

Towards a Framework for Developing Educational Software

Farid Meziane⁽¹⁾, Chan Weng Onn⁽²⁾, Jane Labadin⁽³⁾ and Roger Harris⁽³⁾

⁽¹⁾Department of
Computer and
Mathematical Sciences,
University of Salford,
Salford M5 4WT,
U.K.
F.Meziane@salford.ac.uk

⁽²⁾MIMOS Berhad
Malaysia,
4-3 Jalan SR 7/4
Serdang Raya
43300 Seri Kembangan
Selangor, Malaysia
cwonn@mimos.my

⁽³⁾Faculty of
Information
Technology.
UNIMAS,
94300 Kota Samarahan,
Sarawak, Malaysia.
jane@fit.unimas.my
roger@fit.unimas.my

Abstract

There is a need for review in mathematical education to satisfy a growing demand for graduates that can support research and development in many technical fields. Dissatisfaction with current curriculum has pushed many institutions and countries to review their programs and their teaching methods. Due to the availability of technology, many are looking at its use to improve mathematics teaching. At the same time there is few support from the existing software methodologies for educational software. In this paper we describe a framework for developing educational software. The first prototype of the framework is presented together with Disc-Maths, a web-based software which is used to validate the proposed framework. Disc-Maths aims at introducing the learning of discrete mathematics at an early age.

1. Introduction and Problem Statement

1.1 Introduction

For very long software developments were related to business oriented applications. But all software profit-making and curiosity-seeking potential, have largely ignored educational software as a potential market. It is because the investment in the education field is limited. Therefore the software development is seldom related to education although software development has grown rapidly in recent years.

However, the appearance of the Internet has changed the attitude of the society. From the structure point of view, Internet is creating a professional bond between teachers and learners as never seen before in the history of education. The Internet especially the World Wide Web, with its increasing capacity for multimedia, communication and information presentation, easy access to an ever growing body of information, and new ways of data representation, presents educators with exciting opportunities to enhance teaching and learning (Lange, 1993). Due to this awareness of computer technologies as a powerful educational tool, many people and countries attempt to transform technological advances into solutions to their school problems. In Malaysia for example, there is a Smart School project, which is part of the Multimedia Super Corridor project. In smart school various information technologies are applied in teaching, learning, staff training and management. Through the Multimedia Super

Corridor networks infrastructures, the smart schools can share the teaching-learning materials. In short, smart schools aim to deliver education in a better way. (Smart School, 1997)

Educational applications will play a major role in the reforming of the education system from a conventional static style to a dynamic online system. In this paper, our main focus is the use of technology in mathematics education. We present our approach to develop Disc-Maths an aid tool to introduce some concepts of discrete mathematics for children. In this paper, we will use the term student to mean children of the group K-12.

1.2 Why Mathematics?

A call for changes in mathematical curriculum and teaching methods started in the early 80s. In the United States these calls mainly arose as a result of [Fusion, 1992]:

- (a) evidence of shortcomings of the present school mathematics efforts as evaluated by the National Assessment of Educational Progress
- (b) evidence that children use a variety of different conceptual structures that result in different solution procedures
- (c) an increasing awareness that we are in the process of changing from an industrial to an informational society.

There is a need to produce mathematically literate students, who can function in a society driven by technology [Burril, 1993]. Mathematics education needs to undergo a shift from teaching student to be calculators for known problem situations to helping them learn to use calculators, and other future tools, to solve problem situations not yet imagined by adults [Fusion, 1992]. Due to the increased access to technology, some studies have started looking at the technology's impact on mathematics education [Schmidt, 1997]. Now computers are becoming essential tools to daily needs, thus giving the opportunity for educators to enhance teaching and learning by using educational software. Kaput [Kaput, 1992] distinguished two types of educational software, generic tools and specific tools. Generic tools are those originally developed outside education such as calculators, word processing and spreadsheets and later adapted to particular educational purposes. However, the retrofitting of these tools into educational environments is not an easy task and will not be further discussed in this paper. We are more interested in the specific tools developed for educational purposes. These tools seem to be more sophisticated and fulfil user needs. The appearance of the Internet and the ever-increasing demand for distance education has made educational software as a potential market for the software development industries. Unfortunately, educational software development has not reached its maturity compared to business oriented software for example.

In this paper, we are presenting a framework for developing educational software. The software developed, Disc-Maths, is meant to initiate children to the concepts of discrete mathematics [John *et al.*, 1993], one of the many components in mathematics, which includes logic, linear algebra and graph theory. We believe that if children are initiated earlier to discrete mathematics, they will be in a better position to understand and succeed in more advanced fields such as the study of computer science. Indeed, discrete mathematics provide the mathematical foundations for many

computer science courses, including data base theory, formal language, compiler theory and modelling and specification of systems [Kroenke, 1995, Booch, 1994, Jones, 1990]. Schoenfeld [Schoenfeld, 1992] stated that mathematically powerful students use mathematics in practical ways, from simple applications such as using proportional reasoning for recipes or scale models, to complex budget projections, statistical analysis, and computer modelling. A study by the National Council of Teachers of Mathematics (NCTM), found out that children engaged in real-life maths problems are more motivated to learn and use maths skills than those who are not [Eiser, 1993].

The remaining sections of the paper will be organised as follows: Section 2 will state the goals of Disc-Maths and describe the methodology adopted for the development of the system. Section 3 will describe the overall organisation of Disc-Maths. A brief walkthrough of Disc-Maths will be discussed in section 4. Finally, section 5 will look at the future improvement of the system.

2. The Development Approach

The development of Disc-Maths is intended to offer a conceptual framework for implementing technological innovations in education. The goals and design principles of Disc-Maths emphasis on creating applications that provide common and transparent technology to learners. To promote adoption, it is important to make technology as transparent as possible. Teachers and students should not spend an excessive amount of time learning to use the technology. The use of common technology reduces costs in many ways. First, it reduces costs of hardware and software. Second, it minimises training and supporting staff. Third, it saves time and frustrations for the teachers to learn to use several technologies. The Web can be considered a common technology because it can be and is used for many other purposes in schools.

2.1 System Objectives

Based on the early review presented in the introduction, the aims of Disc-Math are as follows:

- a) Develop a system to aid the learning of discrete mathematics through real life problems
- b) Develop the software in a systematic manner based on a rigorous development method
- c) Implement the system using a portable and easy to use technology

The first aim is to support the early statement that students learn better through real life problem solving. The success of the software and its acceptance by children will depend on the way the material will be presented. Therefore a rigorous approach to requirements analysis is necessary. A good software development approach will not only support software specification and design but also ensure future maintainability of the system making any changes possible and at a lower cost. A systematic method will also support reuse for other educational applications. Schools are normally using a variety of hardware and software. Therefore, appropriate educational software should be one that can run on different hardware and software platforms. This will allow students to access the same interface regardless of the hardware used.

2.2 Requirements Elicitation for Educational Software

The first phase of the software development process is requirements definition. According to Sommerville [Sommerville, 1996], this document represents the contract between the client and the analyst and must be understood by both the client and the contractor. It has been acknowledged that failure to produce a correct requirements document will result in the production of the wrong system [Vadera and Meziane, 1997]. Many techniques have been developed to support requirements elicitation and the most effective are those involving users during the analysis phase. When developing educational software for children it is difficult to get them involved in the requirements analysis phase. Are we therefore developing the wrong software for them? How can we ensure that the educational software satisfies the user requirements (learning is taking place)? The existing software methodologies do not address these issues. Therefore we believe that there is a need to adapt the existing methods, if not looking for new methods, to support educational software development. Developing software for education is certainly different from developing software for business. Interactive Training Technologies [Interactive Training Technologies] has for example developed their own method, the New Media Development Method (NMDM), to support their educational software development. Another issue in educational software is human computer interaction (HCI). Research into HCI has prospered in recent years. However, this research has been mainly preoccupied by the use of computers by adults: there has been little interest in design issues that arise when users are children [Crook, 1998]. This is another problem that is worth investigating. A framework for developing educational software is described in the next section.

2.3 The Method

We have used a development method adapted from the Booch method [Booch, 1994], an object-oriented method, for the development of Disc-Maths. The Booch method is selected because it is an incremental object-oriented method widely accepted and used by the software community. It will support reuse and the steps are not very strict, they can be adapted to satisfy the goals of a particular application. Figure 1 shows the steps of the methodology adopted for the development of Disc-Maths. The Booch method gives great consideration to the requirements analysis phase. Two documents are produced during this phase: The *system charter*, which outlines the responsibilities of the system and the *system function statement*, which outlines the key use cases of the system. The use cases approach is also used by the Object-Oriented Software Engineering method [Jacobson, 1992]. However we think that these are not enough to cater for educational software. We kept all the steps defined by the Booch methods for the requirements phase to keep all the rigor of the method. Since it is difficult to evaluate user requirements, it is important to develop a first prototype to ensure that we are developing the "correct" software. While developing Disc-Maths, the proposed method is also tested and later its success will be measured with the success of Disc-Maths. So far in this project, only the first three phases are accomplished. These phases will be described in the following subsections.

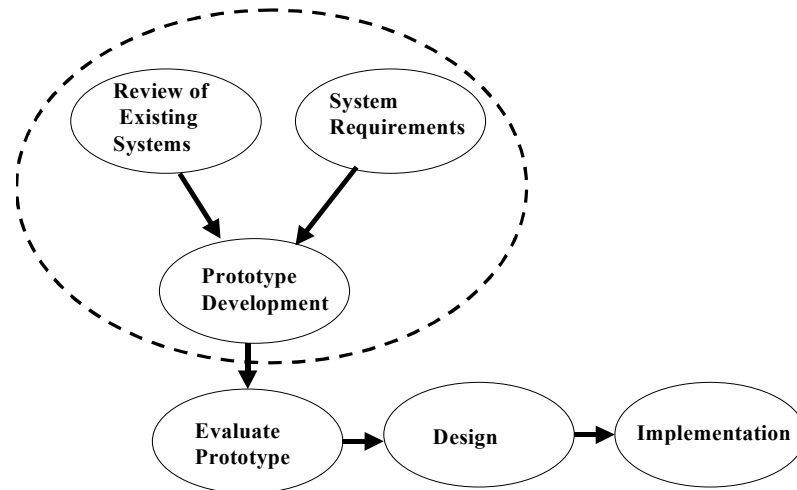


Figure1: The method for the development of Disc-Maths

2.3.1 Review of Existing Systems

There was a need to study some existing software to have an idea on the technology and pedagogy used for the design of some educational software. The following four systems were reviewed:

- Lets go Shopping [Dataworks, 1995]
- MathWise Astronomy[Moore, 1996]
- Virtual Math [Huang, 1997]
- Math Scavenger Hunt [Superkids, 1997]

The choice of these four systems is dictated by their popularity, availability on the Internet and also because their approach to teaching mathematics is close to Disc-Maths. Superkids Educational Software Review has also conducted an evaluation of some mathematics software [Superkids, 1997].

Let's Go Shopping, is developed by the Australian company Dataworks. The software helps kids go shopping and learn maths and money as they go. *Let's Go Shopping* not only lets children pay in Australian currency but also fits into the Australian Maths Curriculum and real life. The skills required for successful shopping require a range of understanding. *Let's go Shopping* can be set to suit a five-year-old or challenge an eleven-year-old. The sound structure, attractive activities, colourful graphics and high interest value, makes *Let's go Shopping* a good program for Primary school students.

The *MathWise* astronomy module can be regarded in two ways. On one hand, it is an excellent introduction to astronomy, which makes good use of basic mathematical techniques. On the other, it uses astronomy as a medium to illustrate the practical application of those same mathematical techniques. Either way, the combination is a winning one. Whether the aim is to use astronomy to teach mathematics, or vice versa, this module is a very good educational tool.

Virtual Maths is an on-line application that serves as a graphic calculator, an equation solver, a plotter of user-defined functions, derivatives and integrals. It teaches the fundamental of graphs and functions. It aims to teach the users powerful images in the shape of the graphs of the functions so they can see how the gradient of these graphs varies (leading to differentiation) and how the area under the graph can be calculated. This is a very useful mathematics education tool.

Maths Scavenger Hunt is an educational game that combines reading, mathematics practical, critical-thinking, and problem solving skills in a virtual board-game setting. The objectives of the game is to collect a set of objects in a scavenger hunt that leads the user through San Francisco, stopping at various landmarks to solve real life maths problems.

After the analysis of these existing software we have identified some criteria that we think were important when evaluating these software. The five criteria involved are:

- a) The technology used
- b) The models used for the presentation of the contents
- c) The layout of the content and the level of control
- d) The navigation approach
- e) The user interaction

As it can be noticed, our criteria are more on the technical aspects of the software design and they are all input related. We could not at this stage evaluate the software regarding output criteria such as the depth of learning.

The system requirements phase involves the definition of the system's objectives and its functional requirements. Functional requirements of Disc-Maths included the content of the course, method of delivery, the evaluation method, system interactivity, technology used etc.

2.3.2 Prototype development

The design of the prototype is based on the requirements of Disc-Maths using the strengths of the analysed existing software. The Booch method is used for the development of the prototype. It is important at this stage to use a structured method because we are not expecting to completely throw away the initial prototype. To the contrary, we intend to increment the earlier prototype to a full working version of the system. The use of such a rigorous development method will support the required changes.

3. The Organisation of the Disc-Maths System

The analysis phase of the project has identified three components for the system: The lecture, the tutorial and the self-testing components which are considered as the main components of Disc-Maths since the students learning process is centred around these three activities and they complement each other. In addition, we have identified two more components, the services and the Graphical User Interface (GUI) components. Figure 2 summarises the architectural design of the Disc-Maths System.

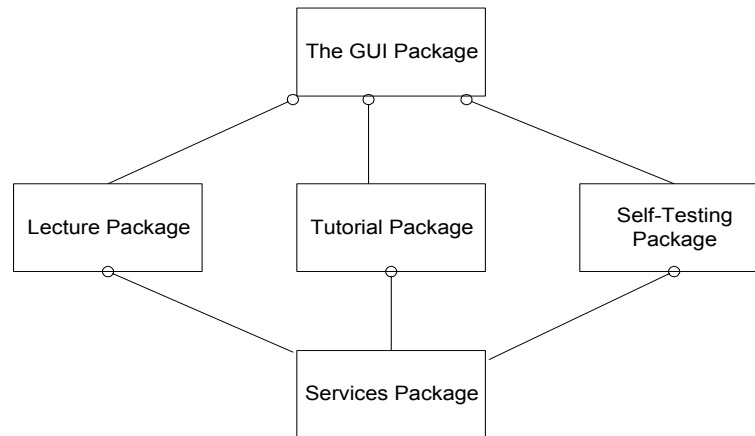


Figure 2: The architectural design of the Disc-Maths system

3.1 The Technology Used

Disc-Maths is designed and implemented as a multimedia web based application. Web-based approach has a number of advantages for the distribution of application. They can be accessed, re-used and annotated without any visible restrictions. Compared with non-network applications, they reduce cost and time in terms of management and maintenance. Multimedia has strong attractions for both teachers and students at all levels across a wide variety of subjects. Multimedia-based interactive learning technology allows students to experience a subject from a diverse series of angles in such a way that their understanding of the subject is multi-dimensional. Apart from providing various perspectives in the teaching of a subject, such systems allow users to explore the subject as deeply as needed while the incorporation and utilisation of various resources keeps the interest level high. Java has been chosen as the implementation language because its ideal for the development of secure, distributed, network-based, interactive applications which run on a variety of computer platforms and supports for multimedia applications [Reed and Abdollah, 1997].

3.2 Model for the Presentation of the Contents

We have used real life scenarios very close to students' daily life to illustrate the different concepts presented for the study. At this stage only the set theory and the relationships concepts are implemented. We have illustrated the set theory by using the scenario of two friends going shopping. Their trolleys representing the containers for the set elements that are represented by the items purchased. The different set operations were illustrated by comparing the items purchased (intersection, inclusion) or by putting them together (union). Definitions and theorems are given after the concepts are first illustrated.

3.3 The Layout of the Content and the Level of Control

The layout of Disc-Maths is composed of three frames. A left frame that contains the three content packages of the software namely, lectures, tutorials and self-testing and a help. The right frame contains the details of the package content and the main frame (centre) contains the content itself. This approach gives the user total control during the learning process. The students will have a general mindset about what the system offers and what they can learn. Figure 3 illustrates the structure of the lecture package and a screen introducing set operations.

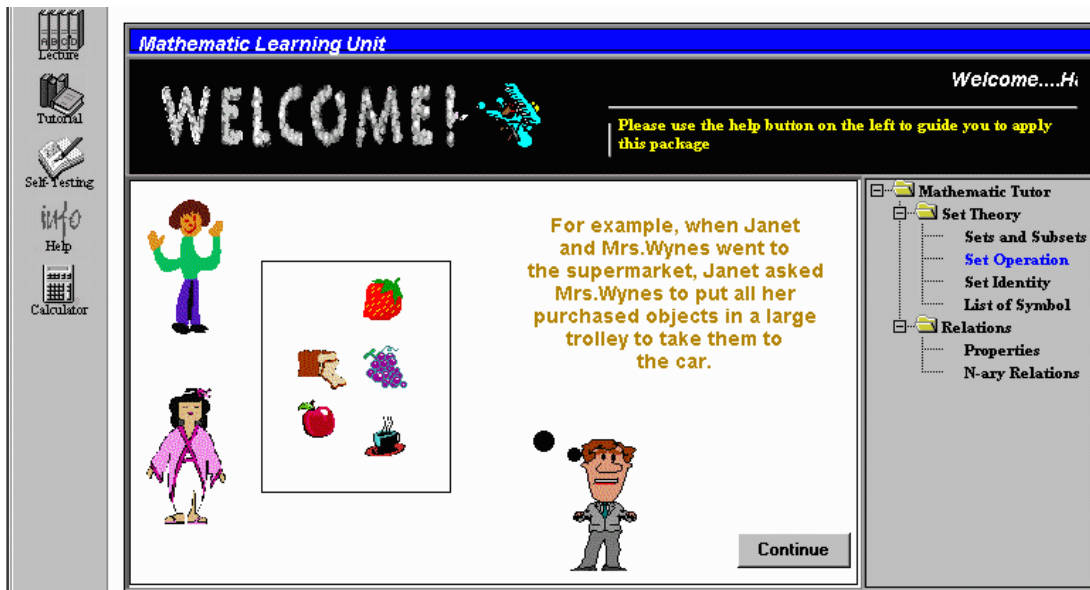


Figure 3: The structure of the lecture package

3.4 The Navigation Approach

Navigation approach here refers to the screen display method of the software. Within a topic, a card layout approach was adopted for Disc-Maths. It consists of a stack of cards, piled one on top of the other. Two buttons, "next" and "previous" will assist the student in going through the course material. This navigation approach is more systematic and controllable regarding the amount of information displayed. Within the course or a particular chapter, students can go directly to any course component or a topic. This is less confusing than the zoom-in and zoom-out approach where the students may lose their way during navigation. The card approach is also more efficient than the multi-frame approach when downloading the application.

3.5 The User Interaction

A graphical user interface is used for Disc-Maths. The user interface supports the navigation through the software components and allows some interactivity with the system. Interaction is a desirable feature for any educational software. There is little interactivity during the lecture process for Disc-Maths, however a lot of interaction is introduced during tutorial activities. Figure 4 for example shows a tutorial session and illustrates the set operation topic. For this first prototype, Disc-Maths does not include many multimedia features such as sounds or videos. These will be included in the final version of the system as there are many considerations to take into account when using interactivity in educational software. A detailed study of visual elements and interactivity for courseware design can be found in [Thibodeau, 1997]

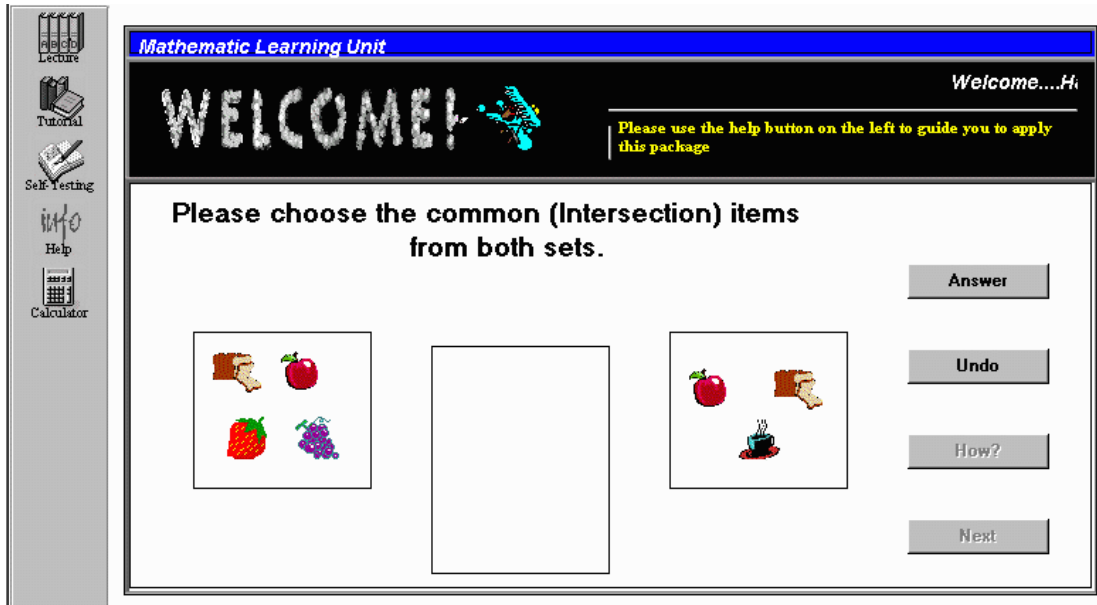


Figure 4: A tutorial session

4. A Session with the Disc-Maths System

Since the five components of the systems (Figure 2) were meant to be independent, the implementation of the system was performed as a set of five modules. First, an integrated user interface that will support all the other modules is implemented. This module will ensure that the system is flexible enough to permit additions and removals of course content. If the students start by selecting lectures, then the contents of the lectures will be displayed. Students have to choose the specific topic they want to go through. A scenario will be first used to explain a concept through real life situation. Formal definitions and theorems will then follow. The organisation of the content allows the students to start the session from any topics of the lecture. After the lecture, the student may want to go through a tutorial associated with the lecture. A tutorial is normally a set of exercises both related to the real life scenario and some other theoretical exercises. In case a student is having problems in understanding or solving a particular problem, a step by step function will demonstrate the solution of the exercise on the screen. The student can repeat the tutorial as many time as he wishes. The student may also take a self-test at any time during his/her session. Self-assessment is regarded as an important part of learning and has been acknowledged in many descriptions and models of instructional design [Gagne *et al*, 1988; Rowntree, 1991; Laurillard, 1993 and Taylor, 1998]. The test is generated from a pool of question and a time is set for answering the questions. The test can be terminated either when the time is over or when the students presses the submit button. After submission an analysis of the students answers is performed and the result is displayed. The students can request some feedback from the system. This is given as answers to the wrongly answered questions.

5. Conclusions and Future Work

The work presented in this paper is the description of the first three phases of the design of Disc-Maths. A framework for the development of educational software has been presented and a first prototype was built. The next step is the evaluation of the prototype. This phase is very important for the final design. It is difficult to validate user requirements at an early stage because the users of the software (the learners) are

normally not involved during the requirements analysis phase for educational software. The evaluation of the prototype is necessary to find out if the software is really meeting the initial needs, support learning, use the correct pedagogy etc. The feedback from the evaluation will initiate a new set of requirements for the final design of the application. Criteria for the evaluation of the software need first to be defined. Several points of view must be considered when evaluating educational software. These include domain dependent demands, instructional demands, user interface demands and pragmatic demands [Kuittinen, 1998]. Previous work on Educational software evaluation should also be taken into account [Squires and Preece, 1996; Scanlon *et al*, 1997]

Since it is an online webware that can be distributed across networks, the delivery time of the webware from the web server to the user's computer will be a critical factor when the size of the webware becomes larger. As a solution, we should focus on two factors in the final design and implementation. First, we should reduce the downloading size of the webware. A solution might be to design a control mechanism that will only load the main functionality of the package and the remaining will be loaded when needed. Second, we should maximise the reusability of the webware's components. For this purpose, we need to develop more general and primitive components that can be shared by different applications and thus reduce the code size. As we have stated earlier, other multimedia components such as audio and video should be included.

Acknowledgements

We would like to thank Professor Justo Diaz for his helpful comments on an early version of this paper.

References

Booch, G. (1994). *Object-oriented Analysis and Design with Application*, Benjamin/Cummings Publishing Company.

Burril G. (1993). Daily-life Applications in the Maths Class, in de Lange, J., Keitel, C., Huntley, I. and Niss, M. (Eds), *Innovation in Maths Education by Modelling and Application*, Ellis Horwood, 165-176.

Crooks, C (1998). Children as Computer Users: The Case of Collaborative Learning, *Computers & Education*, 30(3/4), 237-274

Dataworks, (1995). Let's Go Shopping, <http://www.edutainment.com.au/mshop.html>

Eiser, L. (1993), Maths for a Reason, *Technology & Learning*, 13(58), 52-55.

Fusion, K.C. (1992). Research on Whole Number Addition and Subtraction, in D. A. Grouws D. A., (Ed), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, 243-275.

Gagne, R.M., Briggs, L.S., and Wagner, W.W. (1988). *Principle of Instructional Design* (3rd Edition), Holt, Rinehart and Winston.

Huang W. (1997). Visual Maths for Java 1.03,

URL : <http://acsusun.acsu.unsw.edu.au/~s9300078/VisualMath.html>

Interactive Training Technologies, *The New Media Development Method*
<http://www.itrain.co.uk/newmed1.htm>

Jacobson, I. (1992). *Object-Oriented Software Engineering, A Use Case Driven Approach*, Addison-Wesley.

John, D. A., Albert, O. D., Lawrence, S. E. and Charles, E. V. (1993). *Discrete Mathematics*, New York: HarperCollins College Publishers.

Jones, C.B. (1990) *Systematic Software Development Using VDM*, Prentice Hall International.

Kaput, J. J. (1992). Technology and Mathematics Education, in Grouws, D. A. (Ed), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, 515-556.

Kroenke D.M. (1995). *Database Processing Fundamentals, Design, and Implementation*, Prentice Hall International.

Kuittinen, M.(1998). Criteria for Evaluating CAI Applications, *Computers & Education* 31, 1-16.

Lange D. J. ,Huntley I.,Keitel C.,Niss M (1993). Innovation in Math Education by Modelling and Applications. New York: Ellis Horwood.

Laurillard, D. (1993). *Rethinking University Teaching: A Framework for the Effective use of Educational Technology*, Routledge, New York.

Math Scavenger Hunt (1996).

<http://www.superkids.com/aweb/pages/reviews/math3/scavhunt/merge.shtml>

Moore, P. (1996). Mathwise Astronomy module.

<http://www.birmingham.ac.uk/ctimath/reviews/aug96/astro.htm>

Reed A.J, Abdollah, A.A. (1997). Developing Interactive Educational Engineering Software for the World Wide Web with Java, *Computers & Education*, 30(3/4) 183-194.

Rowntree, D. (1991). *Teaching through Self-instruction: How to Develop Open Learning Materials*, Nichols Publishing New York.

Scanlon, E., Tosunoglu, C., Jones, A., Butcher, P., Ross, S., Greenberg, J., Taylor, J., and Murphy, P. (1997). Learning with Computers: Experiences of Evaluation, *Computers & Education*, 30(1/2), 9-14.

Squires, D. and Preece, J. (1996). Usability and Learning: Evaluating the Potential of Educational Software, *Computers & Education*, 27(1), 15-22.

Schmidt, P. (1997). Some Knowledge Structures and their use in a Stand-by System for Secondary School Mathematics, *Proceeding of the International Conference on Computers in Education, (ICCE97), Kuching, Malaysia*, pp. 122-129.

Schoenfeld, A.H. (1992). Learning to Think Mathematically: Problem Solving, Metacognition and Sense Making in Mathematics, in D. A. Grouws (Ed), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, 334-370.

Smart School, (1997). <http://www.mdc.com.my/msc/flagship/index.html>

Sommerville, I. (1996). *Software Engineering*, 5th Edition, Addison-Wesley.

Superkids, (1997).

http://www.superkids.com/aweb/pages/reviews/math3/sw_sum1.shtml

Taylor, J.A. (1998). Self Test: A Flexible Self Assessment Package for Distance and Other Learners, *Computers & Education*, 31(1998) 319-328.

Thibodeau, P. (1997). Design Standards for Visual Elements and Interactivity for Courseware, *Technology Horizons in Education Journal* 25(6).

<http://www.thejournal.com/journal/past/feb/0297feat3.html>

Vadera, S. and Meziane F. (1997), Tools for Producing Formal Specifications: a View of Current Architectures and Future Directions, *Annals of Software Engineering*, Volume 3, 273-290.

Authors' Biography

Dr. Farid Meziane is a lecturer in computer Science at the University of Salford, UK. He holds a PhD in computer Science from the University of Salford. His research interests cover Formal Methods, Software Engineering, Requirements Engineering and Artificial Intelligence. His research papers are published in journals that include The Computer Journal and the Annals of Software Engineering.

Chan Weng Onn holds a BSc in Information Technology with major in Software Engineering from the Faculty of Information Technology, University Malaysia Sarawak. He is currently a researcher at MIMOS Berhad, Malaysia. He is specialised in Software Engineering, Multimedia and Network Security

Ms Jane Labadin holds a BSc in Mathematics from Imperial College London and an MSc in Computer Science from UMSIT, Manchester. She is a lecturer in Information Technology at the University Malaysia Sarawak (UNIMAS). Her research interests cover Simulation, real time systems and mathematical modelling. She is currently on study leave at Imperial College studying for a PhD and she is working on the theory and computation of fluid flow around a small hump/obstacle introduced within a boundary layer. This study forms a basis for the studies of fluid flow around tall buildings.

Dr Roger Harris has an MSc. in Business Systems Analysis and Design from the City University of London and a Ph.D. in Information Systems from the City University of Hong Kong. He is a Lecturer in the Faculty of Information Technology at University Malaysia Sarawak (UNIMAS). Roger has occupied a variety of management and consultancy positions in Africa and Asia, advising organisations on process improvement and organisational effectiveness. He is Vice Chair of the International Federation of Information Processing Working Group 9.4 on IT in Developing Countries, representing the Asian region and the program committee chair of the conference on Information Technology in Asia (CITA 99) held in Kuching, Malaysia.

Roger is a member of the editorial advisory boards of the "Journal of End-User Computing", the "Journal of Global Information Management" and "Computer Personnel". He was guest editor for a special edition of the "Journal of Global Information Technology Management", on IT in Developing Countries.