorType2 = ErrorType2+1
        }
    ELSE
        IF wrong action
            {
            ErrorType1 = ErrorType1+1
            }
Mark node in solution path

END.

Select InitialValues

START

    CASE *PlayerType*

        POOR: ***ProblemType*** = Easy
            ProblemCause = 1
            HelpLevel = 1
        AVERAGE: ***ProblemType*** = Medium
            ProblemCause = 1
            HelpLevel = 2
        GOOD: ***ProblemTyp***e = Hard
            ProblemCause = 1
            HelpLevel = 3

    END CASE;

END.

Select NextProblem_ P (NextProblem for Poor Player)

START
// One who did well in the current series and did well in the previous series
// Assumption: Good at theory and practice different cause, much less help,
//harder problem

CASE

(*PreviousSeries_Good* > = *Min_CorrectlySolved_Questions-ToBeGood*)
AND (*NextSeries_Good* >= *Min_CorrectlySolved_Questions-ToBeGood*) :
{
ProblemCause = *ProblemCause*+1
ProblemType = *ProblemType*+1
HelpLevel = *HelpLevel*+1
}

// And did ok in the previous series. Assumption: Needed more help or didn't
// know the area. Different cause, less help, different type problem

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions-ToBeGood*) **AND**
(*Min_CorrectlySolved_Questions-ToBeGood* > *NextSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg) :
{
ProblemCause = *ProblemCause*+1
HelpLevel = *HelpLevel*+1
}

// And did below average in the previous series. Assumption: Needed more
//help and didn't know the area. Different cause, less help, same type problem

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions-ToBeGood*) **AND**
(*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >=
Min_CorrectlySolved_Questions_ToBePoor):
{
ProblemCause = *ProblemCause*+1
HelpLevel = *HelpLevel*
}

// One who did ok in the current series and did well in the previous series
// Assumption: and didn't know the area. Different, same type problem

(*Min_CorrectlySolved_Questions-ToBeGood* > *PreviousSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*) **AND** (*NextSeries_Good* >= *Min_CorrectlySolved_Questions-ToBeGood*) :

{
ProblemType = *ProblemType*+1
HelpLevel = *HelpLevel*+1
}

// And did ok in the previous series

(*Min_CorrectlySolved_Questions-ToBeGood* > *PreviousSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*) **AND**
(*Min_CorrectlySolved_Questions-ToBeGood* > *NextSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg* :

{
ProblemType = *ProblemType*+1
ProblemCause = *ProblemCause*+1
}

// And did below average in the previous series

(*Min_CorrectlySolved_Questions-ToBeGood* > *PreviousSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*) **AND**
(*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) :

{
ProblemCause = *ProblemCause*+1
HelpLevel = *HelpLevel*+1
}

// One who did poor in the current series

// And did well in the previous series

(*Min_CorrectlySolved_Questions_ToBeAvg* > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) **AND** (*NextSeries_Good* >= *Min_CorrectlySolved_Questions-ToBeGood*) :

{
ProblemType = *ProblemType*+1
HelpLevel = *HelpLevel*+1
}

// And did ok in the previous series

(*Min_CorrectlySolved_Questions_ToBeAvg* > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) AND
(*Min_CorrectlySolved_Questions-ToBeGood* > *NextSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*):

{
ProblemType = *ProblemType*+1
HelpLevel = *HelpLevel*+1
}

// And did below average in the previous series

(*Min_CorrectlySolved_Questions_ToBeAvg* > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) AND
(*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*):

}
ProblemType = *ProblemType*
ProblemCause = *ProblemCause*+1
}

END CASE
END.

Select NextProblem_G (NextProblem for Good Player)

START

// One who did well in the current series

// And did well in the previous series. Assumption: Good at theory and practice
// Different cause, much less help, harder problem

CASE
(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) AND
(*NextSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) :

{
ProblemCause = *ProblemCause*+1
ProblemType = Hard
HelpLevel = 3
}

// And did ok in the previous series. Assumption: Needed more help or didn't
//know the area. Different cause, less help, different type problem

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) **AND**
(*Min_CorrectlySolved_Questions_ToBeGood* > *NextSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg):

{
ProblemType = Hard
ProblemCause = ***ProblemCause***+1
HelpLevel = 2
}

// And did below average in the previous series. Assumption: Needed more
//help and didn't know the area. Different cause, less help, same type problem

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) **AND**
(*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >=
Min_CorrectlySolved_Questions_ToBePoor):

{
ProblemType = Hard
HelpLevel = 2
}

// One who did ok in the current series

// And did well in the previous series. Assumption: and didn't know the area
// Different, same type problem

(*Min_CorrectlySolved_Questions_ToBeGood* > *PreviousSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg) **AND** (*NextSeries_Good* >=
Min_CorrectlySolved_Questions_ToBeGood) :

{
ProblemCause = ***ProblemCause***+1
ProblemType = Medium
HelpLevel = 2
}

// And did ok in the previous series

(*Min_CorrectlySolved_Questions_ToBeGood* > *PreviousSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg) **AND**
(*Min_CorrectlySolved_Questions_ToBeGood* > *NextSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg) :

}
ProblemType = Medium
ProblemCause = Same
HelpLevel = 3
}

// And did below average in the previous series

(***Min_CorrectlySolved_Questions_ToBeGood*** > ***PreviousSeries_Avg*** >= ***Min_CorrectlySolved_Questions_ToBeAvg***) **AND**
(***Min_CorrectlySolved_Questions_ToBeAvg*** > ***NextSeries_Poor*** >= ***Min_CorrectlySolved_Questions_ToBePoor***):
{
ProblemType = Medium
ProblemCause = ***ProblemCause***+1
HelpLevel = 3
}

// One who did poor in the current series

// And did well in the previous series

(***Min_CorrectlySolved_Questions_ToBeAvg*** > ***PreviousSeries_Poor*** >= ***Min_CorrectlySolved_Questions_ToBePoor***) **AND** (***NextSeries_Good*** >= ***Min_CorrectlySolved_Questions_ToBeGood***) :
{
ProblemType = Medium
ProblemCause = ***ProblemCause***+1
HelpLevel = 3
}

// And did ok in the previous series

(***Min_CorrectlySolved_Questions_ToBeAvg*** > ***PreviousSeries_Poor*** >= ***Min_CorrectlySolved_Questions_ToBePoor***) **AND**
(***Min_CorrectlySolved_Questions_ToBeGood*** > ***NextSeries_Avg*** >= ***Min_CorrectlySolved_Questions_ToBeAvg***):
{
ProblemType = Medium
ProblemCause = ***ProblemCause***+1
HelpLevel = 2
}

// And did below average in the previous series

(*Min_CorrectlySolved_Questions_ToBeAvg* > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) **AND**
(*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*):

{
ProblemType = Medium
ProblemCause = *ProblemCause*+1
HelpLevel = 1
}

END CASE

END.

NextProblem_A (Next Problem for Average Player)

START

// One who did well in the current series

// And did well in the previous series. Assumption: Good at theory and practice
// Different cause, much less help, harder problem

CASE

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) **AND**
(*NextSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) :

{
ProblemCause = *ProblemCause*+1
ProblemType = Hard
HelpLevel = 3
}

// And did ok in the previous series. Assumption: Needed more help or didn't
//know the area. Different cause, less help, different type problem

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) **AND**
(*Min_CorrectlySolved_Questions_ToBeGood* > *NextSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg):

{
ProblemCause = *ProblemCause*+1
ProblemType = Medium
HelpLevel = 2
}

// And did below average in the previous series. Assumption: Needed more
//help and didn't know the area. Different cause, less help, same type problem

(*PreviousSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) **AND**
(*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >=
Min_CorrectlySolved_Questions_ToBePoor):

{
ProblemType = Medium
HelpLevel = 2
}

// One who did ok in the current series

// And did well in the previous series. Assumption: and didn't know the area
// Different, same type problem

(*Min_CorrectlySolved_Questions_ToBeGood* > *PreviousSeries_Avg* >=
Min_CorrectlySolved_Questions_ToBeAvg) **AND** (*NextSeries_Good* >=
Min_CorrectlySolved_Questions_ToBeGood) :

{
ProblemType = Hard
HelpLevel = 2
ProblemCause = *ProblemCause*+1
}

// And did ok in the previous series

(*Min_CorrectlySolved_Questions_ToBeGood* > *PreviousSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*) **AND** (*Min_CorrectlySolved_Questions_ToBeGood* > *NextSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*) :

{
ProblemType = Medium
ProblemCause = *ProblemCause*+1
HelpLevel = 2
}

// And did below average in the previous series

(*Min_CorrectlySolved_Questions_ToBeGood* > *PreviousSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*) **AND** (*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*):

{
ProblemType = Medium
HelpLevel = 1
}

// One who did poor in the current series

// And did well in the previous series

(Min_CorrectlySolved_Questions_ToBeAvg > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) **AND** (*NextSeries_Good* >= *Min_CorrectlySolved_Questions_ToBeGood*) :

{
ProblemType = Medium
ProblemCause = *ProblemCause*+1
HelpLevel = 2
}

// And did ok in the previous series

(*Min_CorrectlySolved_Questions_ToBeAvg* > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) **AND** (*Min_CorrectlySolved_Questions_ToBeGood* > *NextSeries_Avg* >= *Min_CorrectlySolved_Questions_ToBeAvg*):

{
ProblemType = *ProblemType*+1
HelpLevel = 1
}

// And did below average in the previous series

(*Min_CorrectlySolved_Questions_ToBeAvg* > *PreviousSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*) **AND** (*Min_CorrectlySolved_Questions_ToBeAvg* > *NextSeries_Poor* >= *Min_CorrectlySolved_Questions_ToBePoor*):
{
ProblemType = Easy
ProblemCause = *ProblemCause*+1
HelpLevel = 1
}

END CASE

END.

Error Check

START

// Avoiding other problems

// Running out of problems

IF (*ProblemType* = 3) **AND** (*HelpLevel* = 3) **AND** (*No_of_Questions* > *Questions_in_Series*)
{
Exit
}

IF (*ProblemCause* > 8) **AND** (*No_of_Questions* > *Questions_in_Series*)
{
Start from *ProblemCause* = 1 unattempted problems
}

```
IF (ProblemType>= 3) AND (HelpLevel = 3) AND
(No_of_Questions > Questions_in_Series)
        {
                Display Final Message
                Exit
        }

// upgrade player

IF (PreviousSeries_Good = 1) AND (NextSeries_Good  = 1) AND
(PlayerType =  Poor )
        {
        PlayerType =  Average
        }

IF (PreviousSeries_Good  = 1) AND (NextSeries_Good  = 1) AND
(PlayerType =  Average)
        {
        PlayerType =  Good
        }

// Downgrade player

IF (PreviousSeries_Poor  = 1) AND (NextSeries_Poor  = 1) AND
(PlayerType =  Average)
        {
        PlayerType =  Poor
        }

IF (PreviousSeries_Good  = 1) AND (NextSeries_Good  = 1) AND
(PlayerType =  Good)
        {
        PlayerType =  Average
        }

END.
```

Present Problem

START

Display Initial Scenario
ErrorCount
Let interrogate field
ErrorCount

Get *InitialDiagnosis*
IF Correct
{
Get *InitialRemedy*
}
IF Correct // Both Diagnosis and Remedy are correct
{
Get final diagnosis
Show solution
Check problem-list
}

ELSE // Diagnosis correct and Remedy wrong
{
Display Warning Message
REPEAT
ErrorCount
Get *InitialRemedy*
UNTIL (Correct) **OR** (Try > *MaxTries*)

ELSE // Diagnosis wrong

Display Warning Message
REPEAT
ErrorCount
Get *InitialDiagnosis*
UNTIL (Correct) **OR** (Try > *MaxTries*)

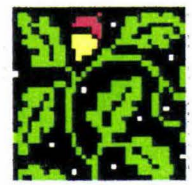
REPEAT
Get *InitialRemedy*
ErrorCount
UNTIL (Correct) **OR** (Try > *MaxTries*)

```
Allow to perform lab procedures
ErrorCount

Get final diagnosis

Compare optimal path list with Path list
IF Not same
        {
        Create Debrief file
        REPEAT
                Check Node
                IF NOT same
                        {
                        Select appropriate error message
                        Add to the file
                        }
        UNTIL (No more nodes)
ELSE
        Debrief file  =  Yours was the optimal path

Show solution
Display debrief file
Display Optimal path
Mark 'Done' at the problem-list
Problem  =  Problem+1

END.
```

Appendix G

Sample screens for proposed Diagnosis

Diagnosis Builder



Player Model | Problem Description | Field Observation | Lab Procedures

Initial Diagnosis | Debriefing Session | Reference Book | Menu Structure

Main menu for proposed Diagnosis Builder

Diagnosis Builder

Menu Structure

- Player Model
  - Strategy
    - About Strategy
    - Create Test
    - Modify Test
    - Test on/off
  - Model
    - Score Ranges
    - About Bugs..
- Problem Description
- Field Observation
- Lab Procedures
- Initial Diagnosis
- Debriefing Session
- Reference Book

Menu Hierarchy of proposed Diagnosis Builder

Diagnosis Builder

Optimal Path

Tick the nodes in the optimal path

☑	Look at Leaves
☑	Collect leaves
☑	Cut into leaves
☑	Look at Flowers
☐	Cut into Root
☐	Collect Root
☑	Ask Neighbour
☐	Look at soil
☐	Collect soil
☑	Goto Lab
☑	Nutrient Check
☐	Baiting test
☑	Initial Diagnosis
☑	pH Check
☑	Final Diagnosis

Optimal path construction in the proposed Diagnosis Builder