The South African Institute of Computer Scientists
and
Information Technologists

Proceedings

of the

1996 National Research and Development Conference

Industry meets Academia

Interaction Conference Centre, University of Natal,
Durban.
26 & 27 September

Edited by
Vevek Ram

ISBN 0-620-20568-7

Cover printed by Natal Printers (Pty) Ltd, Pietermaritzburg
Copying by the Multicopy Centre, University of Natal, Pietermaritzburg
Binding by Library Technical Services, University of Natal, Pietermaritzburg

The views expressed in this book are those of the individual authors

FOREWORD

This book is a collection of papers presented at the National Research and Development Conference of the Institute of Computer Scientists and Information Technologists, held on 26 & 27 September, at the Interaction Conference Centre, University of Natal, Durban. The Conference was organised by the Department of Computer Science and Information Systems of The University of Natal, Pietermaritzburg.

The papers contained herein range from serious technical research to work-in-progress reports of current research to industry and commercial practice and experience. It has been a difficult task maintaining an adequate and representative spread of interests and a high standard of scholarship at the same time. Nevertheless, the conference boasts a wide range of high quality papers. The program committee decided not only to accept papers that are publishable in their present form, but also papers which reflect this potential in order to encourage young researchers and to involve practitioners from commerce and industry.

The organisers would like to thank IBM South Africa for their generous sponsorship and all the members of the organising and program committees, and the referees for making the conference a success. The organisers are indebted to the Computer Society of South Africa (Natal Chapter) for promoting the conference among its members and also to the staff and management of the Interaction Conference Centre for their contribution to the success of the conference.

On behalf of the Organising Committee
Vevek Ram
Editor and Program Chair
Pietermaritzburg, September 1996

Organising Committee

Conference General Chairs
Mr Rob Dempster and Prof Peter Warren (UNP)

Organising Chair
Dr Don Petkov (UNP)

Secretariat
Mrs Jenny Wilson

Program Chair
Prof Vevek Ram (UNP)

Program Committee

Prof Peter Wentworth, Rhodes
Dr Milan Hajek, UDW
Prof Derek Smith, UCT
Prof Anthony Krzesinski, Stellenbosch
Dr Don Petkov, UNP
Mr Rob Dempster, UNP
Prof Peter Warren, UNP

Table of Contents

Real-Time Systems

Database Systems

Education

Practice and Experience

Information Technology and Organizational Issues

Abstracts

List of Contributors

Vladimir B Bajic
Centre for Engineering Research,
Technikon Natal,
P O Box 953
Durban 4000

C N Blewett
Department of Accounting
University of Natal
King George V Avenue
Durban 4001

Justin Cansfield
Department of Accounting
University of Natal
King George V Avenue
Durban 4001

Tom Considine
Apron Services (Pty) Ltd
P O Johannesburg
International Airport
1600

Eric Cloete
School of Electrical Engineering
Cape Technikon
Box 652
Cape Town

I Cloete
Computer Science Department
University of Stellenbosch
Stellenbosch
7600

O A Daini
Department of Computer Science
University of Botswana
Gaborone
Botswana

Nirvani Devcharan
Umgeni Water
Box 9
Pietermaritzburg
3200

P J A de Villiers
Department of Computer Science
University of Stellenbosch
Stellenbosch
7700

Ruth de Villiers
Department of Computer Science and
Information Systems
UNISA
Box 392, Pretoria, 0001

G J Erwin
Business Information Systems
University of Durban-Westville
Private Bag X54001
Durban 4000

G Ganchev
Computer Science Department
University of Botswana
PBag 0022
Gaberone, Botswana

J Geldenhuys
Department of Computer Science
University of Stellenbosch
Stellenbosch
7700

Louise Gibson
BIS, Dept Accounting & Finance
University of Durban
Pvt Bag X10
Dalbridge 4014

Mike Hart
Department of Information Systems
University of Cape Town
Rondebosch
7700

M. Hajek
Department of Computer Science
University of Durban-Westville
Pvt Bag X54001
Durban 4000

A C Hansen
Dept of Agricultural Engineering
University of Natal
Private Bag X01
Scottsville 3209

J M Hattingh
Department of Computer Science
Potchefstroom University for CHE
Potchefstroom 2520

Boris Jankovic
Centre for Engineering Research
Technikon Natal
P O Box 953,
Durban 4000

Meredith McLeod
Centre for Engineering Research,
Technikon Natal,
P O Box 953
Durban 4000

Paula Kotze
Department of Computer Science and
Information Systems
UNISA
Box 392
Pretoria, 0001

D Moodley
Computer Management Systems
Box 451
Umhlanga Rocks
4320

J W Kruger
Vista University
Soweto Campus
Box 359
Westhoven 2124

Andrew Morris
P O Box 34200
Rhodes Gift
7707

A C Leonard
Dept of Informatics
University of Pretoria
Pretoria
2000

R C Nienaber
Technikon Pretoria
Dept of Information Technology
Private Bag X680
Pretoria 0001

Ben Laauwen
Laauwen and Associates
P O Box 13773
Sinoville
0129

E Parkinson
Department of Computer Science
University of Port Elizabeth
Box 1600
Port Elizabeth 6000

Mari Le Roux
Information technology, development: project
leader
Telkom IT 1015
Box 2753
Pretoria 0001

Don Petkov
Department of Computer Science and
Information Systems
University of Natal
PBag x01
Scottsville 3209

P W L Lyne
Dept of Agricultural Engineering
University of Natal
Private Bag X01
Scottsville 3209

Olga Petkov
Technikon Natal
Box 11078
Dorpspruit 3206
Pietermaritzburg

Rose Mazhindu-Shumba
Computer Science Department
University of Zimbabwe
Box MP167
Harare, Zimbabwe

N Pillay
Technikon Natal
Box 11078
Dorpspruit 3206
Pietermaritzburg

V Ram
Department of Computer Science and
Information Systems
University of Natal
PBag x01
Scottsville 3209

Melinda Redelinghuys
Department of Computer Science and
Information Systems
UNISA
Box 392
Pretoria, 0001

Karen Renaud
Computer Science and Information Systems
UNISA
Box 392
Pretoria, 0001

H N Roux
Department of Computer Science
University of Stellenbosch
Stellenbosch
7700

T G Scott
Department of Computer Science
Potchefstroom University for CHE
Potchefstroom
2520

T Seipone
Department of Computer Science
University of Botswana
Gaborone
Botswana

Derek Smith
Department of Information Systems
University of Cape Town
Rondebosch
7700

Anette L Steenkamp
Department of Computer Science and
Information Systems
UNISA
Box 392
Pretoria, 0001

T Steyn
Department of Computer Science
Potchefstroom University for CHE
Potchefstroom 2520

H. Suleman
Department of Computer Science
University of Durban-Westville
Pvt Bag X54001
Durban 4000

A Vahed
Department of Computer Science
University of Western Cape
Private Bag X17
Bellville 7530

A Van der Merwe
Computer science and Informations Systems
UNISA
P O Box 392
Pretoria,0001

Tjaart J Van Der Walt
Foundation for Research and Development
Box 2600
Pretoria, 0001

K Vavatzandis
Department of Information Systems
University of Cape Town
Rondebosch
7700

Y Velinov
Dept Computer Science
University of Natal
Private Bag X01
Scottsville 3209

H Venter
Department of Computer Science
University of Port Elizabeth
Box 1600
Port Elizabeth 6000

H L Viktor
Computer Science Department
University of Stellenbosch
Stellenbosch
7600

R Von Solms
Department of Information Technology
Port Elizabeth Technikon
Private Bag X6011
Port Elizabeth 6000

A J Walker
Software Engineering Applications
Laboratory
Electrical Engineering
University of Witwatersrand
Johannesburg

P Warren
Computer Science Department
University of Natal
P/Bag X01
Scottsville 3209

Max Watzenboeck
University of Botswana
Private Bag 0022
Gaberone
Botswana

COMPUTER-BASED APPLICATIONS FOR ENGINEERING EDUCATION

A C Hansen and P W L Lyne
Department of Agricultural Engineering
University of Natal
Pietermaritzburg

Abstract

There is a rapidly growing awareness in the world of the need to modify the educational process at all levels of human development. In South Africa the situation is made worse by a substantial backlog in education. Technology in the form of computers presents itself as a valuable teaching resource. In the USA conclusive results indicate that the use of computers in training accelerates learning by up to 40%. However, a study of the pedagogical aspects of computer based education shows that there is a great complexity involved in learning and it is impossible to expect that a computer can fulfil all the requirements in an effective learning environment. The inclusion of images, video or sound in a computer-based application can enhance the learning experience significantly as well as increase self-motivation. In the USA, increasing use of multimedia is being made in the field of engineering in both tertiary education and extension. This paper presents some methods for classifying instructional materials. Examples of applications in the different classifications are used to illustrate some dynamic interactive tools that are available for educating engineers.

Introduction

From the time that computers came into being, people have predicted that computer-based education would revolutionise education and training with respect to both effectiveness and efficiency. In spite of these expectations very few of the systems developed over the past three decades have lived up to the promises made for them (Reeves, 1994). However, a study of the literature in the engineering education field indicates that the situation is changing rapidly with much effort being expended in the development of properly designed computer-based instructional materials. Coupled with this development is a greater awareness of the need to incorporate sound pedagogical objectives into the system and to perform appropriate and substantial evaluations in an iterative manner during the development process. Engineers, who in the past were perhaps ignorant of the learning strategies that should be followed, are now seeking the assistance of education specialists in providing instructional materials in the engineering field.

Some of the advantages in making use of computer-based instruction as compared to traditional classroom tuition include interactivity, learner control, consistency, motivation, reduced administration and greater cost-effectiveness (Sheldon *et al.*, 1993). In addition, conclusive results in the USA indicate that the use of computers in training accelerates learning by up to 40% (Andrews, 1994). However, a major disadvantage of computer-based instruction is that it can erode interpersonal relationships between teachers or facilitators and learners. The computer remains an inanimate tool. The technical difficulties inherent in building student models and simulating human-like communications have been greatly underestimated by proponents of intelligent tutoring systems (Reeves, 1994). In relation to the teacher or trainer, computers are more efficient and reliable sources of content while human instructors are better providers of individual guidance, encouragement, and detailed explanation (Reeves, 1994). Hence the teacher's role in the development and implementation of computer-based instruction cannot be neglected and has to be evaluated very carefully.

The objectives of this paper are firstly to put forward a set of educational objectives that should underpin any development strategy for computer-based instructional materials, and secondly to provide examples of applications that fall into different categories that address different objectives and learning styles.

Educational Objectives

The development of effective computer-based instructional materials can be achieved only if the learning objectives of such materials are clearly understood at the beginning of the process. Montgomery and Fogler (1996) state that a well designed package can supplement the presentation of basic course concepts, test their understanding, provide feedback for their efforts, and generally increase both the quantity and the depth of their learning. However, a poorly designed program can lead to massive expenditures of time by both teachers and students without enhancing learning at all.

A system suggested by Montgomery and Fogler (1996) for classifying software in terms of potential effectiveness in a given classroom is based on answers to the following three questions:

a) What thinking skills is the software designed to challenge?
b) What student learning styles does the software accommodate?
c) What are the intended roles of the software?

The well-known and widely used Bloom's taxonomy of educational objectives is composed of the following six skill levels (Montgomery and Fogler, 1996):

a) *Knowledge:* the rote repetition of memorised information. Can the problem be solved simply by defining terms and by recalling specific facts, trends, criteria, sequences, or procedures? This is the lowest intellectual skill level.
b) *Comprehension:* repetition with understanding. Given a familiar piece of information, can the problem be solved by recalling appropriate information and using it? This is the first level of understanding.
c) *Application:* recognition of which set of principles, ideas, rules, equations, or methods should be applied, given all the pertinent data. Once the principle is identified, the problem is solved using skill levels 1 and 2.
d) *Analysis:* process of breaking the problem into parts so that the hierarchy of sub-problems or ideas and their relationships are made clear. Missing, redundant and contradictory information are identified and after the analysis each sub-problem can be solved using skills 1,2 and 3.
e) *Synthesis:* creation of a new whole from a number of parts such as in a design project. Once the various parts are synthesised, each part or problem requires the use of skill level 4 to continue towards a complete solution. This level relies on creative thinking.
f) *Evaluation:* qualitative and quantitative judgements are made about the extent to which the materials or methods satisfy internal and external criteria. This level emphasises critical thinking.

Undergraduate courses typically focus on the first three levels only. Properly formulated computer-based applications represent one way of allowing students to exercise higher level thinking skills.

After the desired skills have been identified, it is necessary to take into account the learning styles of the students in order to reach them more effectively and ensure that these skills are exercised (Montgomery and Fogler, 1996). The issue of learning styles has become a major consideration in the teaching/learning process. Rosati and Felder (1995) suggest that a student's learning style is the collective way in which a student preferentially takes in and processes information.

A learning style classification system proposed by Felder and Silverman (1988), cited by Rosati and Felder (1995) has been applied by a number of researchers and was identified by Montgomery and Fogler (1996) as being the most useful model in helping students understand their learning needs and preferences. This system comprises five dimensions as shown in Table 1. Results of assessment of learning styles of engineering students by Montgomery (1995) and Rosati and Felder (1995) indicated that approximately 67% of students learn best actively, 57% are sensors, 69% are visual and 28% are global.

The combination of Bloom's taxonomy of educational objectives and the learning style classification system provide a means of determining both the skills being exercised by the software and the mode of interaction with the student. These two considerations can be used in the assessment of the role of computer-based applications in engineering education.

Table 1 Dimensions of the inventory of learning styles (Montgomery and Fogler, 1996)

Dimension	Range	Comments
Perception	Sensing/Intuitive	Sensors prefer data and facts, intuitors prefer theories and interpretations of factual information.
Input	Visual/Verbal	Visual learners prefer charts, diagrams and pictures, while verbal learners prefer the spoken or written word.
Organisation	Inductive/Deductive	Inductive learners prefer to infer principles from facts and observations, while deductive learners deduce consequences from underlying principles.
Processing	Active/Reflective	Active learners learn best by doing something physical with the information, while reflective learners do the processing in their heads.
Understanding	Sequential/Global	Sequential learners make linear connections between individual steps easily, while global learners must get the "big picture" before the individual pieces fall into place.

Classification of Software

A vast range of software, available at present, is able to contribute to the education of engineers in various ways. Montgomery and Fogler (1996) provide a classification scheme that can be used to categorise this software by its role in addressing specific educational objectives and learning styles. The following four categories are presented by Montgomery and Fogler (1996):

a) *Presentation:* emphasis is on the knowledge, comprehension and application levels of Bloom's taxonomy. The software focuses on the delivery of technical material. Montgomery and Fogler (1996) list the ways in which these materials can be presented with their corresponding learning style dimensions:
•Display of text material (verbal)
•Access to expanded explanation of text material through hot keys (active, sequential)
•Visual and graphical representation of material (visual, sensing, global)
•Use of animation to display phenomena (global), or manipulate equations (active)
•Display of video clips to illustrate industrial situations (global, visual, sensing).

b) *Assessment:* student is tested on mastering of material, such as through the use of multiple choice questions, which are often closed-ended and tend to focus on the first four levels of Bloom's taxonomy, although the upper two levels of synthesis and evaluation can be reached. Certain types of assessment can cater for active (through interaction), sequential (orderly) and sensing (if it deals with real situations) learners.

c) *Exploration:* allows users to better understand the role of various parameters on the performance of a given process through exploration of the process. These are exploratory simulations within a confined parameter space which provide students with a variety of problem definition alternatives and solution pathways to follow. Active learners appreciate the chance to manipulate parameters, visual learners benefit from graphical representations of phenomena, deductive learners can practice drawing their own conclusions, and sensors and global learners get to experience a real process, or at least a simulation of it. This type of software focuses on application and analysis in Bloom's taxonomy.

d) *Analysis:* includes software packages that allow students to enter equations and parameter values for any system. These packages include spreadsheets and equation solvers which allow users to create and solve new models and the corresponding sets of equations very easily. Software in this category gives students, particularly active, deductive and visual students, practice of the higher levels of Bloom's taxonomy, synthesis and evaluation.

Development Process

In the development of a Farm Tractor and Machinery Certification CAI (Computer Aided Instruction) program, Sheldon *et al.* (1993) identified six phases:

a) Discovery: this phase involved a task analysis, collection of resources, and generation of ideas.
b) Design: in the design phase, performance objectives were developed from the steps of the discovery phase. Prerequisite skills were identified and an outline of the program structure was developed.
c) Development: the development phase of the model involved specifying learning activities and sketching lesson displays on paper. Then the entire program was outlined on paper.
d) Coding: the actual computer programming took place in this phase.
e) Documentation: both internal documentation which explains the function of coding and external documentation which includes user guides and exercises are important components of this phase of the model.
f) Delivery: this stage includes pilot testing, validation, and field testing of the materials.

Grayson (1996) emphasised that the development of effective computer-based instructional materials should be iterative and interactive with the students. The following steps were put forward by Grayson (1996) as desirable components in a model for instructional software development:

a) Select topic (content area) and target group;
b) Identify student conceptual difficulties;
c) Write preliminary version of program;
d) Try it out with students - observe what they do, and how they interact with the program, test what they learn, and solicit students' comments and suggestions;
e) Modify program on the basis of step 4;
f) Try it out with students;
g) Repeat step 5 and 6 until the program is reasonably robust and fairly stable.

Examples of Applications

Some examples in each of the software categories are presented in this section. These examples are drawn mainly from the range of software applied in the Department of Agricultural Engineering, University of Natal.

Presentation

Many examples can be found in this category. One such interactive program called ChainSaw was developed by Lown *et al.*(1993) from the University of Tennessee in the USA to teach chain saw and tree-cutting safety . Seventy screens in ten sections containing chain saw and tree-cutting facts, figures, procedures, photos, simple animations, and sounds are arranged in a notebook format. Figure 1 illustrates the opening screen or first page of the program. The information can be accessed sequentially page by page or particular topics can be referred to by selecting the appropriate label in the index on the right hand side of the page in Figure 1. Four of the five methods of delivering technical material mentioned earlier are employed in this software.

A major avenue being explored by software developers in the presentation category is the World Wide Web, which incorporates support for multimedia presentation of material. Relatively little interaction has been possible until more recently with the development of Java applets and ActiveX controls. The opportunities and potential for delivery of instructional materials via "the Web" appear to be enormous and hence it can be expected that there will be tremendous expansion in this area subject to connection speeds.

Figure 1 First screen of interactive ChainSaw program

Assessment

Packages in the assessment category allow students to test their knowledge of the material. The ChainSaw program described above includes a quiz in which students are required to select a correct answer to a question from a list. The program keeps track of the number of questions answered correctly.

An example presented by Montgomery and Fogler (1996) involves a program that helps students prepare for an undergraduate laboratory experiment on pumps. Some short questions as illustrated in Figure 2 give students a chance to practise their laboratory skills. Sensing and active learners benefit from this type of interaction, as do reflective learners, in that it allows them to gain confidence in tackling the actual equipment.

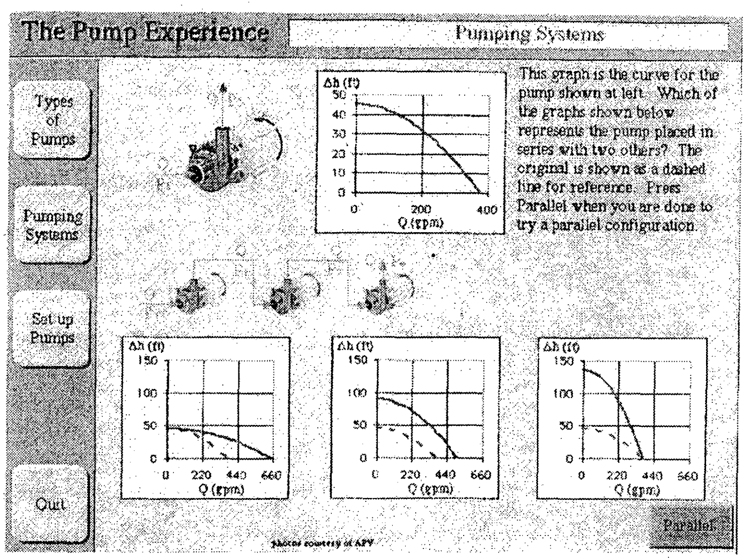

Figure 2 Interaction in pumps for assessment of understanding of laboratory equipment (Montgomery and Fogler, 1996)

Exploration

Computers offer great opportunities for developing simulations. One example of such a simulation was reported by Hansen *et al.*(1995), in which a simple PC-based diesel engine simulator was developed with the objective of providing students with an exploratory tool for learning how a diesel engine responds to adjustments in governor control lever position and to changes in engine load. The simulation was programmed for the Windows environment using a multimedia authoring package. The engine functioned in accordance with simple textbook principles. The first part of the simulator required the input of engine performance parameters that were used to define the bounds within which the engine was expected to operate. For the second part, a full screen engine panel illustrated in Figure 1 was designed to incorporate various elements including push buttons, slider buttons, digital meters and a graph panel to allow the student to interact directly with the engine and to observe its responses on a dynamic and instantaneous basis.

This type of interaction is ideal for visual, active, sensing and global learners, and if used correctly, can exercise their application, analysis, synthesis and evaluation skills.

Analysis

In this category, one of the most commonly used software packages is spreadsheets, which allow the user to assemble relatively complex models that rely on suitable input data, equations and macros where necessary, to generate results and graphs. Another group of software packages that fall into the category of analysis are the equation solvers such as Mathematica, MATLAB and Mathcad. Students are able to set up a system of equations with inputs in an appropriate sequence and obtain a solution in numerical and graphical form. The interactive nature of these programs allows the student to use a significant amount of time exploring complex problems by varying system parameters and operating conditions and observing the results almost immediately rather than spending tedious time programming the system of equations used to model the physical system as well as the numerical techniques needed to solve these equations. With the emphasis on analysis rather than programming, the student not only learns through discovery from the results of varying parameters, but he/she has the opportunity to practise his/her creativity and synthesis skills.

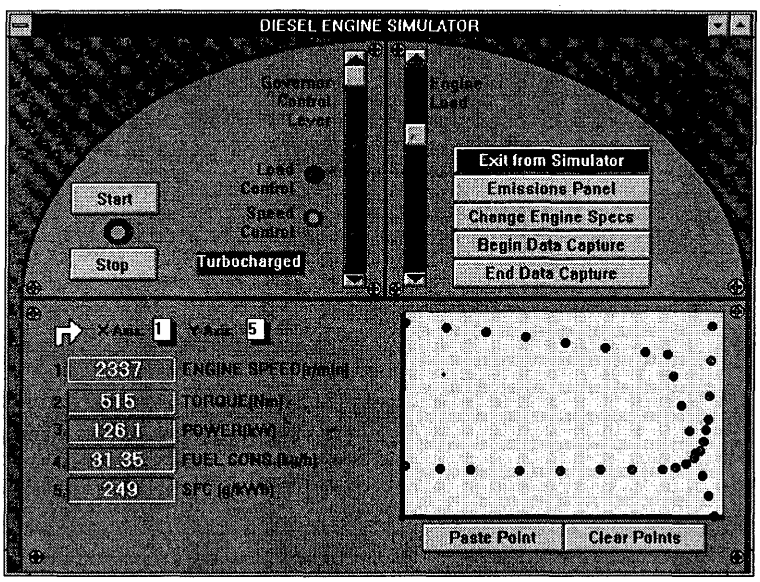

Figure 3 Engine control panel of simulator

This process has been taken a step further by combining textbooks with equation solvers, in which solutions to specific problems in the text are provided. An example is the Schaum's Interactive Outline series published by McGraw Hill and MathSoft Inc., which incorporate a Mathcad Electronic Book. Such applications address all six levels of Bloom's taxonomy and help virtually all learner types in Table 1.

Another example involving the development of qualitative problem solving skills is a set of computer modules developed to supplement a problem solving textbook by Fogler and LeBlanc (1994). An example of a screen from the problem analysis module is shown in Figure 4.

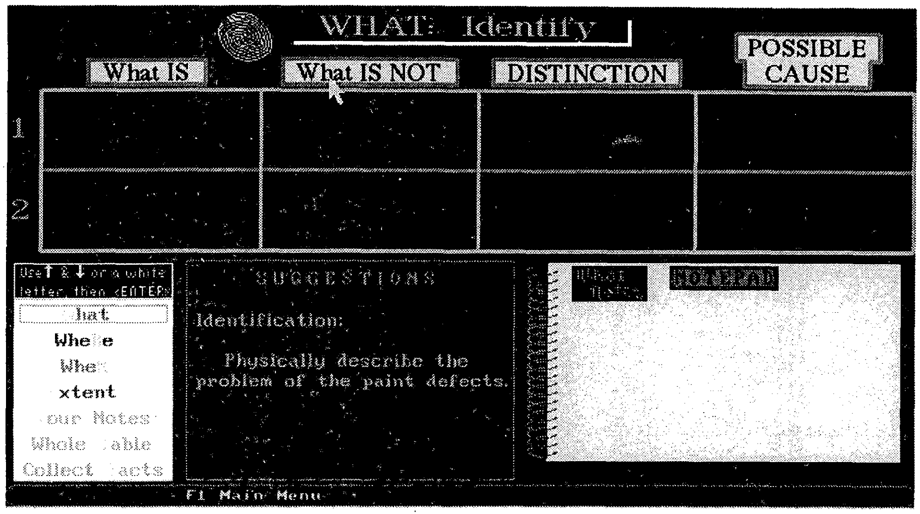

Figure 4 Problem analysis sample screen (Fogler and LeBlanc, 1994)

Conclusions

Software that has been developed more recently for instruction of engineering students addresses many of the educational objectives put forward in Bloom's taxonomy. In future development of computer-based instructional materials, it is vital that appropriate pedagogical objectives and evaluation methods be identified right from the start to ensure that the software is effective.

References

Andrews, SJ, 1994. Some cultural and perceptual implications of courseware development and the use of technology within a multicultural, multilingual society (a cautionary tale). Proc. International Conference on Computer-Assisted Education and Training in Developing Countries, Eskom Conference and Exhibition Centre, Halfway House, Johannesburg.

Felder, RM and Silverman, LK, 1988. Learning and teaching styles in engineering education. Engineering Education, 78(7), 674-681.

Fogler, HS and LeBlanc, SE, 1994. Strategies for creative problem solving. Publ. Prentice-Hall PTR, New Jersey, USA.

Grayson, DJ, 1996. Using education research to develop waves courseware. Computers in Physics, 10(1): 30-37

Hansen, AC, Reid, JF and Goering, CE, 1995. Diesel engine simulator as a multimedia educational tool. Proc. Annual Conference of American Society for Engineering Education, Anaheim, California, USA. Vol 1: 140-144.

Lown, JB, Prather, TG and Peters, PA, 1993. Multimedia training for loggers, arborists and woodcutters. ASAE paper 937523, Winter Meeting. American Society of Agricultural Engineers, Chicago, Illinois, USA.

Montgomery, S, 1995. Addressing diverse learning styles in a large classroom. Proc. Annual Conference of American Society for Engineering Education, Anaheim, California, USA. Vol 1: 344-349.

Montgomery, S and Fogler, HS, 1996. Selecting computer-aided instructional software. Journal of Engineering Education, 85(1): 53-60.

Reeves, TC, 1994. A model of the effective dimensions of interactive learning. Proc. International Conference on Computer-Assisted Education and Training in Developing Countries, Eskom Conference and Exhibition Centre, Halfway House, Johannesburg.

Rosati, PA and Felder, RM, 1995. Engineering student responses to an index of learning styles. Proc. Annual Conference of American Society for Engineering Education, Anaheim, California, USA. Vol 1: 739-743.

Sheldon, EJ, Field, WE and Tormoehlen, RL, 1993. CAI/Multimedia approach to farm tractor and machinery safety certification. ASAE paper 933540, Winter Meeting. American Society of Agricultural Engineers, Chicago, Illinois, USA.