*Alter*nation is an international journal which publishes interdisciplinary contributions in the fields of the Arts and Humanities in Southern Africa.
* Prior to publication, each publication in *Alter*nation is refereed by at least two independent peer referees.
* *Alter*nation is indexed in The Index to South African Periodicals (ISAP) and reviewed in The African Book Publishing Record (ABPR).
* *Alter*nation is published every semester.
* *Alter*nation was accredited in 1996.

COVER DESIGN
R Klopper

CORRESPONDENCE ADDRESS
The Editor: Alternation, CSSALL, Univ. of KwaZulu-Natal, Priv. Bag X54001, 4000 DURBAN, South Africa; Tel: +27-(0)31-260-7343;
Fax: +27-(0)31-260-7286; e-mail: smitj@ukzn.ac.za;
CSSALL web-site: http://www.udw.ac.za:80/~stewartg/
*Alter*nation home page: http://www.udw.ac.za:80/~stewartg/alternat.html
ISSN 1023-1757

*Alter*nation

Interdisciplinary Journal for the Study of the Arts and Humanities in Southern Africa

Informatics in South African Higher Education

Guest Editor
Rembrandt Klopper

2005

CSSALL
Durban

Alternation
Volume 12.1b, 2005
ISSN: 1023-1757

Interdisciplinary Journal for the Study of the Arts and Humanities In Southern Africa

ARTICLES

PRINT CONNECTION Tel (031) 202-7766; 202-7767

Acknowledgement of Reviewers

We wish to acknowledge the contribution of the following reviewers to the production of the special *Informatics* issues of *Alter*nation:

Patricia Alexander
Udo Averweg
Irwin Brown
Ruth De Villiers
Geoff Erwin
Rembrandt Klopper
Kirstin Krauss
Sam Lubbe
Manoj Maharaj
Johan Nel
Shaun Pather
Vijay Rugbeer
Anesh Singh
Marie Spruyt

e-Learning Artefacts: Are They Based on Learning Theory?

MR de Villiers

Abstract

With the advent of e-learning, educators and designers of learning resources should view technology as a tool and a medium, but not as the message. This paper poses the rhetorical question as to whether e-learning artefacts and variants are based on sound learning theory. It traces the evolution of e-learning and describes characteristics that indicate underlying theoretical biases in traditional educational software, as well as in online courses and web-based instruction. The paper introduces a synthesis of contemporary learning theory, the Hexa-C Metamodel (De Villiers, 2002, 2003), whose six elements can play a role in the design and development of e-learning environments and instructional systems, and which can also be used in evaluating educational applications from a learning theory perspective.

Keywords

e-Learning, design of instruction, educational technology, learning theory, interactive learning environments, web-based learning

1 Introduction

The role of technology in learning must be subservient to the envisaged learning outcomes. With the advent of e-learning, educators and designers of learning resources should view technology as a tool and a medium, but not as the message. This paper questions whether e-learning artefacts and environments are based on recognized learning theory. It describes various perspectives on e-learning and characteristics that indicate an underlying theoretical bias, though artefacts may also be hybrids, with foundations emanating from various perspectives. Web-based instruction, in particular, can be characterized by its position in respect of various 'pedagogical dimensions' (Reeves and Reeves, 1997). The paper introduces a synthesis of contemporary learning theory, the Hexa-C Metamodel (De Villiers, 2002, 2003), whose six elements: cognitive learning, constructivism, components, creativity, customization, and collaborative learning, can play a role in the design and development of e-learning environments and instructional systems, and which can also be used in evaluating educational applications from a learning theory perspective.

Section 2 defines e-learning, while Section 3 reviews its development from the computer-aided instruction (CAI) systems of the 1960s to the present, dominated by the pervasive WWW and Internet. Roles and forms of e-learning in different domains are then overviewed and some underlying learning theories described in Section 4. In Section 5 the Hexa-C Metamodel is introduced and its relevance explained.

2 Definition of e-Learning

Some narrow definitions equate e-learning exclusively with the use of the Internet in instruction and learning. However, other definitions are broader. Clark and Mayer (2003:13) define e-learning as 'instruction delivered on a computer by way of CD-ROM, Internet, or Intranet, which ...

- Includes content relevant to the learning objective.
- Uses instructional methods such as examples and practice to help learning.
- Uses elements such as words and pictures to deliver content and methods.
- Builds new knowledge and skills linked to individual learning goals or to improve organisational performance'.

This view thus comprises both content (information) and instructional methods (techniques) to support the process of learning the content, but via a restricted range of hardware and media. The definition adopted by Cedefop (2002:5) is broader still, namely: 'learning that is supported by information and communication technologies (ICT). e-Learning is, therefore, not limited to ... the acquisition of IT competence but may encompass multiple formats and hybrid methodologies, in particular, the use of software, Internet, CD-ROM, online learning or any other electronic or interactive media'. This definition better suits the purposes of this paper, which takes an all-embracing approach, incorporating a broad range of educational technology and types of learning/instruction.

3 Evolution of e-Learning

In this section the advent and evolution of educational computing is overviewed as a background to Section 4, which surveys roles and forms of artefacts and their associated theoretical foundations.

e-Learning artefacts come, and have come, in many variants (Alessi and Trollip, 2001; O'Shea and Self, 1983; Jonassen, 1988), ranging from computer-based tutorials and drill-and-practice software through sophisticated intelligent tutoring systems to state-of-the-art interactive learning environments and problem-solving courseware, where learners use software as tools and not as tutors. One of the first platforms for CAI was PLATO, a project commenced by the University of Illinois in the 1960s and funded by the American National Science Foundation (NSF). PLATO ran on mainframe hardware and eventually enabled sophisticated multi-terminal interactive systems, which integrated text and graphics. It provided instructors with the first programming environment in which they could develop instructional courseware. CAI gravitated to minicomputers in the early 1970s with another NSF project, TICCIT (Time-shared, Interactive, Computer-Controlled Information Television project). It was aimed to be high performance and was based on factory-like production and presentation of course materials. TICCIT also offered the concept of learner-control. However, due to the high costs incurred and the advent of microcomputers, such systems went into demise and the field of instructional computing suffered a setback, losing the benefits of networking (benefits that were only regained with the advent of the Internet).

In the late 1960s/early 1970s the artificial intelligence (AI) community came on the scene with intelligent tutoring systems (ITSs) or intelligent CAI (ICAI). Carbonell (1970) encapsulated these approaches with the term 'AI in CAI' as he described *Scholar*, a mixed-initiative tutor, which interactively presented concepts of geography, permitting both user and computer to share control of the dialogue and ask questions. Other ITSs had self-improvement facilities, whereby the ITS itself would learn by experience. ICAI did not move into the territory of general education, but tended to remain in academic research domains, largely because building intelligent software to simulate human tutors takes man-years of expert programming. During the 1980s and 1990s, conventional CAI in the form of tutorials and drill-and-practice software became popular, particularly in the USA and the UK. Such courseware was stand-alone, running off diskettes on micro- and desktop computers. Drill-and-practice software offers exercises in basic skills. The computer stores and randomly presents practice items to support specific instructional objectives. It provides record keeping and different levels of difficulty, often placing the learner on a level according to achievement in a pre-test. There may be an explicit 'game' ethos to extrinsically motivate performance. Tutorials are the classic instructional programs, using interactive dialogue to coach learners. Teaching segments are typically alternated with question segments, which respond with diagnostic feedback based on the learner's input. Unfortunately such systems have their origins in programmed instruction and operant conditioning, based on Skinnerian behaviourism and epitomized by the stimulus-response-reinforcement paradigm, an approach in which learners are largely treated as passive

recipients of information (Alessi and Trollip, 2001; O'Shea and Self, 1983).

In attempts to support more active learning, problem-solving software (Jonassen, 1988) originated, allowing learners to take active roles with the computer not as tutor, but as tutee or a tool. Other developments are open learning environments (Kok and Poorthuis, 1990), where students undertake limited exploration in defined task situations. The aim is to support them in independent acquisition of knowledge as they solve given problems. In another venture, physical classrooms are converted to collaborative electronic environments designed for active learning (Shneiderman *et al.*, 1998).

In this chronicle of e-learning, issues that have been noted – implicitly or explicitly – as sound features, include learner-control, networking, cost-effective production and dissemination, use of computing systems as tools instead of tutors, and the independence offered by active exploration rather than passive transfer. Some of these features occurred in certain systems, only to be sidelined in the next generation. Yet all of them are now incorporated and globally available within current technology, which has been transformed by the advent of the World Wide Web. The milieu of interactive educational computing has become an accessible resource at affordable prices in the form of portals, educational content websites, learning management systems, communication forums, etc. on the pervasive Internet.

In line with democratic paradigm shifts, the role of the learner is becoming more active, while the educator is viewed as less of an instructor and more of a facilitator.

4 e-Learning: Domains, Roles, and Underlying Learning Theories

This section outlines various perspectives on educational computing, considering, first, the type of domain from which the learning content is extracted and, second, the learning theories underlying e-learning artefacts. Third, there is a categorization of the types of systems and the roles that technology can assume in the learning process and finally, the relationship between an artefact and its software construction process is addressed.

4.1 Type of Domain

In overviewing the underlying theoretical bias of e-learning artefacts, this paper first considers the type of domain from which the content to be taught or learned, originates. (Note the distinction between teaching, i.e. the role of the educator, and learning, i.e. the role of the student/pupil/learner.) There is a major difference between *well-structured* domains and *ill-structured* domains (De Villiers, 2003; Hannafin, Land and Oliver, 1999; Jonassen, 1999; Landa, 1998):

- Well-structured or closed domains contain concepts termed tightly defined, procedural or algorithmic, for example, syntactic, mathematical, scientific and computational subject matter, where rules and procedures are prescribed and problems solved by objective principles.
- Ill-structured or open domains contain problems with multiple solutions and alternative approaches, some aspects of which emerge only during the problem-solving process, for example, social sciences, management sciences, environmental disciplines and design disciplines. They require reflective practice (Schön, 1987) and heuristic, expert-type knowledge.

4.2 Theories of Learning and Cognition as Foundations of e-Learning

The functionality of e-learning applications is not conventional data processing as in commercial operations that process *business transactions*. Rather, e-learning entails supporting learners in the process of learning. It involves:

- Information transfer rather than information translation,
- Managing educational interaction,

- Supporting human cognition,
- Implementing behavioural change, and
- The leverage of technology as a medium and messenger, rather than as a message or showpiece in its own right.

In short, an *instructional transaction* entails the effective transfer or presentation of knowledge and skills, and unlike a business transaction, is not carried out by a professional doing his/her task in the workplace, but frequently by a novice, for whom the computer is a means of learning, not earning. It is therefore particularly important that the functionality of instructional systems is easy to learn and to use. Foundations for e-learning must be based on sound principles of learning theory and instructional design, in order to facilitate effective learning.

e-Learning applications reflect diverse views on cognition and learning, which are portrayed in Figure 1. The didactic approach views learning primarily as the acquisition by learners of knowledge structures and skills. This approach may tend to fixed transfer of information, and can be traced to the psychological school of *behaviourism* (Skinner, 1938). The associated instructional design models are somewhat rigid (Dick and Carey, 1996) and frequently program-controlled, although user-control and interactivity are on the increase. While behaviourism concentrates on shaping the learner's behaviour, the *cognitive* approach (elaborated in Section 5.1) emphasises the mental processes involved in learning. Greeno (1991) supports this classification, consolidating the roles of computing in education into *didactic* and *exploratory*. In the didactic approach, computers present information in a systematic individualized manner. The learning experience should be 'efficient' in terms of minimum errors on the part of the learner. Cognition is viewed as a mental system of information structures and procedures, and learning as the acquisition of these. The exploratory view treats learning less systematically, as the computer system presents phenomena that learners can investigate through interaction, discovering the transformations and constraints, e.g. Logo (Papert, 1988). This view fits a theory that considers cognition as situated in social and physical contexts.

At the extreme end of the spectrum is the *constructivist* ethos (Inhelder and Piaget, 1958; Bruner, 1994), which is elaborated in Section 5.2. This involves an open-ended, flexible, exploratory view, which situates learning contextually, encouraging learner-centricity and active learner-generation of products. Learning is scaffolded, rather than tutored.

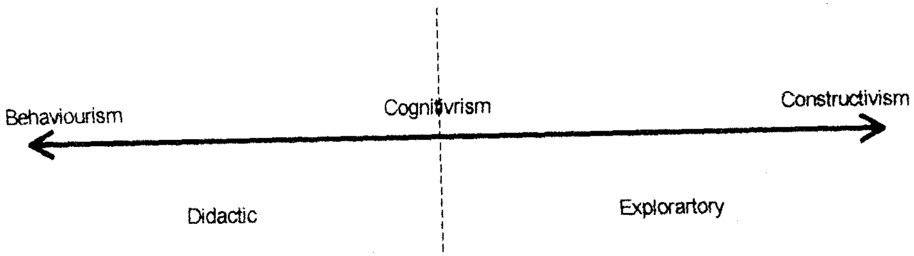


Figure 1: Spectrum of leaning theories

A further set of artefacts and weblications have no explicit theoretical foundation and are *pragmatically* constructed. This can occur, among others, in corporate vocational training, where designers may in fact, instinctively do 'the right thing', but may also produce systems with image appeal but superficial educational worth.

Explicit pedagogy should be integrated into learning resources. This is particularly important in online instruction and web learning environments (Firdyiwek, 1999; Winn, 1999), which are frequently used in distance learning, where misconceptions are harder to rectify.

Web-based instruction (WBI) and -learning (WBL) can be characterized by positions in respect of various 'pedagogical dimensions' (Reeves and Reeves, 1997). Based on research and theory in instructional technology, cognitive science and adult learning, Reeves and Reeves define ten dimensions of interactive learning that can be en-

abled via the WWW, namely: (1) pedagogical philosophy, (2) learning theory, (3) goal orientation, (4) task orientation, (5) source of motivation, (6) role of the teacher, (7) metacognitive support, (8) collaborative learning, (9) cultural sensitivity, and (10) structural flexibility. Each of these can be represented in a simplified manner by considering it as a continuum between two extremes. Eight of the Reeves dimensions are particularly relevant to this paper: *Pedagogical philosophy* varies from strict instructivism to radical constructivism, where instructivism is based on the teaching of rigid structured objectives, viewing learners as passive recipients. *Learning theory* (as already stated) spans two dominant approaches in instructional design – the behavioural (based on the shaping of observable behaviours) and the cognitive (emphasizing internal mental states). *Task orientation* relates to the context of learning, whether it is abstract, such as a typical academic exercise, or authentic, e.g. training in a real-world activity. *Motivation* varies from extrinsic – the gold star, bells-and-whistles syndrome, to intrinsic motivation, where the learner's locus of control is integral to the learning. *Educators' roles* range between a didactic 'sage on the stage' and the facilitative, 'guide on the side'. *Metacognitive support* refers to the learner's own awareness of objectives, ability to plan and monitor one's own progress, access to resources or information, and the ability to adjust actions and reactions to accommodate requirements. The level of support can vary between unsupported and integrated. *Cultural sensitivity* is of paramount importance. Although it is unlikely that WBI can be adaptive to every cultural norm, websites should be respectful, avoiding insensitivity in terms of icons, symbols, graphics and terminology, so as to accommodate diverse ethnic and cultural backgrounds. Finally, there is *structural flexibility* with respect to the time and/or place of learning. This is related to the concept of the fixed instructional mode in a traditional academic setting as opposed to open education. One strength of WBI is its asynchronous nature, independent of place or time.

Learning artefacts and events can be measured on continuums, according to these dimensions, and may well occupy central positions that show them to be hybrids.

4.3 Roles of Technology in Learning

Another perspective on technology in instruction and learning is based on the learner-content relationship as defined in Winn's (1992) *full* and *empty* instructional technologies. A full technology contains information to be transferred interactively to students, e.g. tutorials. An empty technology is a shell that supports exploration, communication, and construction, e.g. searches and generation of products using the Internet, and use by learners of commercial software as tools for manipulation, documentation, and generation of deliverables. A list follows of e-learning applications and approaches. Although a rigid categorization is impossible, since use varies according to the context and the users, this arrangement tends to be from the full to the empty.

- Direct instruction via online textbooks/workbooks;
- Drill and practice;
- CAI tutorials;
- Interactive television;
- Multi-media productions;
- Electronic classroom lectures;
- Simulations;
- Educational games;
- Interactive learning/practice environments;
- CD ROMs, dynamic Web-based resources, online courses;
- Open-ended learning environments and constructivist learning environments;
- Immersive virtual reality technology;

- Forums for educator-learner, learner-learner, and learner-external world communication:
  - Asynchronous examples: e-mail (which can be threaded), newsgroups, bulletin boards, listservs;
  - Synchronous examples: chat rooms, multi-user domains (MUDs), object-oriented MUDs (MOOs), video- or audio-conferencing; and
- Electronic portfolios of learners' work.

4.4 Relationship between an e-Learning Product and Its Construction Process

Another consideration relates not just to the *product*, but to its software development *process*. Trends in learning theory should be translated into reality in such a way that there is synergy between the underlying theoretical ethos, the development environment and the instructional strategies. There is a close relationship between the development paradigm and the resultant product (Winn, 1999).

For example, early procedural programming languages were process-oriented, describing steps with fixed algorithms. For educational applications this naturally generated systems that led learners through predetermined instructional sequences and activities, the nature of which was completely prescribed, yet which was appropriate for the predominantly behaviourist pedagogical ethos of the time.

Object-oriented programming and web-programming, on the other hand, require the developer to construct objects and operations that these objects should execute when specified events occur. Sequences are not prescribed and systems are open-ended. Such techniques radically alter the ways in which users act within those environments. They lend themselves to flexible interaction in rich multimedia environments, in harmony with cognitive and constructivist learning theory approaches. They are learner-centric in that students take the initiative and the environment responds, and are thus well suited to inquiry-based and problem-based learning.

5 The Hexa-C Metamodel

Various factors impact on determining suitable theoretical foundations for a learning resource or artefact. No single paradigm is appropriate for all situations – domain, context and content all play roles. One particular approach may be relevant, or a hybrid approach. Technology should be matched to the pedagogical philosophy, as teaching/learning patterns are identified (Shneiderman *et al.*, 1998). The question arises as to whether some systems are based on theory at all. Are they are pragmatically constructed to suit the technology, ignoring theory or using an eclectic mix? Evaluation from the perspective of learning theory can be conducted in an attempt to answer this question for a particular application.

A six-sided approach, the Hexa-C Metamodel (HCMm) (De Villiers, 1999; 2002; 2003) integrates concepts from contemporary learning theory into a framework which serves as a design aid and for evaluating existing resources from the perspective of learning theory. It is termed 'Hexa-C' because each of its six inter-related elements commences with the letter 'C', and 'Metamodel', because it is a synthesis of existing theories, models, and paradigm. The HCMm incorporates several concepts addressed thus far in this paper and concisely suggests six elements to which the designer, educator or evaluator should pay cognizance, determining which are relevant to an intended or existing learning environment and, if relevant, to what extent and in what way the system should conform to those elements. Although the HCMm can be applied to any learning resource, the focus here is its relevance to e-learning and educational technology.

Three of the Cs: *constructivism*, *cognitive learning theory* and *components*, are primarily theoretical, while the others: *collaborative learning*, *creativity*, and *customization*, are practical means used by educators to foster effective and affective learning. Figure 2 shows the

hexagonal framework of the HCMm, representing its elements as merging segments around the hub of *technology*. This indicates technology's role as the mechanism that transfers the message, but not the message itself. The whole is embedded in *context*, emphasizing that the nature of each e-learning artefact or environment should be determined by its content and situation.

The HCMm has been used in evaluations of diverse learning systems, using triangulated data, both qualitative and quantitative (De Villiers, 2000; De Villiers and Cronje, 2001; De Villiers and Dersley, 2003).

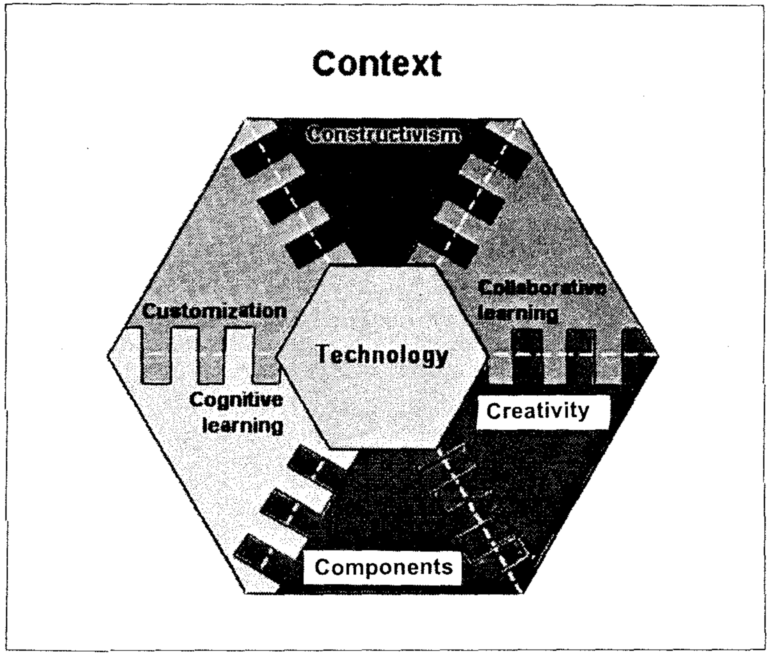


Figure 2: The framework of the Hexa-C Metamodel

The six HCMm elements are now introduced. It is not the intention that every resource or instructional system should conform to them all. The framework can support designers and practitioners in a multi-faceted approach to effective and affective learning, but means of implementation vary. Each application, with its target learners and stage of learning, has a unique focus and is characterized by its educational content, problem-solving methods, and its physical/virtual situation. For example:

- Is the content domain well-structured or ill-structured?
- Is the instruction formal or informal; full-time or part-time; primary, secondary or tertiary education; market-related or workplace training?
- Is the context one of continuing education; distance-learning or contact-teaching; isolated or collaborative; based in a lab or at home?
- Are there formally graded outcomes?
- Does the application require a full or an empty technology?

5.1 Cognitive Learning Theory

Learning should support cognition, retention, and transfer. Cognitivism relates to the results of cognitive processes such as the formation of mental models, human information processing, metacognition, and self-regulation. New knowledge should be integrated with prior learning, building new skills on previous knowledge. Cognitive processes are seen as being as important as generating learning products. Cognitive learning aims to foster critical thinking skills by authentic problem solving or by explicit teaching of cognitive strategies alongside content knowledge (Anderson, 1983; Gagné and Merrill, 1990; Inhelder and Piaget, 1958; Minsky, 1975; Newell and Simon, 1972; Osman and Hannafin, 1992; Reigeluth, 1999; Reigeluth and Moore, 1999; West, Farmer and Wolff, 1991; Winn, 1990).

Even in closed domains, cognitive learning can be fostered. The traditional tutorial program can evolve beyond its behaviourist roots into a cognitive system. Although the tutorial process is systematically structured, with given information and predetermined relationships between chunks of content, features such as high interactivity, animated process/object depiction and individualized diagnostic feedback, can support cognition. The relationship between theory and application may be addressed deductively – introducing concepts and rules before applying them, or inductively – moving from examples to underlying theory. Advanced 'challenge' activities should be provided for self-regulation and to stimulate higher-order thinking skills (HOTS).

5.2 Constructivism

Constructivism relates to personal knowledge construction and interpretation, active learning, anchored instruction, and multiple perspectives on an issue. Constructivist mechanisms include problem/project-based learning, open-ended learning environments, flexible learning within ill-structured domains, and authentic tasks – without simplification of complexity. Constructivism is not direct instruction; rather, it entails setting up learner-centric environments and activities, within which learners can explore and undertake discovery learning. Where possible, tasks should be authentic. The constructivist approach may use multiple modes of presentation (audio, visual, textual, interactive, etc.). It aims to instil personal goals and active involvement within real-world situated learning, leading to application skills and transfer. It emphasizes collaborative activities and learner-research using a wide variety of resources (Bruner, 1967; 1994; Cunningham, 1992; Duffy and Jonassen, 1991; Hannafin, 1992; Hannafin *et al.*, 1997; Jonassen 1994; 1999; Land and Greene, 2000; Lebow, 1993; Perkins, 1991; Savery and Duffy, 1995; Willis, 2000; Winn, 1992).

For project- and problem-based learning, software tools can be used directly by learners to search out information and to manipulate and present it using, for example, the WWW as an information resource, spreadsheets as cognitive tools to display findings and manipulate multiple parameters, and databases for storage and inquiry. (Information from the WWW should, however, be subjected to quality tests, checking its accuracy, authority, currency, uniqueness, links, and writing quality (Smith, 1997).) In the design disciplines, graphics packages and animation can be used to convey information. Learners can develop multi-media products and create websites on which to post their work. Real-world activities enforce standards beyond the norm for academic efforts, demanding superior efforts and can result in constructivist frustration. In some cases, beyond academia, real-world projects become real-life products, usable in the workplace or the market.

5.3 Components

Components within learning and instruction (Reigeluth, 1999) relate to the basic knowledge, skills and methods of a domain. One approach is component display theory (CDT) (Merrill, 1983), based on relationships between the kind of content taught (fact, concept, procedure, and principle) and the level of performance required (remember, use, or find). CDT examines whether the instructional strategies used in a learning event achieve its instructional goals. Each learning objective is related to the appropriate content and desired performance, resulting in an instructional component that is positioned in a performance-content matrix.

Componential instruction is more relevant in well-structured domains, where explicit teaching is needed of the basic knowledge and skills, often in decontextualized settings. Unitary components can be integrated to form composite components (De Villiers, 2002). Merrill (2001) emphasizes the role of instructional components as theoretical tools to facilitate the design of effective, efficient and appealing instructional products, both in directive tutorials and in environments for experiential learning. In the latter, knowledge of the basics should be assumed, but linked access can be provided to subject matter resources.

5.4 Creativity and Motivation

Creativity supports the affective aspects of instruction, aiming for novelty within functionality in ways that motivate learners intrinsically

(Caropreso and Couch 1996; De Bono, 1970; Dick 1995; Jones, 1998; Malone, 1981). Creative instruction aims to incorporate affective aspects within learning, seeking to apply innovative instructional strategies, to engage learners, and to strengthen the affective-cognitive bond (Price, 1998; Wager, 1998), whereby values and emotions influence learners' initial ability to acquire knowledge (external affective aspects) and their ongoing attitude and perseverance (internal affective aspects).

Creativity is closely connected to intrinsic motivation and engagement of learners. Application of the ARCS model (Keller and Suzuki, 1988) suggests that instruction should: gain attention, ensure relevance, instil confidence, and lead to learner-satisfaction. Creativity is rewarded when learners experience 'flow' (Csikszentmihalyi, 1990), forgetting time and tackling more than envisaged!

Creativity can be in evidence, first, in the learning environment and second, in the activities or product development undertaken by learners. Novel, innovative environments and presentations help to engross learners, provided that such means or themes are inherently part of the learning experience and not creeping featurism (Norman, 1998) or distractive 'bells-and-whistles'. Multimedia can support creativity, also serving as multi-gate reinforcement. Open learning systems, such as web-based 'virtual classrooms', have scope for informality and humour, particularly in learner-learner contact on discussion forums. HCMm research has shown that creativity fosters creativity, as novel instructional situations stimulate learners, in turn, to De Bono's (1970) lateral thinking.

5.5 Customization

The movement towards customizing/individualizing learning (Alessi and Trollip, 2001; Bruner, 1967; Norman and Spohrer, 1996; Reigeluth, 1999) aims for learner-centric instruction that adapts to individual profiles, supports personal processes and products, and allows learners to take initiative regarding (some of) the methods, time, place, content, and sequence of learning. It supports learner-control, negotiated goals, and the ethos of matching learners' needs and interests.

Classic CAI incorporated program-controlled 'branching', navigating learners through material according to their performance. Learner-controlled customization, by contrast, allows learners some say regarding time and place of learning, tasks, modes and media, levels of difficulty, and degree of help.

True learner-centricity is enabled in unstructured domains where learners tackle open-ended projects. Auto-customization and customization by content occur as learners:

- Choose their own approach within set content, using tools and techniques in ways that are personally optimal;
- Determine own content and direction within a broad domain, developing own product; or
- Customize learning by choosing between options when they do tasks/assignments and by taking specialized roles within teams.

In problem-based contexts, learners should direct the learning, conduct independent research, collect, analyse and manipulate information, draw conclusions and present findings.

5.6 Collaborative Learning

Collaborative learning involves joint work, social negotiation, a team approach, accountability, and peer evaluation, i.e. sharing of responsibility within a group. It optimizes on complementarity and instils collaborative skills in learners (Johnson and Johnson, 1991; Nelson, 1999; Panitz, 1996; Singhanayok and Hooper, 1998). Collaboration is closely associated with constructivism and open learning, and is applied in project-based and problem-based learning.

Collaborative learning is not usually considered suitable for algorithmic tasks with tightly defined procedures, yet experience shows spontaneous joint use, two-at-a-computer, interacting with artefacts intended for individual use. Co-operative problem solving and peer-

teaching can be effective means of learning and confidence building. In less structured tasks and projects, role allocation should capitalize on skills and strengths and support weaknesses, providing an efficient approach where complex knowledge and varied expertise are required. It is excellent preparation for the real world and the workplace. A further form of collaborative participation is electronic voting within a physical or virtual classroom.

5.7 Application and Relevance of the C-Elements

The HCMm and its constituent elements have been introduced. Can these elements be converted directly into principles that set out underlying theoretical foundations for the phenomenon of e-learning? No, not directly, but in adapted contextualized forms, yes. For example, sets of evaluation criteria/questions have been compiled and applied in specific survey evaluations and in heuristic evaluations (De Villiers, 2000; De Villiers and Cronje, 2001; De Villiers and Dersley, 2003). However, no single set of design guidelines or evaluation criteria would be generally applicable since artefacts differ in terms of underlying domain, subject matter, situation of use, and the purpose – to teach, tutor, be a tutee or toolset, or to serve as an exploration environment.

It is up to the educator or designer of e-learning to take the HCMm as a conceptual framework and translate its elements into principles, design guidelines, and evaluation criteria appropriate for the specific requirements. Evaluations may be conducted on existing applications to determine whether they are grounded in learning theory. More important, for designers and would-be designers, there is the opportunity to construct new systems and environment that are constructed on a foundation of learning theory, either with a purist ethos or by combining various theoretical stances which are coherent and consistent with each other.

At the risk of simplification, and while acknowledging exceptions, the matrix in Figure 3 lists forms of e-learning against the C-elements, marking with an **x** those appropriate for consideration in the design of a particular artefact or system. The large **X**s indicate stronger relationships.

	Components	Cognitive learning	Constructivism	Customization	Collaborative learning	Creativity
Online books	X	x		x		x
Interactive television	X	X		x		x
Drill and practice	X			X		
Tutorials	X	X	x	X	x	X
Multimedia productions	x	x	x		x	X
Electronic class lectures	X	x			x	x
Simulations		x	X	X	x	x
Educational games	x		X	X	X	x
Interactive and open learning environments			X	X	X	X
CD ROMs	x	x	x	x	x	x
Online courses	X	x	x	X	x	x
Dynamic WBI	x		X	X	x	X
Virtual reality			x	x		X
Asynchronous communication forums: email, newsgroups, etc		x	x	X	X	
Synchronous communication forums: chat rooms, etc		x	x	x	X	
Electronic portfolios			X	X		X
Video / audio conferencing				X	X	

Figure 3 Relationship between HCMm elements and types of e-learning

6 Conclusion

This paper overviews the concept and evolution of e-learning and investigates the theoretical foundations of educational artefacts. The HCMm is presented as a conceptual framework which integrates tenets of contemporary learning theory. The design and implementation of a variety of e-learning systems can be enhanced by considering these

elements and explicitly applying those appropriate to the domain. N single paradigm is appropriate for all situations, but its elements can b translated into principles, design guidelines, and evaluation criteri customized for specific domains, subject matter and contexts. The in structional designer, e-learning practitioner or educational web devel oper should ensure that technology serves as the hub which transfer the message and in no way detracts or distract from the message. Fi nally, the question posed in the title of this article, namely whether e Learning artefacts are based on learning theory, can be answered in th affirmative.

References

Alessi, SM & SR Trollip 2001. *Multimedia for Learning: Methods anc Development*. Englewood Cliffs, N.J.: Prentice Hall.

Anderson, JR 1983. *The Architecture of Cognition*. Cambridge MA Harvard University Press.

Bruner, JS 1967. *Toward a Theory of Instruction*. Cambridge Mass Harvard University Press.

Bruner, J 1994. Constructivist Theory. http://tip.psychology.org/bruner.html

Carbonell, J.R. 1970. AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Studies* 11: 190-202.

Caropreso, EJ & RA Couch 1996. Creativity and Innovation in Instructional Design and Development: The Individual in the Workplace. *Educational Technology* 36(6):31-39.

Cedefop (European Centre for the Development of Vocational Training) 2002. *E-learning and training in Europe*. Luxenbourg Office for Official Publications of the European Communities.

Clark, RC & RE Mayer, 2003. *E-Learning and the Science of Instruction*. San Francisco: Pfeiffer.

Csikszentmihalyi, M 1990. *Flow: the Psychology of Optimal Experience*. New York: Harper and Row.

Cunningham, DJ 1992. Assessing Constructions and Constructing Assessments: A Dialogue. In TM Duffy & DH Jonassen (Eds.) *Constructivism and the Technology of Instruction: A Conversation*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

De Bono, E 1970. *Lateral Thinking: A Textbook of Creativity*. Harmondsworth: Penguin Books.

De Villiers, MR 1999. Applying the Hexa-C Metamodel of Instructional Theory and Design to Educational Web Applications. In P De Bra & J Leggett (Eds.) *Proceedings of Web Net 99 – World Conference on the WWW and Internet*. Honolulu: Association for the Advancement of Computing in Education.

De Villiers, MR 2000. Evaluation of an Interactive Learning Environment: Theory and Practice. *South African Journal of Higher Education* 14(3):120-131.

De Villiers, MR 2002. *The Dynamics of Theory and Practice in Instructional Systems Design*. Unpublished PhD dissertation. Pretoria: University of Pretoria.

De Villiers, MR 2003. Foundation for Structure: A Learning Theory Approach to Instructional Design for e-Learning. *International Journal of Learning Vol 10 2003/04*. Proceedings of the Learning Conference 2003. Australia: Common Ground Publishing. .

De Villiers, MR & JC Cronjé 2001. Co-Operative Learning in an Immersed Internet-Based Virtual Learning Community. *Proceedings of Web Net 2001– World Conference on the WWW and Internet*. Orlando, USA.

De Villiers, MR & R Dersley 2003. Emerging Technology for Emerging Peoples: Evaluation of an African AIDS-Awareness Training System. *Proceedings of Ed-Media 2003 World Conference on Educational Hypermedia and Telecommunications*. Honolulu, USA.

Dick, W 1995. Instructional Design and Creativity: A Response to the Critics. *Educational Technology* 35(4):5-11.

Dick, W & L Carey 1996. *The Systematic Design of Instruction* (4th Edition). New York: Longman.

Duffy, TM & DH Jonassen, 1991. Constructivism: New Implications for Instructional Technology? *Educational Technology* 31(5):7-12.

Firdyiwek, Y 1999. Web-Based Courseware Tools: Where Is The Pedagogy? *Educational Technology* 39(1):29-34.

Gagné, RM & MD Merrill 1990. Integrative Goals for Instructional Design. *Educational Technology Research and Development* 38(1):23-30.

Greeno, JG 1991. Productive Learning Environments. In R Lewis, & S Otsuki (Eds) *Advanced Research on Computers in Education*. North Holland: Elsevier Science Publishers.

Hannafin, MJ 1992. Emerging Technologies, ISD, and Learning Environments: Critical Perspectives. *Educational Technology Research and Development* 40(1):49-63.

Hannafin, MJ, KM Hannafin, SM Land & K Oliver 1997. Grounded Practice and the Design of Constructivist Learning Environments. *Educational Technology Research and Development* 45(3):101-117.

Hannafin, M, S Land & K Oliver 1999. Open Learning Environments: Foundations, Methods, and Models. In CM Reigeluth (Ed.) *Instructional-Design Theories and Models Volume II: A new paradigm of instructional theory*. Mahwah, N.J.: Lawrence Erlbaum Associates.

Inhelder, B & J Piaget, 1958. *The Growth of Logical Thinking from Childhood to Adolescence*. New York: Basic Books Inc. Publishers.

Johnson, DW & RT Johnson, 1991. *Learning Together and Alone*. Englewood Cliffs, NJ: Prentice Hall.

Jonassen, DH 1988 (Ed.) *Instructional Designs for Microcomputer Courseware*. Hillsdale. NJ: Lawrence Erlbaum Associates.

Jonassen, DH 1994. Thinking Technology: Towards a Constructivist Design Model. *Educational Technology* 34(4):34-37.

Jonassen, D 1999. Designing constructivist Learning Environments. In CM Reigeluth (Ed.) *Instructional-Design Theories and Models Volume II: A New Paradigm of Instructional Theory*. Mahwah, N.J.: Lawrence Erlbaum Associates.

Jones, MG 1998. *Creating Engagement in Computer-Based Learning Environments*. ITForum Paper #30. http://it.coe.uga.edu/itforum/paper30/paper30.html

Keller, JM & K Suzuki, 1988. Use of the ARCS Motivational Model in Courseware Design. In DH Jonassen (Ed.): *Instructional Designs for Microcomputer Courseware*. New Jersey: Lawrence Erlbaum Associates.

Kok, W & G Poorthuis 1990. The Effects of Different Teaching Strategies in Three Cai Programs within the Same Content Area. In JM, Pieters, PRJ Simons & L De Leeuw (Eds.): *Research on Computer-Based Instruction*. Amsterdam: Swets and Zeitlinger.

Land, SM & BA Greene 2000. Project-Based Learning with the World Wide Web: A Qualitative Study of Resource Integration. *Educational Technology Research and Development* 48(1):45-68.

Landa, LN 1998. Methodological Drawbacks of Current Educational Software and Suggestions for Radical Improvement Based on Algo-Heuristic Theory. *Educational Technology* 38(5):54-56.

Lebow, D 1993. Constructivist Values for Instructional Systems Design: Five Principles Toward a New Mindset. *Educational Technology Research and Development* 41(3):4-16.

Malone, TW. 1981. Toward a Theory of Intrinsically Motivating Instruction. *Cognitive Science* 4:333-369.

Merrill, M.D 1983. Component Display Theory. In CM Reigeluth, (Ed.) *Instructional Design Theories and Models: An Overview of Their Current Status*. Hillsdale, N.J: Lawrence Erlbaum Associates.

Merrill, MD 2001. Components of Instruction toward a Theoretical Tool for Instructional Design. *Instructional Science* 29: 291-310.

Minsky, ML 1975. A framework for Representing Knowledge. In PH Winston, (Ed.) *The Psychology of Computer Vision*. New York: McGraw-Hill.

Nelson, LM 1999. Collaborative Problem Solving. In CM Reigeluth (Ed.) *Instructional-Design Theories and Models Volume ii: A New Paradigm of Instructional Theory*. Mahwah, N.J: Lawrence Erlbaum Associates.

Newell, A & HA Simon, 1972. *Human Problem Solving*. Englewood Cliffs, N.J: Prentice-Hall Inc.

Norman, DA 1998. *The Design of Everyday Things*. London: MIT Press.

Norman, DA & JC Spohrer, 1996. *Learner-Centered Education*. ITForum Paper #12. http://it.coe.uga.edu/itforum/paper12/paper12.html

O'Shea, T, & J Self 1983. *Learning and Teaching with Computers*. Brighton: The Harvester Press.

Osman, ME & MJ Hannafin 1992. Metacognition Research and Theory: Analysis and Implications for Instructional Design. *Educational Technology Research and Development* 40(2):83-99.

Panitz, T 1996. *A Definition of Collaborative Versus Cooperative Learning*. http://www.lgu.ac.uk/deliberations/collab.learning/panitz2.html

Papert, S 1980. *Mindstorms: Children, Computers and Powerful Ideas*. New York: Basic Books.

Perkins, DN 1991. What Constructivism Demands of The Learner. *Educational Technology* 31(9):19-21.

Price, EA 1998. Instructional Systems Design and the Affective Domain. *Educational Technology* 38(6):17-28.

Reeves, TC & PM Reeves, 1997. Effective Dimensions of Interactive Learning on the World Wide Web. In BH Khan (Ed.): *Web-Based Instruction*. Englewood Cliffs, N.J New Jersey: Educational Technology Publications.

Reigeluth, CM 1999. What Is Instructional-Design Theory and How Is It Changing? In CM Reigeluth, (Ed.): *Instructional-Design*

Theories and Models Volume II: A New Paradigm Of Instructional Theory. Mahwah, N.J: Lawrence Erlbaum Associates.
Reigeluth, CM & J Moore, 1999. Cognitive Education and the Cognitive Domain. In CM Reigeluth, (Ed.): *Instructional-Design Theories and Models Volume II: A New Paradigm of Instructional Theory.* Mahwah, N.J: Lawrence Erlbaum Associates.
Savery, JR & TM Duffy, 1995. Problem Based Learning: An Instructional Model and Its Constructivist Framework. *Educational Technology* 35(5):31-38.
Schön, D 1987. *Educating the reflective practitioner.* San Francisco: Jossey-Bass Publishers.
Shneiderman, B, EY Borkowski, M Alavi, & K.Norman 1998. Emergent Patterns of Teaching/Learning in Electronic Classrooms. *Educational Technology Research and Development* 46 (4): 23-42.
Singhanayok, C & S Hooper 1998. The Effects of Cooperative Learning and Learner Control on Students' Achievement, Option Selections, and Attitudes. *Educational Technology Research and Development* 46(2):17-33.
Skinner, BF 1938. *The Behaviour of Organisms: An Experimental Analysis.* New York: Longman.
Smith, AG 1997. Testing the Surf: Criteria for Evaluating Internet Information Resources. *The Public-Access Computer Systems Review* 8(3).
http://info.lib.uh.edu/pr/v8/n3/smit8n3.html
Squires, D 1999. Educational software for Constructivist Learning Environments: Subversive Use and Volatile Design. *Educational Technology* 39(3):48-53.
Wager, W 1998. Social determinants of Affective Behavior and Learning. *Educational Technology* 38(6):15-16.
West, CK, JA Farmer & PM Wolff 1991. *Instructional Design: Implications from Cognitive Science.* Englewood Cliffs, N.J.: Prentice Hall.

Willis, J 2000. The Maturing of Constructivist Instructional Design: Some Basic Principles that Can Guide Practice. *Educational Technology* 40(1):5-16.
Winn, W 1990. Some implications of Cognitive Theory for Instructional Design. *Instructional Science* 19(1):53-69.
Winn, W 1992. The Assumptions of Constructivism and Instructional Design. In TM Duffy & DH Jonassen (Eds.): *Constructivism and the Technology of Instruction: A Conversation.* New Jersey: Lawrence Erlbaum Associates.
Winn, W 1999. Learning in Virtual Environments: A Theoretical Framework and Considerations for Design. *Educational Media International* 36(4): 271-279.

Author's Contact Details

M R de Villiers (DVILLMR@unisa.ac.za)
School of Computing
University of South Africa, Pretoria, South Africa

The Development and Evolution of an Information Systems Honours Module in e-Learning

BJ Mallinson
DA Sewry

Abstract

This paper describes the introduction and evolution of an e-Learning module for Information Systems students, from a half-module in 2001 through to its current form as a full module within the Information Systems Honours / Fourth year Programme. The opportunity to use a blended approach to module delivery has afforded a wealth of pedagogical opportunities for the module facilitator, and proved to be an exciting learning environment for the learners, while at the same time taking advantage of the 'medium is the message'. Whilst few problems have been experienced with regard to implementing the desired technologies for the module, it has become evident that the blended or hybrid mode of module delivery is best suited to a more lengthy delivery time span.

1. Introduction

In accordance with international trends, there is an increased emphasis on education and training in commerce and industry in South Africa. It is particularly interesting to note the innovative developments in the use of technology to facilitate teaching and learning. The e-Learning module at the Honours Fourth year level in Information Systems (IS) at Rhodes University (RU) was initiated in 2001 (Information Systems Department, 2001) as a response to this trend, and provides an introduction to and exploration of online learning technologies that would lead to the implementation of a variety of e-Learning systems and environments. The module is designed to equip IS Hon-

*Alter*nation 12.1b (2005) 373-396 *ISSN 1023-1757*

ours graduates with the foundational knowledge, skills and attitudes necessary to participate in the formulation of e-Learning policies, the design and implementation of e-Learning systems, the critical analysis of current e-Learning technologies, methodologies and practise, and to enable the learners to conduct further research within this highly topical and rapidly evolving field.

This elective module has evolved from a half-module in 2001 through to its current form as a full module within the IS Honours / Fourth year Programme (Information Systems Department, 2004). The content of the module has grown from an initial view of Online Educational Systems through to encompassing a wider spectrum of e-Learning Tools and Technologies (Mallinson, 2004). A feature of the module has been the blended or hybrid model that was adopted from the outset, comprising face-to-face meetings, synchronous and asynchronous online conferencing, learner self-study and group project work. However, many of the decisions made concerning the structure and timing of the module have been dictated by constraints within the scheduling and context of a regular on-campus environment.

While few problems have been experienced with regard to implementing the desired technologies for the module due to the high level of technical expertise within the Department together with a well supported software and hardware environment within the Hamilton Building in which the Department is housed, it has become evident that this mode of module delivery is best suited to a more lengthy delivery time. That is, the very nature of asynchronous online communication and a constructivist learning environment dictates that the learning experience should ideally be spread over a longer period than is currently scheduled for a full module offering in the IS Dept Honours programme. The opportunity of the use of a hybrid approach to module delivery has afforded a wealth of pedagogical opportunities for the module facilitator, and proved to be an exciting learning environment for the learners, while at the same time taking advantage of the medium being the message.

Section 2 describes the current University and Departmental environment, Section 3 describes the motivation for mounting an e-Learning module within the IS Department, and Section 4 describes the research design. Section 5 describes the evolution of the module curriculum, Section 6 presents an analysis of learner evaluations and facilitator reflections, and Section 7 concludes the paper.

2. Environment

The University is of a traditional, residential nature and hosts a modern Information and Communication Technology (ICT) infrastructure that is managed through a variety of ICT-based committees. Despite this, the current use of any form of e-Learning on campus is limited, and few, if any, mechanisms exist to promote or support its implementation within the wider academic community. It should be noted that while distance education may have exposed the academic staff to technologies and ideas that could more easily have been ported to an e-Learning environment, RU has firmly maintained its view of an on-campus, highly collegial, traditional environment.

The IS Department is officially situated within the Commerce Faculty of RU, Grahamstown, although academic staff members also sit on the University's Board of the Faculty of Science, due to common interests with the Department of Computer Science (CS) within certain degree programmes, such as the BSc(IS) (Rhodes University, 2004). Reflecting this synergy, the IS Department is housed together with the CS Department within the recently completed Hamilton Building. This building, fully occupied in 2002, is a world-class ICT facility providing cutting-edge technical infrastructure to both the staff and learners of the two departments, with a rich wired and wireless networked environment. Consequently, the staffs of the two departments operate in an extremely well-supported, highly technical and enabling environment for the exploration of e-Learning technologies.

3. Motivation for an e-Learning Module

Teles (1999:3) maintains that technologies have always played an important role in the configuration of the learning environment, from the advent of writing systems and materials, followed by the introduction of formal schools and continuing through to the current potential in the use of current computer-based technological systems.

Harasim, Hiltz, Teles and Turoff (1998:271) believe that one of the basic requirements for education this century is to prepare learners for participation in a knowledge-based economy in which knowledge will be the most critical resource for social and economic development. They believe that current education models, structures and approaches will prove to be inadequate. What learners now require is an expanded domain encompassing the broader society, other educational institutions, and in short, access to a wider range of expertise and environments than that provided within a local educational institution.

According to Harasim (Chapter 4 in Mason & Kaye 1989), the key attributes characterising this expanded domain are that it is a largely asynchronous, place independent, many-to-many interactive communication medium. Figure 1 illustrates this view, with the intersections of distance, online and face-to-face clearly demarcated.

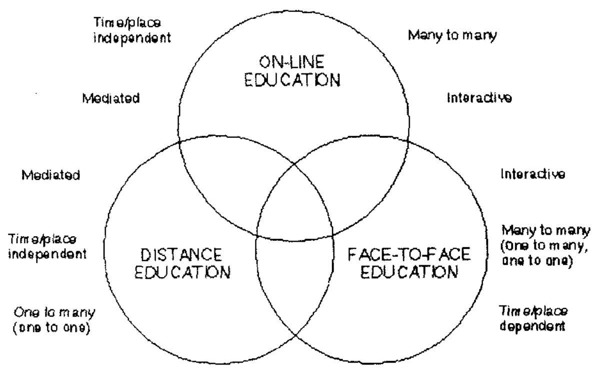


Figure 1 On-line Education as a New Domain (Harasim, 1989: 51)

This view of intersecting relationships within the expanded domain is contrary to the interpretations with respect to distance and on-campus learning provided by the National Working Group on Higher Education (NWGHE) (2001:9-10), in South Africa which prefers to designate higher education institutions in South Africa to be purely distance or purely on-campus, with approval for exceptions to this being considered on application at a programme rather than institutional level. However, although RU has largely abided by the somewhat limited interpretation of the NWGHE, a few individual departments have made some headway in terms of using online education as an adjunct to face-to-face teaching. In terms of being able to initiate, maintain and support independent projects for this purpose, the IS Department has been in a favourable position to adopt an autonomous approach.

Substantiating the shift towards a new educational paradigm that satisfies the learners' needs for new and different information resources, skills, roles and relationships, Harasim *et al.* (1998:27-30) elaborate on what they see to be the characteristics of network learning i.e. expanded educational access, collaborative learning and group work, active learning, learner-centeredness and fluid roles, and online learning communities. This view is supported by Kearsley (2000:4-10), who describes the themes that shape online learning as: collaboration, connectivity, student-centeredness, unboundedness, community, exploration, shared knowledge, multi-sensory experience and authenticity. These online learning themes are integrated with the characteristics identified by Harasim *et al.* (1998:27-30) and are contrasted to traditional classrooms in Table 1.

Theme / Characteristic	Online	Traditional
Collaboration	Promoted by easy online interaction, teamwork and networking are valued	Classroom is a self-contained unit, basis of individualism and competition.
Connectivity and unboundedness / Expanded education access	Fast and unobtrusive contact through email and conferencing – interaction with peers and experts Not place and time dependent.	Larger explicit and implicit barriers between peers and experts. Bounded geographically into "one size fits all" programme.
Student-centeredness	Students largely determine direction through participation. Instructors define goals, then largely facilitate or manage.	More structure provided by instructor, less responsibility for learners.
Community	Virtual communities possible	Community defined by physical location
Exploration	Informal: games Formal: problem based learning Facilitated by access to resources and expertise	Harder to facilitate in the closed environment
Shared knowledge	Via the web – huge resource	Via books, journals – smaller resource: limited by what is at hand locally.
Multi-sensory experience	Interactive media-rich offerings	More static media available
Authenticity	Learning activities can be realistic	Accusation of lack of realism

Table 1: Online learning: Characteristics and Themes.

Simpson (2003) contends that e-Learning that is heavily text based results in poor knowledge retention. She advocates the use of a combination of textual components with pictures, audio and video and interactivity in order to move learners up the Online Learning Continuum as depicted by Van Dam (2003, in Simpson 2003) in Figure 2. Simpson (2003) further states that blending self-paced learning with web-based, instructor-led events promotes interactivity and therefore knowledge retention.

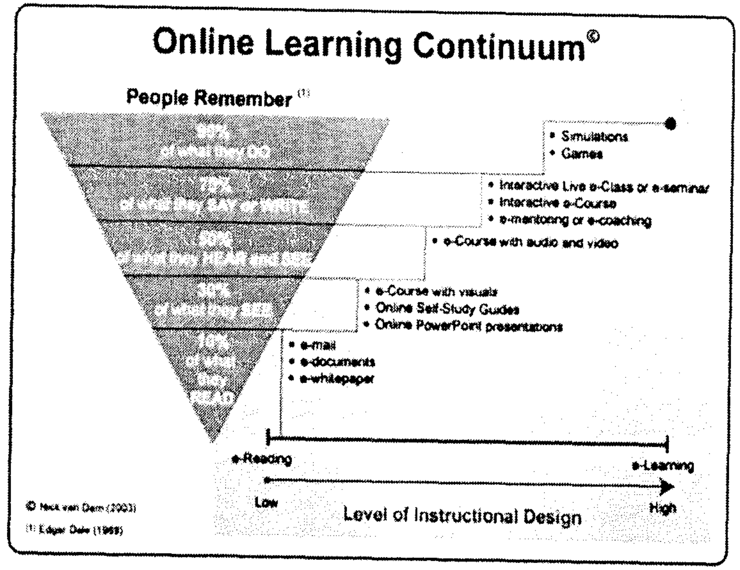


Figure 2: The Online Learning Continuum, Van Dam (2003, in Simpson 2003).

The case for introducing an online component as a teaching and learning mode within the IS Department, despite the University's purely on-campus status, is made both in terms of pedagogical soundness and technological expertise and relevance within the discipline. Given that the current status of the overall environment is unlikely to change, the question is *how* to select and design a hybrid or blend of electronic and traditional methods for module implementation. The feeling of isolation reported by Czerniewicz (2001:17-18), together with a lack of sense of audience, and the challenges of creating an online persona will not play a role in such a blended environment.

Mantyla (2001:3) states that engaging in blended learning means taking two or more presentation and distribution methods, and combining them to enhance both the content and the learning experience. This involves using different types of technologies and software applications, and the resulting blend is a carefully selected and designed mix of electronic and traditional methods. Brodsky (2003) describes proactive blended learning as a well-thought out combination of e-Learning and other training methods, and points out that while the notion of blended learning is not new, its value in the current climate is significant. This view is endorsed by Gery (in Brodsky 2003), who emphasizes that the future trend is to use the concept of blended learning more effectively. Finally, Flanagan (in Brodsky 2003) cautions that reactive blending may be used as a crutch for a poorly designed e-Learning solution, without due regard for consideration of the strengths and weaknesses of technology-mediated learning.

The discipline of Information Systems is described in the IS Department Handbook (2004: 2) as 'the effective analysis, design, construction, delivery, management and use of information and information technology in organisations and society'. Within this framework, e-Learning can be viewed simply as a particular application area for an information system, and may be encompassed by an academic IS Department in the same way that e-Business or e-Commerce have recently been included in many IS curricula. Indeed, the underlying technologies for e-Learning and e-Commerce are not dissimilar.

4. Research Design

Throughout the period of this study, the curriculum development of the e-Learning module within the IS Department was guided by the research literature available in the evolving field of e-Learning tools, technologies and methodologies, and their relevance and application to the teaching and learning environment. In addition to this, each of the subsequent iterations was informed by learner evaluations, facilitator reflection and interviews with the participant observer, all pertaining to the previous iteration.

Baskerville (1999:10) categorised the characteristics of IS Action Research by the process model, the structure, the typical involvement and the primary goals. In this study, a reflective process model was used which was iterative in nature. The structure of the research was fluid, in that activities were defined loosely and in collaboration with the learner group, as well as the same participant observer who was constant throughout the four years of study. Although the author as researcher was involved in determining the interventions, a facilitative rather than expert involvement was aimed for as defined by Baskerville and Wood-Harper (1998: 94-95). The primary goal was the continuous improvement of the e-Learning module curriculum, as well as the goal of individual learning for all participants including the learners, the observers and the facilitator. The outcome of the research cannot necessarily be generalized.

5. The e-Learning Module Curriculum

5.1 Assumptions of Prior Knowledge

The curriculum should be viewed in the context of the assumptions concerning the learners' prior knowledge. The IS students have all had a thorough undergraduate grounding in Information Technology (IT) Project Management, Corporate Communications, and the Systems Development Life Cycle incorporating Planning, Analysis, Design, Implementation, and Support. A systems development project undertaken within their practical groups has afforded the learners some real project development experience. The timing of the e-Learning module at honours level is also such that the post-graduate learners have already been exposed to specialist honours modules in Requirements Elicitation (RE), Human Computer Interaction (HCI), and more recently, Knowledge Management (KM) (Information Systems Department, 2004). From the 2003 iteration of the module, basic HTML (Hyper Text Markup Language) or knowledge of another web development tool such as Microsoft's FrontPage was also assumed, as these had been introduced in an undergraduate e-Business module. In short, these students are ideally prepared to enter, and make a significant contribution to the area of e-Learning.

5.2 The Evolving Curriculum

Due to the blended nature of the module delivery mode, a highly detailed schedule is drawn up well ahead of time for each year of implementation. A sample schedule from 2003 (Mallinson, 2003) provides an illustration of the planning involved and is illustrated in Appendix A. This schedule depicts the mode of each learning intervention, the topic, the presenter or facilitator, a description of each activity (individual or group), the assigned follow up task, and finally the suggested reading or practical work as appropriate. A pack of selected readings is given to the learners at the start of the module. Participants include post-graduate students of the IS and CS departments of RU and the University of Fort Hare (UFH), and guests drawn from the wider academic staff at Rhodes.

5.2.1 First Iteration, 2001

The module was initiated as a short (half module) offering within the Honours course in 2001 (Information Systems Department, 2001), and made available to both IS and CS students. At this stage the module was entitled 'Online Education Systems' (OLES), and included the following topics: Introduction to Online Instructional Technology and Networked Learning, Implementation Platforms, Cost Effectiveness, the Changed Role of the Instructor and Learner, Online Assessment and Evaluation, Technical Considerations, the Design of Instructional Systems, Multimedia and Interactivity, Challenges to the Adoption of Online Learning, and Future Trends.

Software used included WebCT as the base Virtual Learning Environment (VLE), while other VLEs analysed and evaluated during the course included Virtual-U and Blackboard. e-Learning portals of the time that were investigated included eDegree and Educor, both South African based online learning providers.

The initial OLES module had six face-to-face contact sessions scheduled, with the remaining interactions taking place asynchronously online. Learner activities included seminars and discussion, research and presentation via individual and group work, a group Learning Management Systems (LMS) report, and individual essay. The components and features used in WebCT included the email, discussion forums, chat, content and resource management, and course administration. Attempted use of the assignment upload feature was unsuccessful. The module evaluation was implemented as a Small Group Interaction Diagnosis (SGID) via a synchronous chat function, and was conducted by a participant observer (not a student).

5.2.2 Second Iteration, 2002

The second iteration of the module in 2002 saw the inclusion of a further topic of relevance, namely Learning Objects (Information Systems Department, 2002). The functionality, components & features of Integrated Distributed Learning Environments (IDLEs) were discussed in detail. The module was implemented using PowerTutor, a locally developed commercial learning management system (LMS), which was kindly made available to the Department on a trial basis by e-technik, Port Elizabeth. The learners were also introduced to a Computer Assisted Assessment (CAA) tool, Perception, although they were not required to undertake formal practical work using the software. The number of face-to-face contacts remained constant at six, although the module's overall duration (elapsed time) was extended by 50% despite retaining its half-module status. This was due to feedback obtained from the learners in the initial iteration where concern was expressed about the blended nature of the learner interactions taking longer to implement satisfactorily than traditional module delivery.

5.2.3 Third Iteration, 2003

In 2003, the third iteration of the newly extended (full) and renamed e-Learning module took place within the relatively constrained time frame of four weeks (Information Systems Department, 2003). This was due to scheduling constraints within the overall Honours / Fourth year programme. This year also saw the first learner from the UFH CS Honours class participating in the module. A decision was made to physically locate this learner together with the IS Honours group in the Hamilton Building during the period of the module implementation in order to avoid any feeling of isolation or segregation being developed by the learner.

The module topics remained largely the same as before with the inclusion of new issues concerning a valid theoretical foundation for e-Learning, Blended e-Learning, Streaming Media via the WWW, and the Educational applications of Mobile or Wireless computing. All topics covered in 2003 are depicted within the schedule presented in Appendix A. The module was implemented using KEWL (Knowledge Environment for Web-Based Learning), a locally developed open source VLE, which was piloted at Rhodes by the Educational Technology Coordinator of the Academic Development Centre (ADC) in 2003.

Participation in an Online Desktop Lecture Series run by HorizonLive.com was included in the module activities. These 'webinars' stream live e-Learning lectures of interest by experts in the field, free of charge to remote participants using video, audio, slide presentations, application sharing and interactive audio and chat facilities.

The most significant extension within the 2003 curriculum involved a new emphasis on the practical component of the module. Learners were required to undertake group software mini-projects with the aim of producing an e-Learning web presence on a relevant theme. The themes chosen were Online Simulations and Intelligent Tutoring Systems; a screen shot of the resulting web site for the Intelligent Tutoring Systems group mini-project is illustrated in Figure 3.

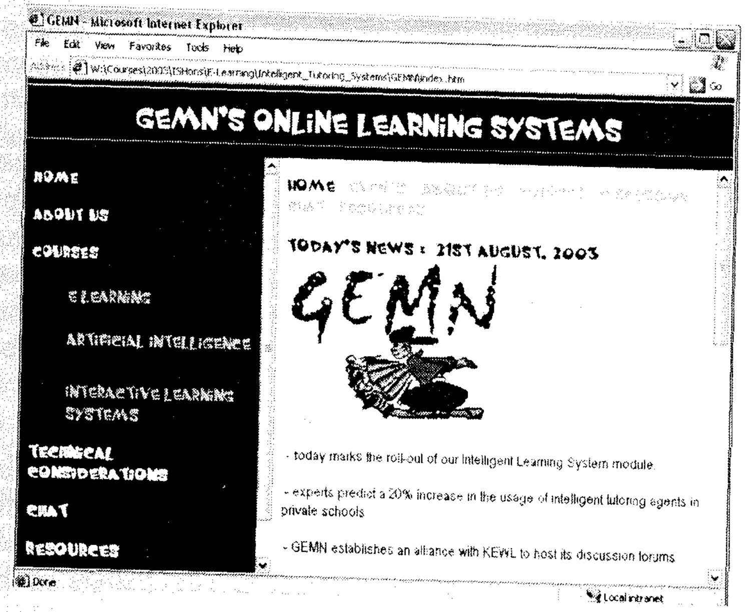


Figure 3: The Intelligent Tutoring Systems Mini-Project Web Site.

In order to facilitate the mini-projects, a variety of software was included in the 2003 module curriculum, including: Perception (CAA tool from QuestionMark), Authorware (a media rich authoring tool from Macromedia), Viewlet Builder (an animated screen capture tool from Qarbon) and Windows Media for streaming via the web. Basic HTML or other web development tools such as Microsoft's FrontPage were not included in the curriculum, although they were available to the learners, and knowledge of such a tool was required for mounting the mini-project.

As the mini-project was a new innovation, the project requirements specifications as well as the assessment criteria were negotiated with the learners with due regard to the restricted timeframe of the activity and the module in general. The project assessment took the place of

50% of the examination mark, thus reducing the summative assessment considerably.

5.2.4 Fourth Iteration, 2004

The fourth iteration of the e-Learning module took place in August 2004 (Information Systems Department, 2004). New additions to topics include blogs (web logs) within the e-Learning context; e-Learning standards; open content and its relationship to open source, intellectual property and copyright; the relationship between Knowledge Management and Enterprise e-Learning; and to include more explicitly, the principles of Instructional Design. The full module was again delivered within a constrained time frame of four weeks.

Feedback from the 2003 learners concerning the practical mini-project informed the 2004 projects in that the project themes drew from the theoretical topics already covered in the module, so as to allow the learners to build on existing knowledge and concentrate on the practical aspect of the project, rather than the content. Selected high quality projects may be used as learning objects for future iterations of the module.

With the increased number of UFH learners participating in 2004, the face-to-face synchronous contact sessions were video-conferenced between the two campuses, rather than brining the learners to Grahamstown for the duration of the module. The online portion of the module was not affected, other than to be enhanced by the presence of genuine distance learners. The direction of the video-conference stream varied with the facilitator moving between Grahamstown and Alice, a distance of approximately 100 km. The Grahamstown learners obtained the experience of being at the remote end of the video-conference for the first time, and *vice versa*, which informed both groups' perceptions of video-conferencing as a suitable tool for education. This involved four visits to the remote campus by the facilitator, and was easily implemented. During these visits, the module facilitator took the opportunity to spend some time assisting the UFH learners with practical work pertaining to the module, and generally providing

guidance and support to the remote learners. On one occasion, the module facilitator was accompanied by a teaching assistant and on another occasion by two RU learners, which promoted increased interaction between the geographically dispersed module participants.

5.2.5 Overall Curriculum View

In the American Society for Training and Development's (ASTD) Learning Circuit Glossary (2001), e-Learning is defined as 'covering a wide set of applications and processes, such as Web-based learning, computer-based learning, virtual classrooms, and digital collaboration. It includes the delivery of content via Internet, intranet/extranet (LAN/WAN), audio- and videotape, satellite broadcast, interactive TV, CD-ROM, and more'. Allen (2003:27) uses the all-inclusive definition of 'a structured, purposeful use of electronic systems or computers in support of the learning process'.

The IS Department's e-Learning module may now be considered to very nearly encompass a full spectrum of e-Learning tools and technologies as described in the previous sections and shown in Appendix A. It should be noted that in this module, the context is important. The fact that the learners are IS or CS graduates rather than Education students, informs the curriculum and focus of the module. The important contributions that education issues make in the e-Learning discipline are not ignored; they are simply not the focus of the module within the presented context. Table 2 illustrates the evolution of the e-Learning module curriculum.

Year	2001	2002	2003	2004
Extent	Half Module	Half Module	Full Module	Full Module
Time Period	2 weeks	3 weeks	4 weeks	4 weeks
Activities	Lectures: 3 Learner research & presentation: 2 Online discussion; Reading;	Lectures: 3 Learner research & presentation: 2 Online discussion; Reading;	Lectures: 4 Learner research & presentation: 1 Online discussion; Reading; Practicals: 4 Project: 1 mini-project	Lectures: 4 Learner research & presentation: 1 Online discussion; Reading; Practicals: 4 Project: 1 mini-project
Assessment: Formative	* Individual researched presentation; * Group Report & presentation; * Individual Essay; * Individual Online Contribution to Discussion Forums;	* Group researched presentation; * Group Report & presentation; * Individual Essay; * Individual Online Contribution to Discussion Forums;	* Group researched presentation; * Individual Essay; * Individual Online Contribution to Discussion Forums; * Group Practical Project * Individual Proj. Reflect.	* Group researched presentation; * Individual Essay; * Individual Online Contribution to Discussion Forums; * Group Practical Project * Individual Proj. Reflect.
Summative	* Written Examination.	* Written Examination.	* Written Examination.	* Written Examination
Presentation style / mode	Blended: 6 contact sessions; the remaining interactions online.	Blended: 7 contact sessions; the remaining interactions online.	Blended: 10 contact sessions; the remaining interactions online.	Blended: 10 contact sessions; the remaining interactions online.
Software Tools	WebCT; HTML	PowerTutor; HTML; Perception.	KEWL; HTML; Authorware; Perception; Viewlet Builder; Windows Media; HorizonLive.	Moodle; HTML; Authorware; Perception; Viewlet Builder; Windows Media; HorizonLive.

Table 2: The Evolution of the E-Learning module curriculum

6. Analysis

6.1 Learner Evaluation

Learner evaluations of the module highlighted the following issues:

- Time period over which the module was delivered. The learners reported problems in terms of accomplishing the required activities and tasks within the prescribed time frame. They believed that this was due to the blended mode of the module. For example, discussion threads within an online forum take some time to develop and involve reading and reflection by the learner prior to making their own contribution.
- Information Systems Context. The learners believed that the module was relevant to the discipline of IS. In fact, they felt that as IS students they should have been exposed earlier to the new mode of online teaching and learning.
- Areas of specific enjoyment of the course include the online interaction, and the blend of mode and media which affords the learners added flexibility of their engagement with the module.
- Drawing on their prior knowledge of HCI and systems design, learners were critical of the VLEs used for the module. This occurred in each year, even though a different software solution was used for each of the four module iterations.
- In the 2003 evaluation, dissatisfaction was expressed in terms of the time pressure for the extended practical work, even though the learners expressed enjoyment when undertaking the mini-projects. The learners did not whole-heartedly engage with the negotiated specifications and assessment exercise for the project. The time constraints within which they were working meant that they did not have time to undertake this exercise in as considered and reflective manner as was desired. The learners also suggested that an integrated approach to the theory and practical components would be more conducive to efficient time management and retained learning.

6.2 Facilitator Reflections

- The facilitator's journal and observations over this period have highlighted the following:
- Technical matters. A problem concerning power cuts which coincided with the start of the first iteration of the module in 2001. The Information Systems Department were not yet situated in the Hamilton Building for this iteration, and were subject to the general problems relating to external lecture venues that were not necessarily within the Department's control. At that stage, the Honours' laboratory was also physically external to the IS Department. The move to the Hamilton building in 2002 proved to be of immense value to the network stability and security of the technology supporting the module.
- Assumptions of prior knowledge. The e-Learning module is also open to CS graduates who have not necessarily co-majored in IS. Although these students were found to be lacking in some of the areas mentioned in Section 4.1, they were not excluded from any aspect of the module. However, no special accommodation was made for these learners, and they were expected to acquire skills as and when needed within the e-Learning module. The intrinsic collaborative and supportive environment within the e-Learning module is of assistance to these learners.
- The learners were intrigued by the online interaction, and the notion of synchronous *versus* asynchronous communication is hotly debated each year. The variety of media continues to be extended in order to explore the full range of potential technologies available within the e-Learning spectrum.
- The issue of the extended time period over which the module should be held is highly important. While this concern was acted upon when planning the 2002 iteration of the module, a shift which was appreciated by the learners, the module reverted to a regular time 4-week frame in 2003 and 2004 which is far from ideal. Renewed effort was made by the facilitator to insist that the module is sched-

uled over a longer period of time in order to do justice to the e-Learning module, resulting in a scheduled 5-week period for the 2005 module. Harasim *et al.* (1998:223) report that the amount of time spent in online classes exceed equivalent face-to-face classes. They attribute this partially to the fact that the depth as well as scope of online discussion far exceeds what is possible in face-to-face classroom situations.

- The initial theoretical topics were sound, and a good start was made in terms of initiating learners' awareness of the issues concerning OLES. Additions to the evolving e-Learning module curriculum have reflected trends in the discipline.
- The introduction of extended practical work in 2003 required careful revision. In 2004, a framework for the mini-project specification was provided to the learners early on in the module, for revision if they so wished. Clear boundaries on the nature and extent of the practical work should be set by the learners in collaboration with the facilitator, in order to contain the volume of the project work.
- The VLE used for the 2004 iteration is Moodle, an open source learning environment. Potential exists for the customisation of the software by the learners and post-graduate researchers, with funds for a development server having been secured for 2005. The existing stable version housed within the IT Division at Rhodes will host the module. This software was piloted by the facilitator in an undergraduate IT Project Management module early in 2004 (Mallinson, 2004b), and many of the prospective participants in the e-Learning module have now used Moodle as learners or tutors within the INF301 course.
- Learner interactions have been good in terms of asynchronous discussion, with almost all of the learners contributing satisfactorily, and indeed exceeding the requested output in terms of quality and quantity. The high level of peer activity and the de-centralizing of the facilitator's role for the discussion forums were similar to that described by Harasim *et al.* (1998: 74). This interaction was observed to have been carried over to the Honours laboratory. Learners assisted each other with mini-project and other work, and the environment was generally reported to supportive, irrespective of the facilitator's presence, on- or off-line.

- The module has been enhanced each year by the presence of guests who are Rhodes academic staff members from outside the Department, and who have requested to attend the entire module as participant observers. These observers have added value to the module, and acted as role models in terms of academic staff keen to implement e-Learning within their own disciplines.
- The e-Learning research group, established in 2002, and coordinated by the e-Learning module facilitator, has supported the module via invited guest presentations where appropriate. These post-graduate researchers have motivated the learners and acted as role models for further research and interest in the discipline of e-Learning.

7. Concluding Remarks

The e-Learning module has been an exciting venture for the facilitator concerned. The prior knowledge of the learners has played an important role in the feasibility of mounting such a densely packed learning experience within a restricted time frame as part of the IS Honours / Fourth year programme. The skills, knowledge and expertise of an IS graduate leads one to suggest that they uniquely poised to provide the link or interface between the Educational Specialist and the Computer Programmer in an e-Learning development team.

e-Learning technologies have been established within the IS Department as an application area worthy of enquiry and examination. Ideally, the module should take place over an extended period of time in order to accommodate the blended nature of the module implementation.

The introduction of new technologies and their relevance with the pedagogical framework of the module has added to the enjoyment of the module for both facilitator and learners alike. Technical support has been vital to the success of the module implementation; and the high level of expertise and support available within the Hamilton Building has been much appreciated.

Further iterations of the e-Learning module will continue to be informed by the technological and pedagogical trends in this rapidly evolving discipline, while a concerned eye will be focussed on the developing theoretical foundations underpinning e-Learning methodologies and practise.

References

Allen, MW 2003. *Michael Allen's Guide to e-Learning.* London: John Wiley & Sons, Inc.

Baskerville, RL 1999. Investigating Information Systems with Action Research. *Communications of the Association for Information Systems (AIS).* Volume 2, Article 19.

Baskerville, R & AT Wood-Harper 1998. Diversity in Information Systems Action Research Methods. *European Journal of Information Systems* 7, 90-107.

Brodsky, MW 2003. Designing E-learning from the Outside-In. *e-Learning Magazine.* Accessed on 3 March 2003

http://elearningmag.com/elearning/article/articleDetail.jsp?id=48117

Czerniewicz, L 2001. Reflections on learning Online – the hype and the Reality. SAJHE/SATHO Volume 15 No 3.

Harasim, L. 1989. Online Education: A New Domain. Chapter 4. In *Mindweave: Communication, Computers and Distance Education.* Robin Mason & Anthony Kaye (eds.) Oxford: Pergamon Press.

Harasim, L, R Hiltz, L Teles, & M Turoff 1998. *Learning Networks: A Field Guide to Teaching and Learning Online.* Cambridge, Massachusetts: MIT Press.

Information Systems Department 2001. *Information Systems Handbook 2001.* Rhodes University.

Information Systems Department 2002. *Information Systems Handbook 2002.* Rhodes University.

Information Systems Department 2003. *Information Systems Handbook 2003.* Rhodes University.

Information Systems Department 2004. *Information Systems Handbook 2004.* Rhodes University. Accessed on 29 May 2004 http://www.is.ru.ac.za

Kearsley, G 2000. Online Education: Learning and Teaching in Cyberspace. Wadsworth: Thomson Learning, Inc.

Learning Circuit Glossary 2001. Learning Circuit Magazine, *American Society for Training and Development (ASTD).* Accessed on 29 May 2004 http://www.learningcircuits.org/glossary.

Mallinson, BJ 2003. e-Learning Honours Module Knowledge Area Specification Document, Internal Document, Department of Information Systems, Rhodes University.

Mallinson, BJ 2004a. e-Learning Honours Module Outcomes Based Education Document in *Information Systems Quality Assurance Manual,* Internal Document, Department of Information Systems, Rhodes University.

Mallinson, BJ 2004b. Information Technology Project Management Outcomes Based Education Document in *Information Systems Quality Assurance Manual*, Internal Document, Department of Information Systems, Rhodes University.

Mantyla, K 2001. *Blending E-LEARNING: The Power Is in the Mix.* American Society for Training and Development (ASTD).

National Working Group on Higher Education (NWGHE) 2001. The Restructuring of the Higher Education System in South Africa. Ministry of Education, South Africa. Accessed on 29 May 2004

http://education.pwv.gov.za/content/documents/250.pdf.

Rhodes University 2004, Rhodes University Calendar Centenary Edication 2004. Rhodes University.

Simpson, J 2003. Move aside e-Learning ROI: What's your e-Learning Return on Knowledge Retention? nGRAIN Corporation. May 2003. Accessed on 29 May 2004

http://www.ngrain.com/news/images/e_learning.pdf.

Teles, L (ed.) 1999. The Online Classroom: Design, Delivery and Evaluation – Selected Readings. Vancouver: TELEStraining Publishing.

Van Dam, N 2003. *The E-Learning Fieldbook,* McGraw-Hill Professional.

Authors' Contact Details

BJ Mallinson (B.Mallinson@ru.ac.za)

DA Sewry (D.Sewry@ru.ac.za)

Department of Information Systems,

Rhodes University, Grahamstown, South Africa

The Influence of the ECT Act on a Typical Computing Curriculum

Andries Barnard

Laurette Pretorius

Lucas Venter

Abstract

Recent technological advances have led to the emergence of new technologies, frameworks and methodologies in the field of computing, the regulation of which are governed by scientific principles. Examples include the Internet, global connectivity and mobile agent technology, collectively referred to as cyberspace. Regulating human interaction with cyberspace has become one of the great challenges of the information age – a challenge in which legislation plays a central role.

At present the regulation of human interaction with cyberspace by means of legislation, is awarded prominence on a global level by governments. In this regard the computing instructor plays a central and multi-faceted role. It is within this context that we consider what influence the new South African Electronic Communications and Transactions (ECT) Act (Act No. 25, 2002) may have on a typical computing curriculum.

Keywords

South Africa, ECT Act, Computing Curriculum

1. Introduction

Advances during the past decade have led to the emergence of, among others, new technologies, frameworks and methodologies in the field of computing. Examples include the Internet, global connectivity and mobile agent technology, collectively referred to as cyberspace. Regulating cyberspace has become one of the great challenges of the information age, a challenge in which the computing professional plays a central and multi-faceted role. Therefore, computing professionals have to, in fulfilling their role in society, take into consideration the fact that cyberspace is populated by both humans and artificial entities.

At present the regulation of cyberspace by means of legislation is awarded prominence on a global level by governments. It is within this context that we consider what impact the new South African Electronic Communications and Transactions (ECT) Act (Act No. 25, 2002) could have on a typical computing curriculum. In section 2 we give a brief overview of the Act and discuss some shortcomings. More specifically, we note that the Act deals in particular with Internet related activities. These activities range from providing a mere description of data messages, to complex measures (including various security considerations) relating to and attempting to realise the governance of e-commerce transactions in South Africa. Section 3 contains a discussion of the impact of the Act on three subject areas mentioned in the ACM/IEEE computing curriculum (Computing Curricula ..., 2001). Although the discussion is not exhaustive, it demonstrates how the Act could impact on other aspects of this curriculum.

2. The Electronic Communications and Transactions Act, 25 of 2002

The ECT Act is considered to be of great strategic importance for the whole continent of Africa, as is argued in (Van der Merwe, Pretorius and Barnard, 2004). Because South Africa has been one of the first African countries to adopt this type of legislation, it is possible

that the Act might serve as a model for the entire Sub-Saharan African region. The purpose of the Act, as stated in its preamble (Act No. 25; 2002), is:

- *"to provide for the facilitation and regulation of electronic communications and transactions;*
- *to provide for development of a national e-strategy for the Republic;*
- *to promote universal access to electronic communications and transactions and the use of electronic transactions by SMMEs (Small, Medium and Micro Enterprises);*
- *to provide for human resource development in electronic transactions;*
- *to prevent abuse of information systems;*
- *to encourage the use of e-government services; and*
- *to provide for matters connected therewith."*

In addressing this purpose, the Act covers, amongst others, the following topics (dealt with in individual chapters of the Act):

- Electronic transactions: chapter III;
- E-government services: chapter IV;
- Cryptography providers: chapter V;
- Authentication service providers: chapter VI;
- Consumer protection: chapter VII;
- Protection of personal information: chapter VIII ;
- Protection of critical databases: chapter IX;
- Domain name authority and administration: chapter X;
- Limitation of liability of service providers: chapter XI;
- Cyber inspectors: chapter XII;
- Cyber crime: chapter XIII.

Sections 13 to 20 of chapter III that deal with electronic transactions have overcome one of the biggest obstacles regarding electronic commerce without paper embodiments, namely the question of proof of the contents of electronic documents. By affording data messages admissibility as evidence, the court bypasses the paper trail that has characterised especially civil litigation in the past. A data message is defined in section 1 of the Act as: *"data generated, sent, received or stored by electronic means, and includes a) voice, where the voice is used in an automated transaction and b) a stored record"*. Van der Merwe *et al.* (2004) note that it remains to be seen, however, how much weight a judge or magistrate will afford a document that is only available in electronic format. Section 15(4) provides that a data message created *"in the ordinary course of business"* presents rebuttable proof of its contents, thus transferring the onus to produce evidence of a lack of reliability on the party aggrieved by the admission of the electronic document.

Furthermore, section 20 of the Act should have implications for the use of artificial entities. In particular, chapter I, section 1 of the Act defines an electronic agent as *"a computer program or an electronic or other automated means used independently to initiate an action or respond to data messages or performances in whole or in part in an automated transaction"*. This section provides for an agreement, or *contract* in a commercial sense, to be created by an artificial entity, in this case an electronic agent. This extends the class of entities that can be involved in legal situation.

The recent importance attached to the actions of artificial entities (software agents) in cyberspace (Barnard, Cloete and Pretorius, 2004), raises the question of what would happen if both parties made extensive use of what the Act refers to as electronic agents. In such an event much of the consumer protection built into chapter III, section 20 seems to fall away because natural persons are reduced to peripherals while the electronic agents (code, architecture, software agent, etc.) are responsible for effecting a transaction (Van der Merwe *et al.*, 2004).

Furthermore, the South African Police Services already have a Computer Crime Unit. Buys and Cronjé (2004) argue that the force of cyber inspectors envisioned by the Act, is in conflict with this existing law enforcement agency. They state that the appointment of a force of

cyber inspectors *"that have wide-ranging, invasive powers, yet do not have the experience and additional resources of the SAPS, is an unwarranted extension of the powers of the Department of Communications"* (Buys and Cronjé, 2004:335-336). However, many of the officials required to carry out important functions in terms of the Act still have to be appointed, and the necessary infrastructure must be put in place. For example, chapter IV of the Act provides for an Authentication Service Provider that will include an Accreditation Authority, both of whom are necessary to certify the authenticity of an advanced electronic signature. Similarly, the cyber inspectors provided for in Chapter XII have not yet been appointed. This points to a potential shortcoming of the current provisions of the Act.

Lessig (1999) describes four modalities that impact on the regulation of modern society, viz. norms, law, market forces and computer code (architecture). He argues that in the absence of strong influences by the first three modalities to regulate modern society, increasing prominence may be awarded to the use of computer code (architecture) to effect such regulation. Hence, when the Department of Communications put these measures in place and appoint the relevant officials, the Department would have to avoid possible conflict with existing law enforcement agencies, such as the SAPS Computer Crime Unit. The alternative may well lead to a greater dependence on computer code (architecture) for regulation. In the above example, the guarantee of authenticity may simply be left to computer code (architecture). This may well lead to increased dominance by certain software developers.

An initiative in providing the necessary infrastructure to enforce this legislation was reported on January 12, 2004 (SABC News, 2004). In particular, the *"US Secret Service has donated 22 specialised forensic computers to help combat cyber crime in this country. They are valued at more than a million rand"*. Training is envisioned to take place in two phases. Firstly members of the SAPS Computer Crime Unit will be trained, followed by legal practitioners including judges and prosecutors.

In concluding this section on the ECT Act we briefly note the first conviction in terms of this act. Two perpetrators, Michael Bafatakis and Andrew Stokes, gained illegal access to a South African mobile telecommunications operator, Vodacom, and downloaded personal information of Vodacom clients. They attempted to blackmail Vodacom for an amount of ten million rand not to disclose this information. The conviction (November 2003) of the two perpetrators was the first in terms of this new act (Sake Rapport, 2003), regarding the contravention of sections 86 and 87 of the Act.

3. A Typical Computing Curriculum

For the purposes of this paper we use the proposed ACM/IEEE Computing Curricula (Computing Curricula ..., 2001) as an example of a typical computing curriculum. Both the IEEE and ACM endorsed the curricula recommendation of this report. Many textbooks currently in use at universities worldwide, and certainly at UNISA, are based on the guidelines given in this report. Hence it is clear that this report has a large impact, directly and indirectly, on curricula at many universities. A good example of this is the recent trend to include sections on social and professional issues (Computing Curricula …, 2001: Subject area 12) in many computing textbooks.

This body of knowledge is organised into fourteen disciplinary subfields or subject areas. We consider the impact of the Act on a selected number of subject areas of this curriculum. In the context of the previously stated purpose of the Act and the structure of the CS body of knowledge (Computing Curricula ..., 2001) we are of the opinion that the teaching of the following three subject areas may, and should, take cognisance of the contents of the Act:

- Net-Centric Computing,
- Information Management, and
- Social and Professional Issues.

The reader is referred to Appendix A of (Computing Curricula …, 2001) for more information concerning these subject areas.

3.1 Net-Centric Computing

In the above-mentioned appendix the contents of this area are summarised as follows: *"Recent advances in computer and telecommunications networking, particularly those based on TCP/IP, have increased the importance of networking technologies in the computing discipline. Net-centric computing covers a range of sub-specialties including: computer communication network concepts and protocols, multimedia systems, Web standards and technologies, network security, wireless and mobile computing, and distributed systems."*

In this section, we restrict our discussion to three specific topics of the Act that deal with information security. Chapter III of the act makes provision for the facilitation of electronic transactions. In particular, it defines an electronic document and specifies how to deal with these documents. Chapters V and VI of the Act deal with cryptography and authentication service providers, and chapter XII makes provision for the appointment of Cyber Inspectors. We show how these chapters impact on the teaching of Information Security. Of special interest are the following topics covered in subject area NC3: Network security (Computing Curricula …, 2001):

- Fundamentals of cryptography
- Secret-key algorithms
- Public-key algorithms
- Authentication protocols
- Digital signatures.

3.1.1 Electronic Transactions and Documents

The act in effect affords a data message, discussed in section 2 above, the same status as any other legal document, if it complies with two requirements. The document must be *signed* with an advanced electronic signature, and it must be retained (stored) in such a way that (see section 16 of the Act):

- the information contained in the document is accessible;

- it can be demonstrated that the stored message accurately represents the information contained in the original document;
- the origin and destination, as well as the date and time sent and received, can be determined.

The advanced electronic signature mentioned by the Act is simply a standard electronic signature, which is issued by a registered authentication service provider. The signature should be uniquely linked to a specific user who must be identified on a face-to-face basis, must be capable of identifying that user, and must be created by means under the sole control of that user. The signature should also be linked to a specific data message and should be capable of detecting any subsequent changes to the contents of the message. These requirements are fairly standard, and can to a large extent be provided by using hash functions and public key encryption techniques, such as described in any standard textbook, for example (Pfleeger 1997: 96-98). This can be illustrated by learning objective 4 of topic NC3 (Network security), which states *"Summarize common authentication protocols."* (Computing Curricula ..., 2001).

A problem arises when the storage of data messages is considered. The Act does not specify that the message must be stored in its original format; rather, the Act specifies that the information contained in the document must be accurately represented. Current authentication techniques can only ensure the authenticity of a document if it is stored in its original format (Pfleeger 1997: 97). Research in this area might lead to an advanced degree.

3.1.2 Cryptography and Authentication Service Providers

The Act requires that all providers (not users) of encryption services or products must be registered. It defines an encryption product as any (software) product that contains any form of encryption. This implies for instance that Linux, which is freeware, is classified as an encryption product. Hence versions of Linux distributed through download sites become illegal. Commercial providers of products containing Linux would have to register as a cryptography provider, even though their product might not be available in South Africa, but is simply used by some person in this country. This subtlety of the Act would have to be emphasised in courses where any aspect of cryptography is taught.

Courses on cryptography would also have to deal with ethical issues. For instance, a cryptography provider can be required by law to disclose all information regarding services provided to a specific client. The provider would have to consider ethical issues when deciding how much information he/she will store. This issue is related to SP7: Privacy and civil liberties, mentioned in section 3.3 of this paper.

The act also creates the possibility to register authentication service providers, whose main task will be to provide advanced electronic signatures for data document authentication. This registration is not compulsory, as in the case of cryptography providers. The reason for this might be that most known effective authentication techniques are based on cryptography, and hence that these providers will have to register as such. Typically, these services will be provided by computing professionals and therefore these issues need to be adequately addressed in a computing curriculum.

3.1.3 Cyber Inspectors

The activities of cryptography and authentication service providers will be monitored by Cyber Inspectors. These inspectors will also have the authority to monitor and inspect any web site or activity on an information system in the public domain. They would have to have the skill to determine whether a cryptographic or authentication service is provided. Once this is established, they must also have the ability to determine whether the service rendered is in compliance with the provisions of the Act. Highly specialized training programs would have to be developed to train these inspectors.

3.1.4 Information Management

In Appendix A of (Computing Curricula ..., 2001) the contents of this area are summarised as follows: *"Information Management (IM) plays a critical role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation, and presentation of information; algorithms for efficient and effective access and updating of stored information, data modeling and abstraction, and physical file storage techniques. It also encompasses information security, privacy, integrity, and protection in a shared environment."*

Chapter VII of the Act deals with consumer protection. In particular, section 43 deals with information to be provided by service and goods providers making use of electronic means. This information to be provided to potential customers include, amongst others:

- Code of conduct to which the supplier subscribes (section 43(1)(e)).
- Any terms of agreement, including any guarantees, that will apply to the transaction, and how those terms may be accessed, stored and reproduced electronically by consumers (section 43(1)(k)).
- Security procedures and privacy policies of that supplier in respect of payment, payment information and personal information (section 43(1)(p)).

It is therefore important that the computing instructor should equip the computing student with the technical knowledge and skills to adhere to the requirements of these and other stipulations of this chapter of the Act. This is in accordance with learning objective 4 of topic IM1 (Information models and systems) that reads: *"Describe several technical solutions to the problems related to information privacy, integrity, security, and preservation"* (Computing Curricula ..., 2001).

Chapter VIII of the Act deals with personal information and privacy protection. Section 51 describes the principles for electronically re-

questing, collecting, collating, processing and storing of personal information. Learning objective 3 of topic IM1 (Information models and systems) reads *"[c]ritique/defend a small- to medium-size information application with regard to its satisfying real user information needs"* (Computing Curricula ..., 2001).

The Act defines critical data as *"information which, if compromised, may pose a risk to the Republic's national security or the economic or social well being of its citizens"* (Act No. 25, 2002, chapter 1; Deloitte and Touche, 2002). Chapter IX of the Act makes provision for the protection of critical databases. In particular, sections 52 through 55 of the Act deal with the scope of protection, identification, registration, and management of critical data and databases and will account for the protection of critical data. Buys and Cronjé (2004:173) states that the Act contains universally accepted data protection principles setting out how personal information may be collected and used, and note that selective subscription to the Act is not permissible. We therefore consider the adequate coverage of the following topics suggested by (Computing Curricula ..., 2001) to be of crucial importance:

- IM1: Information models and systems;
- IM2: Database systems;
- IM3: Data modelling;
- IM4: Relational databases;
- IM6: Relational database design;
- IM7: Transaction processing;
- IM8: Distributed databases;
- IM9: Physical database design; and
- IM11: Information storage and retrieval.

We also propose that the study of Chapter IX of the Act, as well as its interpretation, and an associated assessment should be included in the study material for and teaching of information management (IM).

We further note that there are issues in this section of the Act that have close links to certain personal privacy issues that are also reflected in the subject area Social and Professional issues of (Computing Curricula …, 2001). These similarities are echoed by Buys and Cronjé (2004, p.172) who state that *"[d]ata protection can factually be regarded as forming part of information privacy"*.

3.2 Social and Professional Issues

Appendix A of (Computing Curricula …, 2001) justifies the contents of this area as follows: *"Undergraduates also need to understand the basic cultural, social, legal, and ethical issues inherent in the discipline of computing. They should understand where the discipline has been, where it is, and where it is heading. They should also understand their individual roles in this process, as well as appreciate the philosophical questions, technical problems, and aesthetic values that play an important part in the development of the discipline. … [S]tudents need to be aware of the basic legal rights of software and hardware vendors and users, and they also need to appreciate the ethical values that are the basis for those rights. Future practitioners must understand the responsibility that they will bear, and the possible consequences of failure."* (Tucker as quoted in (Computing Curricula …, 2001))

It is therefore clear that the computing professional should acquire knowledge and skills regarding the social and ethical issues in computing. A closer look at the subject area, Social and Professional Issues (SP) of (Computing Curricula …, 2001), reveals that the following topics may be linked to the contents of the law in general and/or the ECT Act in particular:

- SP2: Social context of computing;
- SP4: Professional and ethical responsibilities;
- SP5: Risks and liabilities of computer-based systems;
- SP6: Intellectual property;
- SP7: Privacy and civil liberties, and
- SP8: Computer crime.

If we agree that our professional and ethical responsibilities also include the legal systems in place in society, then SP4 should address those laws that regulate various aspects of cyberspace and the computing profession. In the South African case, the ECT Act will play a significant role in regulating conduct and activities in the field of computing, particularly with regard to the Internet. The computing instructor must therefore include pertinent aspects of the Act within this framework.

Chapter XI of the Act deals with the limitation of liability of service providers, that is, persons who provide information system services. It concerns service providers who act as mere conduits, perform caching (automatic creation of temporary copies of electronic data for quick access) or hosting (renting of space to users who provide their own content). Topic SP5 makes provision for including such material in the computing curriculum.

Chapter VII deals with consumer protection, unsolicited electronic communications (spam) and consumer protection for online card payments, topics that are suitable for inclusion into material regarding SP7: Privacy and civil liberties. For example, learning objective 2 of SP7 illustrates this: *"Describe computer-based threats to privacy."*

Chapter VIII, sections 50 (Scope and protection of personal information) and 51 (Principals for electronically collecting personal information) address the protection of personal information. Computing professionals involved in any application that involve so-called electronic transactions should be aware of this chapter of the Act, and also of the fact that the accurate definition of *"electronic transaction"* is still awaited. *"One of the most urgent issues to address, either by Parliament or the courts, will be the definition of electronic transaction."* (Buys and Cronjé, 2004:160).

While making provision for cyber inspectors, who have the power to *"inspect, search and seize"* (section 82), the final chapter (XIII) of the Act deals with cyber crime. Section 86 concerns unauthorised access to, interception of or interference with data (including hacking, cracking and denial-of-service), section 87 addresses computer-related extortion, fraud and forgery and section 88 refers to attempt, and aiding and abetting cyber crime. Topic SP8 of the typical curriculum under consideration makes provision for including issues concerning computer crime. Future computing professionals should take special cognisance of the fact that, according to (Buys and Cronjé, 2004:323):

- a computer can be involved in computer crime as the *object* of the crime, e.g. as victim where unauthorised access and denial-of-service attack is launched against it;
- a computer can be used as *instrument* (tool) in computer crime; and
- a computer can be incidental to an offence, e.g. where it is used to store records of illegal activities.

This is in accordance with learning objective 1 that reads *"Outline the technical basis of viruses and denial-of-service attacks."*

4. Conclusion

"Although technical issues are obviously central to any computing curriculum, they do not by themselves constitute a complete educational program in the field. Students must also develop an understanding of the social and professional context in which computing is done." (Computing Curricula ..., 2001).

This quote supports the following concluding perspective:

We considered the impact of the ECT Act on a typical computing curriculum as proposed by the ACM/IEEE. We have seen that in order to address the various aspects of the Act we need to incorporate the fundamental subject areas of this curriculum into our educational programs. The Act, in fact, provides a social context within which we can teach these fundamental aspects.

The ECT Act is but one way in which society expresses its social needs with regard to the regulation of cyberspace. There are also other ways for society to articulate such needs, for example, cultural, normative, moral etc. Each of these impacts on the context within which we may teach the fundamental subject areas of our typical curriculum in its own way. We contend that by teaching universal technical subject content within a localized context the computing professional of the future will be better equipped to face the increasing challenges of the information age.

References

Act No. 25 (2002), The South African Electronic Communications and Transactions Act.

Barnard, A, E Cloete & L Pretorius 2004. A Framework for Performing Security and Ethical Analyses in Agent Computing. *Challenges for the Citizen of the Information Society: Proceedings of the seventh International ETHICOMP conference.* Syros, Greece. In TW Bynum, N Pouloudi, S Rogerson & T Spyrou (eds.) pp. 81 - 93.

Buys, R & F Cronjé 2004. Cyberlaw@SAII. *The Law of the Internet in South Africa*. Pretoria: Van Schaik.

Computing Curricula 2001 – Final Draft (December 15, 2001). http://www.computer.org/education/cc2001/final/index.htm, accessed on 30.04.2004.

Deloitte and Touche Legal 2002. Synopsis of the electronic Communications & Transactions Act

http://www.saica.co.za/documents/Synopsis.pdf, accessed on 30-04-2004.

Lessig, L 1999. *Code: and other laws of cyberspace*. New York: Basic Books.

Pfleeger, CP 1997. *Security in Computing*. Second edition. Upper Saddle River, NJ: Prentice-Hall.

SABC News 2004. US Secret Service Donates 'Crime Fighting' Computers http://www.sabcnews.com/world/north_america/0,2172,71878,00.html accessed on 13.01.2004.

Sake Rapport 2003. Krakers Ry aan die Pen, November 23, pp. 7.

Van der Merwe, DP, L Pretorius & A Barnard 2004. Cyberethics and the South African Electronic Communications and Transactions Act; *Challenges for the Citizen of the Information Society: Proceedings of the Seventh International ETHICOMP Conference*; Syros, Greece. In TW Bynum, N Pouloudi, S Rogerson & T Spyrou (eds.) pp. 860 - 870.

Authors' Contact Details

Andries Barnard (**barnaa@unisa.ac.za**)

Laurette Pretorius (**pretol@unisa.ac.za**)

Lucas Venter (**ventelml@unisa.ac.za**)

School of Computing
University of South Africa, Pretoria, South Africa

Just Because You Can, It Doesn't Mean that You Should!

Peter L Mkhize
Sam Lubbe

Abstract

Web entrepreneurship is an area for people who run a variety of online markets of which some succeed and some fail. Examples of the latter are the enterprise heavenly-doors.com, which operated a funeral service over the internet and others that provided retailing services that failed because they did not pay attention to user demands. Entrepreneurs who have technological capabilities started businesses to be convenient and effective to the consumer. However, a significant number of such businesses closed their virtual doors as web traders since 2000.The authors conducted a study to examine the examples of business plans from a businesses plan archive website, to establish if there were common mistakes made by web entrepreneurs. This enabled the authors to determine where web entrepreneurs were short-sighted and to report on possible solutions to be kept in mind by entrepreneurs presently operating e-Learning websites.

Keywords

Computer Based Training, Distance Learning Systems, DotCom, e-Learning, Learning Management Systems, m-Learning, u-Learning

Introduction

Internet, when used properly, is an effective marketing tool. Web services makes life convenient for people with heavy work schedules by simple logging in to the site and utilising the services provided by the service providers and commercial entrepreneurs. The services provided by websites are helpful to working individuals well of as to non-working people e.g. shopping online, advertising second hand furniture, e-Learning etc. On the other hand, an entrepreneur with access to technological capabilities was assured booming business as their client numbers increased. However, too many of the web entrepreneurs are out of business now. Miller (2002) stated that there are a number of failures among the web business. (In 2000, about 835 Internet companies went bust.) This article will therefore investigate some of the more spectacular failures and the reason why they did in order to apply it to e-Learning to ensure successful transfer of knowledge.

Background to Problems Experienced by Web Entrepreneurs

The range of failed websites that went broke offer a lesson to all would-be e-Learner facilitators. It can be noted as: just because you can do it doesn't mean that you should do. The fact that the technological capability is there does not mean that anyone wants the service or product offered. And the fact that you can do it certainly doesn't mean that there is any money in doing it!

Some of these websites that failed in 2000 were creative in the use of the technology, but they really did not have much hope of success as the question of who would want to use them and what advantage there would be for the user was not at all clear. Take for example the case of Heavenly-Door.com a website set up to supply funeral services. After five months and $26 million they closed the business. Now the question is and presumably always was "Who in the state of bereavement would want to go to a web site to arrange a funeral?" Just reflect on the last funeral you went to and you will remember that a funeral is an intensely personal event, which cannot be arranged over a telephone line. The funeral is all to do with the overt expression of sympathy of people and thus just cannot be simulated on a computer. The add-ons for e-Learning have not been determined but thus will have an impact (e.g. the same could be stated of the student whose

complaints might need a personal touch). On the other hand not only does the website need to offer something which is truly needed and wanted, but there has to be a reasonable profit margin in it for the e-Business. The action would be to build synergy between the teaching institution and learner activities – the question is: Do we?

Urbanfetch.co.uk illustrated this problem perfectly. Here was an attempt to give the consumer a really fantastic deal by ignoring the realities of transport costs. But most companies just can't magic away the carriage cost and they have to pay a delivery charge. The delivery charge levied is sometimes not that large, with the vendor partly subsidising this cost out of the profits of the sale. But someone has to pay for the carriage of the goods. Transport is a key element in the economic equation and unless it is properly integrated into the cost model the business will not survive. So Urbanfetch.co.uk lasted about nine months before having to close its doors.

The list of e-Business failures is long and there is no doubt that it will continue to lengthen until web entrepreneurs begin to understand the issue of the business model and start to take it seriously. Institutions that offer e-Learning can be counted on one hand in KwaZulu-Natal and no doubt some of them will fail. In 2000 most web entrepreneurs simply estimated the market size and then said to themselves and their suppliers of funds, that they can easily get 10% or 5% or 1% of that market. This calculation produced an enormous potential figure. They seldom did enough in-depth analysis to understand how to obtain this business and what the cost of obtaining it would be. This analysis is essential to underpin any serious business model, but this is seldom done. Many institutions did not do proper in-depth analysis of in what stage of the learning process they are. (see the conclusion).

In fact a professionally produced business model needs to address two main issues. These are that: A successful website need to have a compelling and preferably unique reason why people should want to come to the site to buy or engage in the activity that e-Learning for example offers and also the website needs to be able to charge a fee which will cover the business's costs and may be make a profit for its investors. If these two questions cannot be confidently answered then perhaps the e-Business will not be a success.

In trying to answer the first part of this demanding question it is essential that the would-be web entrepreneurs take advice from potential users of the website. It's not good enough for the would-be web entrepreneurs to simply imagine the reaction of potential clients. This was one of the issues that brought Boo.Com down. The same could be used for e-Learning; if the course is offered listen to the users.

With regards to the second part of the model there is no point in proceeding with a business for which there is not an adequate number of paying clients. And these clients need to be reachable at not too great a cost. e-Stamp.com gave up and closed their doors because although there was a good market for postage stamps in the USA, it was costing them about $600 to get a client to sign up and this was just prohibitive. Again this is a difficult question to which to find a satisfactory answer because course development can be expensive.

A very interesting example of another case that was not thought out fully is that of Priceline.com. Being a web equivalent of a travel bucket shop which has its doors open 24/7/365 undoubtedly has merit and it is quite probable that with enough throughput or volume of sales a good business can be made of that proposition. However, it is easy to copy this business model, especially if you are one of the airlines being represented by Priceline.com. Thus this was an intrinsically vulnerable model. Another problem surfaced with the notion of reverse auctioning. It is perhaps the excitement of being able to do new and interesting things with the technology that deflects their attention from the business realities with which they must come to terms if they are to become a success.

Here the traveller states how much he or she will pay for the flight and the website tries to find a carrier that will be prepared to offer passage at that price. Such travellers normally offer quite a low price. Of course this deeply discounted approach leaves little margin for the air-

line, not to mention the intermediary and it doesn't suit all that many people to make their travel arrangements this way. But when Priceline.com tried to extend the notion of reverse auctioning to groceries and gasoline the model came under intense strain. Anyone with any knowledge of the retail grocery business knows that making e-Retailing work is very difficult. The fulfilment cost is the spanner in the works. So if it is so difficult to make this work when the groceries are being sold at the normal price, how can a deeply discounted reverse auctioning strategy work for this industry sector? When Jay Walker, the then CEO of Priceline.com said that he intended to re- engineer the DNA of business, he was indeed aiming high, and many sincerely wished him great success. However, it appears that he really had misunderstood the nature of the problems he was addressing. On top of these challenges, a group of the airlines that Priceline.com had been representing started their own discounted website. The vulnerability of the Priceline.com business model was proven beyond any doubt and it began to crumble. What is really fascinating about Priceline.com is that at its height of popularity the stock market valued the company's market capitalisation at more than the combined value of all the American registered airlines flying at that time. Today Priceline.com is worth about 1% of its former value. Priceline.com's business model could not stand up to reality. The question is how will re-engineering affect the DNA of e-Learning education because e-Learning is also expensive (as stated in the next section).

There is of course nothing new about creating a business model to ensure that the business idea will 'fly'. It is as old as the idea of business itself. But, for some reason it seems to be overlooked or forgotten when would-be web entrepreneurs begin to set up their operation. The bottom line reality is that it is difficult to make a success of an e-Business. Specifically e-Business is complex, e-Business is not inexpensive, and e-Business requires a collection of resources and skills or competencies. Understanding disadvantages associated with e-Learning one would ensure that academics are full, prepared and aware of the challenges involved in the development and design of e-

Learning.

Disadvantages of e-Learning

Developing and maintaining instructional models is intriguing but problematic at the same time because of the often daunting amount of training necessary, the potentially confusing variety of hardware and software choice available, and the dizzying pace at which technology evolves (Ries, 2004).

The National Education Association (NEA) (2000-2001) issued cautions on its web site in terms of portfolio assessment. There could be potential hurdles in moving from print to the digital medium of learning. Even though data in digital form can be easily cross referenced, overlaid and analysed, if you want to take advantage of technological tools you should consider several factors before changing from a traditional system. These are:

- *Access.* The hardware and software used to capture and store the student portfolio must be accessible to both parties. If computers, scanner, and printers are still down the hall in the computer lab, this may not be the time to initiate an e-Learning portfolio.
- *High-end Tools.* Depending on the subject matter, data may be stored in multiple data sources (text, voice, video, image, etc). The capacity to store more than a single file format will give a well-rounded representation of the student's work. Therefore, you will need access to at least one high-end workstation with a scanner, OCR software, printer and perhaps digital camera.
- *Space.* Graphics and photographic images take up a more system storage than text does. Be sure that the university's system can support large files without compromising other applications.
- *Labour.* Accumulating information for an electronic portfolio is labour-intensive and time-consuming.
- *Administration.* Before starting, determine how you will administer the electronic portfolio. A database application will be required that establishes an area for each student, stores various file formats and allow annotated comments.

The above-mentioned cautions as issued by the NEA give way to the analysis of the current position of the institution that intend to implement e-Learning. As must as web entrepreneurship had promised a good niche market for those who had resources, e-Learning demands large investment on infrastructure to address the above mentioned disadvantages. Tertiary institutions have yet to understand the history of e-Learning and evaluate their phase at which they are at, before fully committing to the implementation for e-Learning.

Barriers that Might Prevent Proper e-Learning

The prospects of e-Learning growth in both developed and developing countries is threatened by the barriers that are exerted by internal and external forces. The internal barriers are intrinsic factors that can be controlled by the course provider; whilst the external barriers are extrinsic factors that a more difficult to control.

Jones, *et al.* (2003) suggested the subcategories of intrinsic and extrinsic factors namely personal circumstances (PC), course related factors CRF) and student perception (SP), which they deduced from nine reasons for student withdrawal in online courses. These nine reasons are:

- Prime causes
  - Lack of time
  - Job or Business Changed/increasing pressure of work
  - Nature of the course
  - Personal Issues
  - Amount of coursework
  - Technical problems
  - IT skills
  - Did not require further qualification
  - Confusion/lack of understanding

- Other causes
  - Inflexibility in course design
  - Withdrawn by College

The above mentioned cause of withdrawal can be positioned in any of the subcategories of the barriers of e-Learning. Personal circumstances category is an extrinsic factor and is external to the programme such as change of employment and family and health related problem, these variable are more difficult to influence and control. Course related factors and student perceptions are intrinsic or internal factors; course related factors are related to a course component such as technical problems, assessment, reliability of the Virtual Learning Environment, course structure and type and quantity of assessment. Student perception group identifies problems related to the student understanding of the course and how it could benefit them.

The History of e-Learning

The evolution of e-Learning has been revolving around the above mentioned cautions, which assisted in the improvement of e-Learning. Cross and Hamilton (2002) discussed and identified e-Learning evolution in era's, then furnish the lesson learned in a specific era.

1990-1999: The Era of Custom Computer Based Training (CBT)

This era concentrate on computer based training (CBT) which meant using CD-ROM courses playing on the student standalone computers, standalone training stations, and sometimes across client/server LANs.

There were many technology players in this era because it was easy to copy this model (developed desktop-based multimedia authoring systems that could modestly skilled programmers to develop CBT). Like it is presently, digital media tended to be a rich mixture of video, narrative audio as well as music and sound effects, graphics and animations, and formatted text.

The resulting courses were innovative, highly participatory, engaging, and instructionally effective, often conditionally branching simulations, opportunities for learner exploration and discovery, extensive feedback and user-controlled videos, audiovisual slideshows, and so on. However, they were expensive, slow to develop, and monolithic and again easy to copy.

The lesson: Well-designed, high-quality CBT can have instructional and performance-boosting value. However, it is costly and labour intensive to develop, quickly to become obsolete, and suffers from a house-on-cards software limitation. All these prevent CBT's value from gaining long-term or large-scale momentum because there were little competitive advantages for institutions.

1997-1999: The Rise of the Learning Management System (LMS)

Though CBT was a useful tool, the users of the system realised they had significant number of high cost, high promise CD-ROM courses going to all sort of prospective users. But the usefulness and impact of CD-ROM was not known, it was problematic especially when CBT is to be implemented on a large scale (expensive). CBT had to be managed centrally and justified the cost. It was stagnant and was not clear what they want to achieve and how people will reach their goals.

A supplementary feature found in many custom courses, a LAN based student administration and data reporting system, portended a solution to the problem. The projected solution was a more expensive WAN-based or intranet/Web-based version of CBT that works across extended educational institutions.

This system could:

- Automate the administration of CD-ROM-based and even Web-based training deployed across many locations.
- Launch and track CBT courses.
- Work both intra and inter-departmentally.

- ➢ Report on the results of everything, and stratify reporting by location, department, group, etc.
- ➢ Surround and enrich CBT experiences with online collaboration among groups of learners and between instructors and learners, such as threaded discussions, chat rooms, news and document postings.

Thus Computer-Managed Instruction (CMI) also known as Course Management System (CMS) was born. However, the purpose of these CMI/CMS systems became blended with Training Management Systems (TMS).

The Training Management Systems tend to emphasise:

- ➢ Modelling of employee skills and measurement of skill gaps through online testing.
- ➢ Correlation of skill-deficient learners with matching training solutions.
- ➢ Administration of classroom training courses and logistics.
- ➢ Automation of the registration process.
- ➢ Of course, reporting on the results of everything.

The merger of CMI/CMS and TMS resulted in a new breed of Learning Management Systems (LMS), which featured robust technologies and a comprehensive attempt to administer, manage, track, and report on skills, classroom training. It was essential to pay attention in putting strict standards into place, to ensure the CBT portion of LMS adhere to interoperability standards between CBT and LMS.

The lesson: Enterprise control of CBT administration and deployment is good. But it needs strict standards so that different content sources readily plug-and-play on any administrative technology platform.

1999: Everyone Moves to the Web, or at Least Want to

Although companies and institutions have installed Internet and configured their networks it should be noted that a broad bandwidth ensures a commonplace for people to use. It made sense to migrate from CD-ROM-based training to Web-based training. Cross and Hamilton (2002) further mention drivers for this transition:

- ➢ Web-based training helped to justify the cost of the intranet.
- ➢ Implementation of e-Learning to the field became far easier when CD-ROMs weren't being distributed and maintained at endless numbers of locations.
- ➢ Learning could be taken "anywhere, anytime", as long as a browser-based Web connection could be made to the host server.
- ➢ The shelf life of courseware could more easily be extended, as course updates needed to be implemented only once, on a server, rather than endlessly on each training workstations at each training location.
- ➢ Central LMS management and control via easy-to-establish Web connections allowed the promise of the enterprise LMS to reach fruition in pragmatic terms, and catalysed a true empowerment of the central Training/HR Department to manage training across the enterprise.

All species of e-Learning companies (custom courseware developers, packaged content providers, and LMS vendors) moved to Web technology as quick as possible. It had been realised by their companies that e-Learning held potential. This was not possible until something along the lines of the interactive, instructional, and media richness of CD-ROMs could be duplicated and reinvented using Web standards.

The Lesson: The Web is where it's at because it brings administrators, instructors, managers, and the worker-learner together in a consolidated virtual environment. Worthwhile e-Learning on the Web would be challenging to accomplish although reverse engineering does not always ensure sinful learning.

Mid 1999 to 2000: The Internet Land Grab Is on

In this era enterprise deployment of web courseware with central administrative management grew, but neither high quality nor customisable and manageable content was to be seen. Many e-Learning vendors of this era, those that were and that just started up, offered a shopping mart of centrally managed Web content. This was accessible anywhere, anytime and high quantity of content was intended to make up for compromised quality. The technology was portable and the content philosophy was offered to mass quantities.

LMS became the data management backbone of all learning portals, solidifying the perceived fundamental importance of an LMS to making e-Learning happen. For the majority of original high-quality CD-ROM developers the jump to the Web was not successful, the reasons includes their unwillingness to lower quality standards to what was current on the Web, lack of aptitude for the production of mass quantities of content, and general shortage of sufficient technological sophistication to create enterprise-strength solution for Web-server delivery. However, companies that were providing low-quality learning shut down, others quickly sold their venture. The survivors won the right to another round by reinvesting themselves.

The lesson: Opportunities in the new Web economy are staked out by learning portals and acres of bland learning content. Mass aggregations of e-Learning content not designed to serve specific business purposes and offering little instructional quality have no value.

Conclusion

e-Learning is one of more the intensively researched subjects and has been very attractive to tertiary institutions. However, e-Learning has promised a lot in the field of education and training. e-Learning programme developers and users therefore have to be cautious when implementing e-Learning. The lessons learnt by the DotCom companies should be used to develop a proper instructional design model. Miller (2002) also mentioned top ten lessons from the DotCom melt-

down that could be used. However the expectations about e-Learning should calm down as there are still issues that need to be sorted out before proper e-Learning implementation can take place. These are:

1. Nothing changes overnight.
2. New stuff doesn't replace old stuff.
3. Too early to market? Too bad.
4. Many startups were fundamentally uncreative and un-Internet.
5. We all, like sheep, will go astray (with enough pressure).
6. Free is folly.
7. We used narrowcast to broadcast.
8. The R50 million rule can kill.
9. It's difficult to build chicken and egg simultaneously.
10. Prediction tools must improve.

These above mentioned lessons can be used to ensure that e-Learning programmes are sustainable. Through-out the years electronic learning material have been developed and improved upon, e-Learning history or past, present and the future can be illustrated by phases of e-Learning development. These phases provide the trends of the development and improvement of e-Learning, there are lesson learnt in the process; as illustrated in Figure 1.

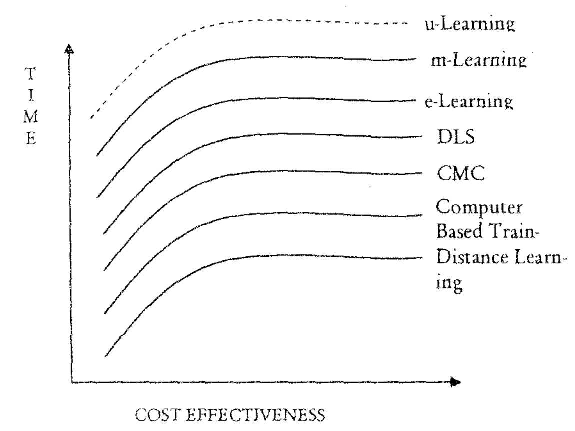


Figure 1: e-Learning phases of development

The figure above graphically displays what the authors envisage would happen in the educational landscape based on the theory presented in this article. The model above predicts that electronic modes of delivery would be cost-effective and that some modes of electronic delivery would be more cost-effective than others. However, it should be kept in mind that the actual predictive power of this model should still be researched in a follow-up study.

References

Cross, J & I Hamilton 2002. The DNA of e-Learning, Excerpt from *Beyond e-Learning*. Internet Time Group www.internettime.com accessed 14 November 2003.

Jones, P, G Packham & C Miller 2004. E-learning for Enterprise Studies: The Case of E-College Wales. *International Journal of Applied Entrepreneurship* Vol. 1, No 1, Spring, ISSN: 1742-5824.

Miller, T 2002. For Starters: Top Ten Lessons from the Dot Com Meltdown

http://www.businessplanarchive.org/whatwecanlearn/tenlessons.htm Accessed 7 January 2004.

National Education Association. 2001. Technology and Portfolio Assessment http://www.eportconsortium.org/old/whatis.aspx accessed 7 January 2004.

Ries, R 2004. *Electronic Learning Portfolio*. Bolton, Massachusetts: Anker Publishing Company.

Authors' Contact Details

Peter L Mkhize (mkhizepl@ukzn.ac.za)
Sam Lubbe (slubbe@ukzn.ac.za)
School of Information Systems & Technology
University of KwaZulu-Natal, Durban, South Africa

An Analysis of Learning Difficulties Experienced by First Time Spreadsheet Users

Tanja Reinhardt
Nelishia Pillay

Abstract

Spreadsheets have become a routine application in most organizations and universities. Consequently, students are required to learn spreadsheet applications such as Microsoft Excel. The learning of spreadsheets is often accompanied by problems related to spreadsheet applications and their mathematical content. The EXITS (Excel Intelligent Tutoring System) research project aims to develop a Microsoft Excel tutor that helps students to overcome their learning difficulties. The paper reports on the first phase of our EXITS research project. We firstly identify barriers that may prevent a student from successfully using a spreadsheet program. We then analyse the different types of errors students make and identify their causes. We also classify the errors. This classification will form the basis for an error library that is required for our student model. At the end of the paper we give an example of how our model will be used for student modelling purposes.

1. Introduction

The main task of a spreadsheet program is to analyse and manipulate numbers. For example, numbers can be entered into cells and mathematical calculations can be performed to solve almost any kind of numeric problem. It also allows the user to represent the numerical data in a more accessible form as charts. Although spreadsheets have been around for about 20 years, novice and expert spreadsheet users still have difficulties when using spreadsheets (Tukiainen 2000: 247; Brown 1987: 258).

*Alter*nation 12.1b (2005) 429-453 *ISSN 1023-1757*

When users start working with spreadsheet programs, they have to cope with two challenges. One is to master the application program and the other is to understand the mathematical concept of spreadsheets. We have found that computer literacy students experience both spreadsheet and mathematical learning difficulties when using a spreadsheet program for the first time. Unfortunately, the majority of available spreadsheet tutorials focus only on spreadsheet concepts, (e.g., how to use a SUM function or how to create a chart), but do not provide mathematical assistance. Thus, these packages will not be able to assist students in performing more complex spreadsheet calculations or overcoming learning difficulties associated with mathematical and logical thinking. Each student has individual needs and problems. Therefore our aim is to develop a software program to provide students with individualised tuition to overcome these problems. We have decided to develop an intelligent tutoring system (ITS), namely EXITS (Excel Intelligent Tutoring System), to provide personal instruction and error-specific feedback for a variety of problems a student may experience.

This paper reports on the first phase of our EXITS research project. The aim of this phase was twofold. For student modelling purposes, barriers that may prevent a student from successfully using a spreadsheet program were identified. Secondly, a survey of the different types of problems experienced by students was conducted. This information will be used to guide the error feedback for EXITS.

2. An Overview of Student Modelling As Part of an Intelligent Tutoring System

Intelligent tutoring systems are software programs that provide individualized tutoring or instructions. They have the ability to present the teaching material in an interactive environment and provide personal instruction and error specific feedback (Mandridou 2003: 157). Compared to other computer-based training techniques ITS systems assess each learner's actions within these interactive

environments and develop a model of their knowledge, skills and expertise. Based on the learner model, the ITS system aims to simulate a human tutor who can adapt his or her instructional strategies in terms of both the content and the style, and provide explanations, hints, examples, demonstrations, and practice problem- solving as needed. The widespread use of ITSs has been impeded by the high costs associated with developing these systems (Mizoguchi 2000: 107).

An intelligent tutoring system usually consists of four interconnected core modules: A domain/ expert module, which defines the expert knowledge in an area and the problem-solving characteristics (Serengul 1998). The pedagogy/-tutoring module contains teaching strategies and teaching instructions. The communication module provides an interface, which controls the interaction among student, teacher and knowledge, and the student module captures the student's knowledge, which is used to guide the feedback and information presented (Baffes 1996: 403). In order to provide individualized tuition to students, one of the main tasks of an ITS is to construct a student model, which is used to guide the feedback and information in response to the student's performance.

A number of different student modelling strategies have been developed over the years. One of the approaches is the perturbation student model (Burton 1982: 157). This model assumes that the student's knowledge is a subset of the expert knowledge and that this student knowledge may contain sets of errors or misconceptions, which must also be represented. To build such a model the system requires a library of mistakes (bug library) students are likely to make. The aim of this is to grow the student's subset of the expert knowledge while eliminating the errors and misconceptions in the student's knowledge.

This paper is addressing the student modelling aspects by identifying factors that influence student performance negatively and analyse errors made by students.

3. Course Background and Experimental Methodology

The spreadsheet course is part of a Computer Literacy course that is held every semester at the University of KwaZulu-Natal. The spreadsheet section usually consists of about 12 spreadsheet lectures, one 1½ -hours and two 3-hour practical sessions. The software package used to teach the students is Microsoft Excel 2000.

A questionnaire was administered to Computer Literacy students at the University of KwaZulu-Natal during a Microsoft Excel practical session to collect data for the student module. We used a practical to hand out our questionnaire because it allowed the students to respond to problems experienced, something which cannot be achieved in a test environment.

In addition of formatting tasks, a multiplication formula, the SUM, AVERAGE, MIN, MAX and COUNT function, and a percentage calculation were included. Functions like SUM, AVERAGE etc. are predefined functions that Microsoft Excel uses to perform calculations by using specific values, called arguments, in a particular order or structure. For example, the SUM function adds together values or ranges of cells, and AVERAGE finds the average of a group of numbers. Both the multiplication formula and the percentage calculation require the use of cell references, which refer to a place in a worksheet where Microsoft Excel picks up the value. Brown (1987: 258) showed that 70% of the reported errors were errors in formulae. From our experience we believed that the students had more problems in dealing with formulae using cell references and functions than with simple formatting aspects.

The aim of the questionnaire was to get information about the students' backgrounds as well as information about problems experienced during the practical. With the analysis of the questionnaire we tried to identify the barriers that may prevent a student from successfully using spreadsheets.

To provide individualized feedback on the errors made by the student, we needed to find a methodology to assess these errors. Our

aim was to identify the problems students experience and to look at the kind of errors the student made without the influence of a human tutor, which would obviously affect the result. This study was therefore done during a spreadsheet test with the purpose of using the outcome as a basis to provide necessary error feedback for the students and to create our error library. Although we are mainly concentrating on errors in formulae and functions in this paper, we will also handle other errors in our student model, for example, errors when creating a chart or sorting data.

Not all the students that attended the test attended the practical and vice versa. Therefore the size of our sample groups varied. A total of 151 students attended the practical and returned the questionnaire and a total of 165 students attended the test. 121 students participated in both test groups.

4. Analysis of Learning Problems and Identification of Problem Groups for Student Modelling Purposes

4.1. Student Background

In order to identify barriers and the relationship between student background and performance, we investigated the background of each student to see if the problems experienced by the students might be a lack of computer knowledge (e.g. that students may have difficulties at all to use a computer) or a lack of mathematical knowledge (e.g. that students struggle when dealing with mathematical calculations in spreadsheets). We investigated whether a correlation exists between the understanding of spreadsheets and the mathematical knowledge as well as computer skills. We also included the language aspect in our survey.

A report on problems in Mathematics education at the former University of Durban-Westville discussed the language issue (Paras 2001:66). They stated that students were not sufficiently exposed to the mathematical concepts in English and most of them struggle when they come to university because the instructions are given in English.

When second-language learners do not understand the English instructions they immediately switch to mother tongue. From interviews with Computer Literacy tutors we also found that if a tutor explained something to one student, this student will explain it to the other students in his home language. Research at the University of Port Elizabeth (South Africa) also showed, that apart from study habits and Mathematics pre-knowledge, the English language proficiency can be used to predict academic performance of students in subjects that require the ability of mathematical thinking (Koch 2001: 138). When looking at the language aspect we didn't test the English language proficiency because it would have been too time consuming for the students to fill in the questionnaire and to do an English language proficiency test. Thus we used the student's first language as an indicator.

Figure 1 shows that the background of our survey group was very mixed in terms of their first language, Mathematics level and computer experience. The majority of the students (53%) had English as their first language followed by Zulu (20%) and Setswana (9%) (Figure 1(a)).

We also asked the students about computer experience before they started the Computer Literacy course. The computer skills of the test group (Figure 1(b)) was very mixed and ranged from students with "Computer Science in matric (school-leaving exam), including programming experience" to students with "no computer experience before the course". About 10% of the students stated that they worked with Microsoft Excel before and 48% of the students had no experience with a computer before they started the course. We will refer to the latter group as "novice users" in this survey.

The breakdown of the mathematical experience is shown in Figure 1(c). 43% had higher-grade 12 Mathematics and 29% had standard grade 12 Mathematics. The mathematical education of 17% of the students was below standard-grade 12 Mathematics.

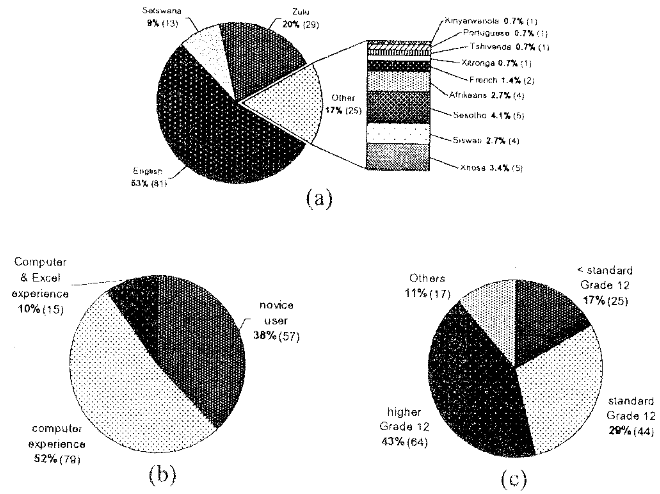


Figure 1: Background of survey group. (a) First language (N=151), (b) Mathematics level (N=150) and (c) Computer experience (N=151).

4.2. Analysis of Problems vs. Student Background

The objective was to determine whether an association exists between the number of problems a student had when doing the practical and their mathematical education, their computer experience and their first language. In order to test these hypothesis we used the Pearson chi-square ($\chi 2$) test, which is often used as a significance test to determine if there is a dependence of two qualitative variables.

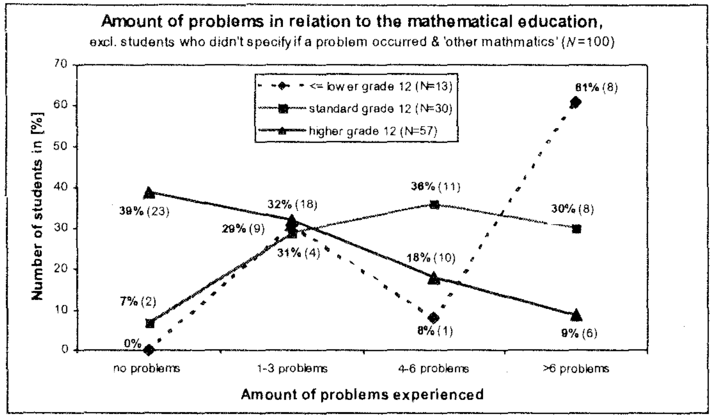
Table 1: Number of problems in relation to mathematical education, first language and computer experience

N	Hypothesis	X^2	df	P	V
100	Mathematical education	30.3	6	P<0.01	0.71
112	First language	16.9	3	P<0.01	0.39
100	Mathematical education/ First language	48.5	15	P<0.01	0.31
112	Computer experience	7.8	3	P<0.055	0.07

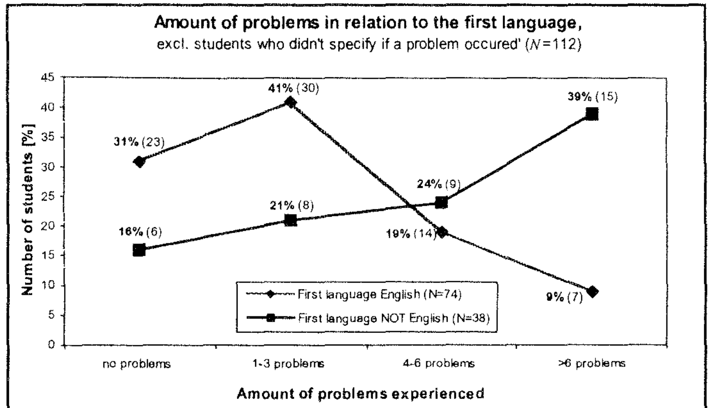
Table 1 shows the preliminary analysis of the demographics with the following results: the mathematical education is strongly associated (V=0.71) with the number of problems experienced with Microsoft Excel. This is also displayed in Figure 2(a), which illustrates the number of problems in relation to the mathematical education. 57 students considered had higher grade 12 Mathematics, 30 students had standard grade 12 maths and 17 had lower grade 12 Mathematics or lower. If we look at the different groups we can see that from the 57 students with higher-grade 12 Mathematics most of them (39%) had no problems with the practical and only 9% of this group had more than 6 problems with the practical. If we compare the group with lower grade Mathematics and below lower grade Mathematics, none of the students in the group had "no problems" and the majority of this group (61%) had more than 6 problems. In general, we observed that students with higher mathematical qualifications had fewer problems doing the Microsoft Excel practical than students with a low level mathematical background. This is what we expected, because spreadsheets in general are mathematically oriented programs and therefore students with a poor mathematics background may find it difficult to form equations and understand the concept of spreadsheet functions. This finding is also reflected in Figure 3 which shows that most of the problems occur when students were requested to write an equation.

Table 1 illustrates that there is a moderate association (V=0.39) between first language English and the number of problems experienced with Microsoft Excel, which is also displayed in Figure 2(b). In this figure, we can see that the number of students with more than 6 problems is larger for students with the first language not being English (39% of the first language not English group) than for student with first language English (9% of the first language English group). Also the number of students with fewer than 4 problems is proportionally higher for students with the first language English (72%) compared to 37% observed for students with the first language not being English.

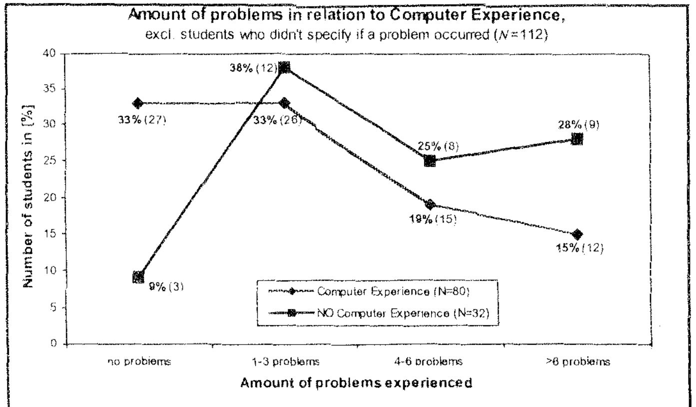
However, there is a correlation between the students' mathematical ability and language proficiency as indicated by a V of 0.31. Thus, there could possibly be an indirect correlation between success in the spreadsheet course and language proficiency due to the methodology employed to teach Mathematics at second language schools. Alternatively, language proficiency could have a direct impact on the students' success in Mathematics.



(a)



(b)



(c)

Figure 2: Number of problems in relation to (a) Mathematical education, (b) First language and (c) Computer experience

Students with no computer experience appear to have had more problems than those students that have previously used a computer. It seems that there is a correlation between computer experience and the number of problems experienced. But if we look at the calculated correlation factor in table 1 we can deduce that there is only a very weak association (V=0.07) between computer experience and the number of problems students' experience. This association is not as strong as the language and mathematical correlations.

4.3. Discussion on Problem Groups

The statistical tests indicated that a significant relationship exists between mathematical education and the number of problems a student experienced. We observed that students with a lower level of mathematical education experienced more problems during their practical than students with a more extensive mathematical background. The questionnaire also revealed a significant relationship between the language proficiency and the number of problems experienced. Students whose mother tongue was not English experienced more problems during their practical session. The relationship between a computer experienced user and the number of

problems is on the other hand not as significant as the mathematical background and the first language. Based on this survey we can now identify two barriers that may prevent a student from successfully using a spreadsheet program. One is the mathematical literacy, and the other is the first language.

Mathematics is used as an indicator of both the student's mathematical knowledge and problem-solving ability. If the mathematical ability of a student is low and he/ she cannot perform calculations, that student will also experience problems performing calculations in Microsoft Excel. If the problem solving ability of students is low we can predict that those students will experience difficulties with logical and complex problems like sorting, if statements, financial functions, and absolute cell addressing. They can be expected to experience fewer problems with simple functions like the SUM function, or simple calculations.

5. Analysis of Spreadsheet Errors

For our model it was necessary to identify both the errors students make when they solve a problem and the causes of these errors. The analysis of the causes will help us to generate the error library and to guide the feedback in our model. Once we know what the problems are we will be able to provide individual instruction and error-specific feedback.

Most of the error classifications described in the literature only provided a general overview of the errors made and thus lack the details needed for our error library (Panko 1997: 21; Tukiainen 2000: 247).

5.4. Errors Using Formulae and Functions

To reduce complexity and to structure our error library we tried to find out if students frequently made similar errors in different formulae and functions.

We looked at the different types of errors students made, and we can make some generalisations (Figure 3, Table 2 in Appendix A). We classified three error categories:

1. Errors that apply to any functions and formulae. For example, errors where students were not able to write down a solution at all (GE1).
2. Errors that occur in formulae and functions with a similar syntax. A lot of functions (predefined formulae that perform specific calculations), like the SUM or AVERAGE function use the same syntax (=function_name(argument)) to perform a calculation. Errors that occurred in any simple function were errors related to the argument (GE4).
3. Errors that only occur in specific formulae or functions. For example if students added individual cells instead of using the SUM function (SE1).

We defined errors that belong to category 1 and 2 as general errors and those errors that belong to category 3 as specific errors. Table 2 (in Appendix A) gives an overview of the different error types with examples of errors student made. 'GE' indicates a general error and 'SE' a specific error.

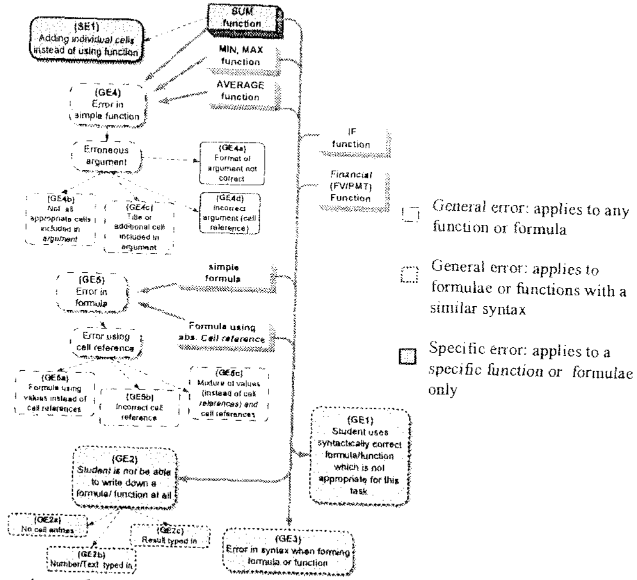


Figure 3: Overview of general errors (GE) and specific (SE) errors using the SUM function

5.5. Classification of Error Categories in Relation to Student Knowledge

To provide error- specific feedback and assistance for our ITS it was necessary to identify the causes of these errors.

If we look at all the errors made when students had to use formulae or functions, we can identify two major causes. Firstly, students made errors because they lack the knowledge of Microsoft Excel concepts. For example, students didn't know how to form a correct argument in a function. Also students often didn't understand some of the concepts like the use of absolute cell references and relative cell references. In our study, we refer to these errors as "spreadsheet errors". For these errors, the assistance will concentrate on the spreadsheet concepts or the syntax of formulae and functions.

Other errors were related to mathematical problems (e.g., the students where not able to set up an equation to calculate a percentage). This also includes logical errors, such as, the use of wrong units when using a financial function. Here we need to assist the student to understand the mathematical concept of that task. For some errors we are not able to distinguish whether the error is due to the lack of spreadsheet knowledge or due to the lack of mathematical/ logical understanding.

If we look, for instance, at errors where students didn't enter anything into the cell (GE2a), we cannot decide whether the student is not able to construct a formula with cell addresses (which would be a "spreadsheet error") or whether the student does not know how to perform the calculation at all (which would be a "mathematics error"). Therefore our feedback needs to cater for both aspects, to assist the student with the mathematical feedback and to assist the student with spreadsheet knowledge.

As mentioned earlier we identified two barriers, a mathematical and a language barrier. So far we have concentrated only on the mathematical/ spreadsheet side and didn't take the language aspect into consideration. From our experience as Computer Literacy tutors we identified some errors that can be related to language problems. For example, if somebody enters a number, which is not the calculated result, as an answer into the cell, there are basically two possibilities as to why the student failed to solve the problem. One possibility is that the student doesn't know how to perform the required calculation, the other is that he/she doesn't know how to apply the required spreadsheet concept. But another possibility is that the student didn't understand or misinterpreted the problem that he was asked to solve due to language problems.

5.6. Error Statistics

An analysis of the distribution of spreadsheet and mathematics errors in formulae and functions revealed that the majority of errors in the SUM function are related to spreadsheet problems (Figure 4a). From the type of errors the student made we can deduce that, although they made errors, they understand the mathematical concept of adding

numbers. 64% of the errors students made were due to the fact that they added single cell references (SE1) and 7% of the students typed the result of the sum calculation into the cell (GE2c). Therefore, we can assume that most of the students who made an error using the SUM function didn't understand the concept of the SUM function.

The errors made when using the AVERAGE function were also mostly (67%) spreadsheet errors (Figure 4b). 42% of the students making errors using the AVERAGE function were able to carry out the calculation. The number of spreadsheet errors made when using the MIN or MAX function (36%) is lower than the number of spreadsheet errors in the SUM and AVERAGE function (Figure 4c).

As soon as a function requires more advanced mathematical/ logical knowledge we observed that the proportion of pure mathematical/ logical errors increases, for example, when the students have to deal with the IF-function (27%) or with the PMT/ FV-function (31%). About a third of the errors made when using the IF function were related to spreadsheet errors. It is obvious that some students know the Mathematics and the logic of the task, but they were not able to construct the required function correctly. For the PMT and the FV function we couldn't identify any "pure" spreadsheet errors. The majority of errors that occurred can be related to both categories.

In simple formulae 46% of the errors made were related to spreadsheet errors. Most of the spreadsheet errors were due to cell referencing. This is even more obvious when we look at absolute cell referencing. 58 % of the errors (GE5a, GE2c, SE5) made were due to the fact that students had difficulties with the concept of absolute cell addresses.

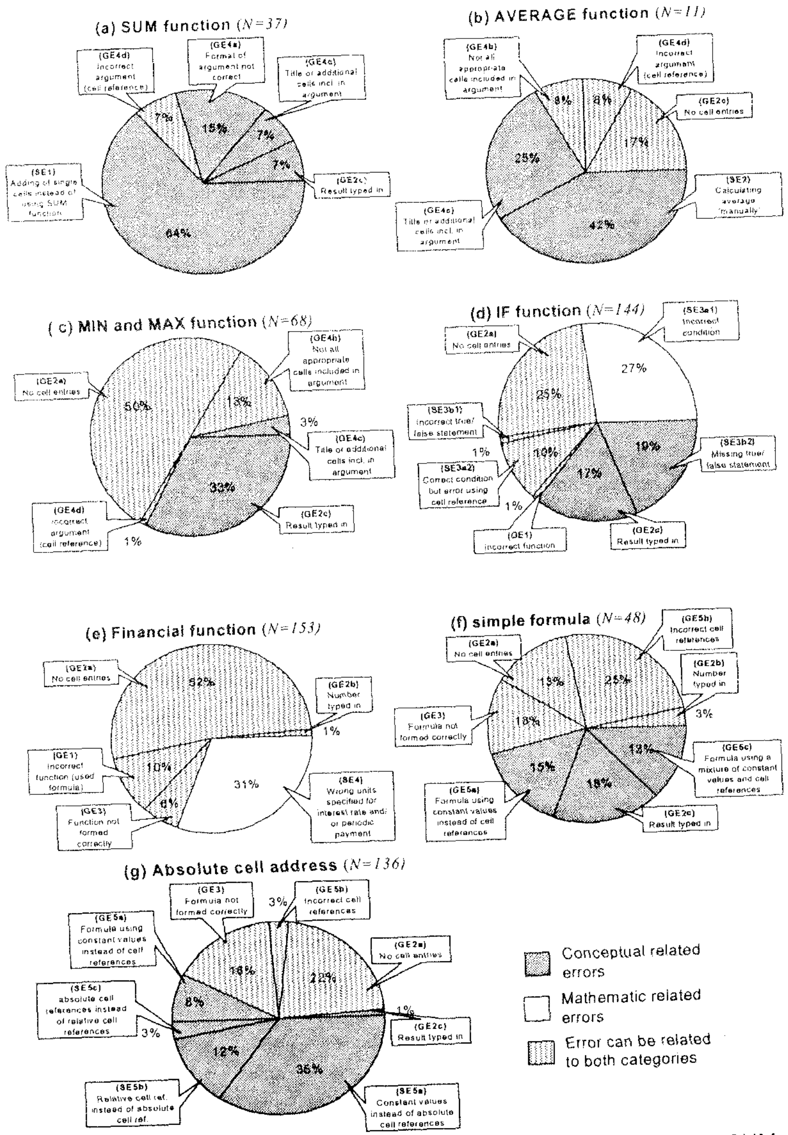
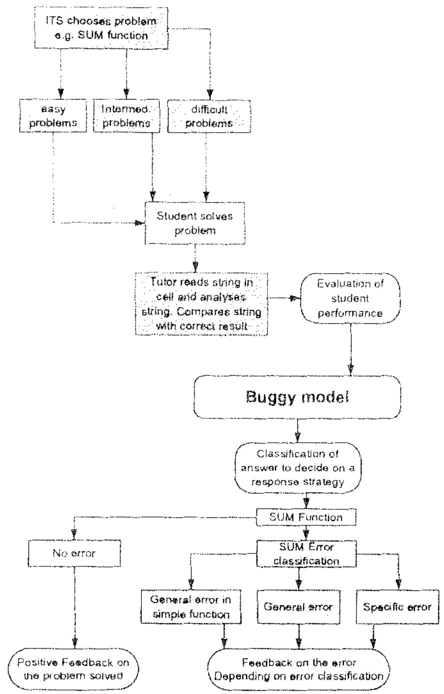


Figure 4: Distribution of spreadsheet and mathematical errors in (a) SUM-, (b), Average- (c) Min/Max-, (d) IF-, (e) PMT/ FV-function, (f) simple formulae and (g) formulae using absolute cell addressing. (Error keys see Table 2 in Appendix A).

6. Feedback Guidance Using the Buggy Model – An Example

Based on the different error categories we are now able to generate a buggy model for our ITS. The classification of general and specific errors allows us to structure the error library and to reduce the complexity of the library. The analysis of spreadsheet errors in terms of mathematical and spreadsheet knowledge will help us to provide the necessary assistance for the students and to guide the feedback in our buggy model. The investigation of student barriers also revealed that it is necessary to provide additional help for students with language problems, which needs to be implemented in our system.

Figure 5 shows the general concept of our buggy model. To demonstrate the operation of our buggy model we use a problem involving the SUM function as an example. The system generates a SUM function problem ("*Calculate the number of tickets sold*") that the student has to solve. Our model will generate problems with different levels of difficulty. These levels will depend on knowledge of the individual student. The student types his solution into the cell ("*1675*") and the system compares this string with the solution (*"1675" ζ "=SUM(C1:E5")*. If the student solves the problem he will get positive feedback. In this example the student made an error. The program now analyses and classifies the error to decide on a response strategy.



Problem:
"Calculate the number of tickets sold"

Student solution: **"1675"**

Comparison with correct result:
"1675" ≠ "=SUM(C1:E5)"
(SUM(C1:E5) returns value 1675)

Error classification:
"GE2c"
(see figure 6 for details)

Feedback on error with hint on what the error was:
"Your calculation is correct. But it is not dynamic. You need to enter a function into the cell."

Figure 5: Flow chart of feedback module with SUM function example

Figure 6 illustrates in detail how the buggy model will analyse the error and provide feedback to the specific problems using a decision tree. The student's solution to this problem is a number, but not the required calculation that uses the SUM-function. The decision tree identifies the error as a *GE2c error* (the student manually calculated the result and typed it in). The feedback presented by the system is similar to a response from a human tutor to this problem. The response that the system will provide is: *"Your calculation is correct, but it is not dynamic. You need to use a function for the calculation."* The feedback gives the student an indication of what his error is *"Your calculation is correct, but it is not dynamic"* and also provides the

student with additional information that his answer was not correct because he didn't use a function. *"You need to use a function for the calculation."*

The student has the possibility to correct his error and to submit his solution again. If this solution is correct he will continue with another problem, otherwise the system will provide additional feedback on the problems the student experiences.

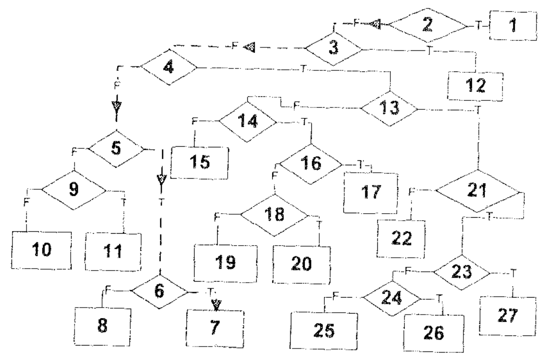


Figure 6: Error decision tree for feedback in SUM function

1 Correct solution
2 Check if "=SUM(cell range)" is correct
3 *GE2a*, Check if cell is empty
4 *GE3*, Check if function starts with "="
5 *GE2*, Check if entry is a number
6 Check if number = "result"
7 *GE2c*, "Your calculation is correct. But it is not dynamic. You need to enter a function into the cell."
8 *GE2b*, "You entered a number into the cell. Did you place your result in another cell? Or do you want me to" 1) "explain the task", 2) "explain the syntax of the SUM function"
9 Check if cell entry starts with "SUM"
10 "You entered a text into the cell. Did you place your result in another cell? Or do you want me to" 1) "explain the task", 2) "explain the syntax of the SUM function"
11 *GE3*, "You are on the right way. Check the syntax of your function if something is

missing."
12 *GE2a*, "You didn't fill anything into the cell. Did you place your result in another cell? Or do you want me to" 1) "explain the task", 2) "explain the syntax of the SUM function"
13 Check if function-name ="SUM"
14 Check if function has syntax "=...+...+"
15 *GE1*, "You used an incorrect function/formula" Do you want me to" 1) "explain the task", 2) "explain the syntax of the SUM function"
16 Check if function adds all cells in cell range
17 *SE1*. "Your calculation is correct. But it is better to use a function instead of a formula."
18 Check if formula adds cell references
19 *GE5a* and *GE5c*, "You should use a function and you should use cell references instead of values"
20 *GE5b*, "You should use a function and you should also check on cell references"
21 Check if cell ref has format "cellref : cellref"
22 *GE4a*, "You used the correct function. But you should check on the format of your argument."
23 Check if stud_cell range = cell range - cells
24 Check if stud_cell range = cell range + cells
25 *GE4d*, "You used the correct function. But your cell references are not correct." Do you want me to" 1) "explain the task", 2) "explain the syntax of the SUM function"
26 *GE4c*, "You used the correct function. But you should check if you included to many cells into your argument."
27 *GE4b*, "You used the correct function. But you should check if you included all the cells into your argument."

7. Conclusions

The main aim of this paper was twofold. For student modelling purposes, barriers that may prevent a student from successfully using a spreadsheet program were identified. Secondly, a survey of the different types of problems experienced by students was conducted. This information forms the basis of our buggy model.

We have demonstrated how the background of the student in terms of English language proficiency and mathematical knowledge can influence the performance in Microsoft Excel spreadsheets.

The study also showed that most errors occurred in areas where students had to use formulae and functions that require logical or mathematical thinking, or an advanced understanding of concepts, such as the use of absolute cell addresses.

The errors that the students made can be classified as spreadsheet errors, or mathematical-/ logical errors. Apart from the two error categories "mathematical/logical" and "spreadsheet" we could also distinguish between general (GE) and specific (SE) errors.

We implemented the result of this study in our buggy model and are now able to provide error- specific feedback and support.

Bibliography

Baffes, PT & RJ Mooney 1996. A Novel Application of Theory Refinement to Student Modeling. Proceedings of the Thirteenth National Conference on Artificial Intelligence, Portland Oregon: 403-408.

Brown, PS & JD Gould 1987. An Experimental Study of People Creating Spreadsheets. *ACM Transaction on Office Information Systems* Volume 5, Issue 3: 258-272.

Burton, RR 1982. Diagnosing Bugs in a Simple Procedural Skill. In DH Sleeman & JS Brown (eds.) *Intelligent Tutoring Systems*. London, UK: Academic Press: 157-184.

Koch, E & M Snyders 2001. The Effects of Video Supplemental Instruction on the Academic Performance in Mathematics of Disadvantaged Students. *South African Journal of Higher Education* Volume 15 Issue 1:138-146.

Mizoguchi, R & J Bourdeau 2000. Using Ontological Engineering to Overcome Common AI-ED Problems. *International Journal of Artificial Intelligence* Volume 11, Issue 2: 107 -121.

Moundridou, M & M Virvou 2003. Analysis and design of a web-based authoring tool generating intelligent tutoring systems." *Computers & Education* Volume 40: 157-181.

Panko, RR & RP Halverson 1997. Are Two heads Better than One? (At Reducing Errors in Spreadsheet Modelling?). *Office Systems Research Journal* Volume 15 Issue 1: 21-32.

Paras, J 2001. Crisis in Mathematics education. Student failure: challenges and possibilities. *South African Journal of Higher Education* Volume 15 Issue 3: 66-73.

Serengul, GV 1998. Intelligent Tutoring Systems Architecture. Accessed on 20 May 2004 http://www.cs.mdx.ac.uk/staffpages/serengul/Intelligent.Tutoring.System.Architectures.htm.

Tukuiainen, M 2000. Uncovering Effects of Programming Paradigms: Errors in Two Spreadsheet Systems. 12th Workshop of the Psychology of Programming Interest Group, Cozenza Italy: 247-266.

Appendix

Table 2: Overview of error types and error examples in formulae and functions (*GE* general-, *SE* function/ formula- specific errors)

	Error type	Error example
GE1	Student uses syntactically correct formula/ function, which is not appropriate for this task.	Student used the future value function although the calculation required the periodic payment function
GE2	Student is not able to write down a formula/ function at all.	
GE2a	No cell entries.	Student didn't enter any formula/ function into the cell at all.
GE2b	Number typed in.	Student entered an arbitrary number, which was not the calculated result.
GE2c	Result typed in.	*Student entered the result in form of a number or a text into the cell.*
GE3	Error in syntax when forming formula or function.	Student left out brackets in a formula or forgot to place an equal sign at the beginning of a formula or function.
GE4	Error in simple function – Erroneous argument	
GE4a	Format of argument not correct.	Student used a formula as the argument instead of cell references.
GE4b	Not all appropriate cells included in argument.	Student didn't include cell references that were required for the correct calculation.
GE4c	Title or additional cells included in argument.	Student included the title or additional cells at the end or at the beginning of the necessary block of cells.
GE4d	Incorrect cell reference (argument)	Student referred to wrong cell references and therefore calculated something completely different from what was required.
GE5	Error in formula – error using cell references	
GE5a	Formula using values instead of cell references.	Student applied the correct equation but instead of using cell references that allows the calculation to be dynamic they typed in the constant value found in that cell.
GE5b	Incorrect cell reference.	Student referred to a cell that contained text or to a cell, which was not relevant for the calculation.
GE5c	Mixture of values (instead of cell references) and cell references.	Student used a cell reference for variable1 and a constant value for variable2 in the same formula although he could have used a cell reference for variable 2.
SE1	Adding individual cells instead of SUM function	Student added individual cells instead of using the SUM function (e.g. =A2+A3+B2+B3 instead of =SUM(A2:B3))
SE2	Calculating average manually	
SE2a	Using SUM function.	Student used SUM function instead of AVERAGE function for the average calculation (e.g. =SUM(A2:A6)/5)).
SE2b	Using formula.	Student used formula instead of AVERAGE function for the average

		calculation (e.g. =(A2+A3+A4+A5+A6)/5).
SE3a	IF-function - Condition	
SE3a1	Incorrect condition	Student used an incorrect condition.
SE3a2	Correct condition, but error in cell reference	Student was supposed to use absolute cell reference in condition but used relative cell reference. Student used an incorrect cell reference in condition.
SE3b	IF-function – True/ False statement	
SE3b1	Incorrect statement.	Student used the same statement for the true and the false statement.
SE3b2	Missing statement.	Student left out false statement.
SE4	Financial function – Erroneous argument	
SE4a	Wrong unit specified for interest rate.	Student used yearly interest rate whereas a monthly interest rate was required.
SE4b	Wrong unit specified for time period.	Student used yearly periodic payment whereas monthly periodic payment was required.
SE5	Formula – Absolute cell reference	
SE5a	Values instead of absolute cell reference.	Similar to GE5a. Student used a constant value instead of an absolute cell reference.
SE5b	Relative cell reference instead of absolute cell reference.	Student used relative cell reference whereas an absolute cell address was required
SE5c	Made cell reference absolute	Student confused the concept of absolute and relative cell addresses and used an absolute cell address whereas a relative cell address was required.

Authors' Contact Details

Tanja Reinhardt (reinhardtt2@ukzn.ac.za)
Nelishia Pillay (pillayn32@ukzn.ac.za)
School of Computer Science
University of KwaZulu-Natal, Durban, South Africa

Establishing a New University Information Systems and Technology School: The Relationship between IS&T, BIS and IM

Ravina Sewlal
Sam Lubbe

Abstract

Very few have the opportunity in their careers to be in at the start up of a new university department – even fewer when this is brought about by the formation of a new university. The development of the University of KwaZulu-Natal campuses in January 2004 is such a development funded by a range of sources. This development has led to the formation of a 'new' School of Information Systems and Technology (IS&T). The question raised is "What is Information Systems and Technology?" This has been considered in some detail by other authors (e.g. Neville & Brayshaw, 1996) but everyone has to ask the question, in their own context, when planning a contribution to the development of the subject. In any event, university management quickly want the question answered in the form of detailed development plans for research and course provision.

This paper begins by giving a short history of the development of Information Systems and Technology at the university, also the nature of the changes taking place in the organisation. A definition of IS is developed by extending the discussion raised by Tully in his paper for the 1st UKAIS Conference (1996) (as *cited* by Neville & Brayshaw (1996)) on the associations between IS and other areas of study in related fields. The paper concludes by raising a number of general considerations all those developing the subject Information Systems and Technology should be wary of if a clear identity is to emerge for the subject area.

*Alter*nation 12.1b (2005) 455-466 *ISSN 1023-1757*

Keywords

Computer Science, Information Systems, Information Systems and Technology, Merger, Mission, Systems Sciences, Vision

Introduction

The Merger of the Universities of Durban-Westville and Natal

The Universities of Natal and Durban-Westville have merged on 1 January 2004. This was because the Department of Education had decided to reduce the number of tertiary institutions. The new merged institution has been renamed the University of KwaZulu-Natal. This name was decided on after consultation with all parties during 2003.

Background to the Merger

Ten years after new democracy in South Africa, the Higher Educational system is still affected by the legacy of a lack of multi-cultural approaches that does not take all factors (e.g. the apartheid years, etc.) into account. Despite internal and piecemeal restructuring efforts by individual institutions, the sector as a whole continues to experience poor human capital production levels, fragmentation along race lines, a lack of sustainability and a structural incapacity to meet the challenges of reconstruction and development. On 9 December 2002, the Minister of Education, Professor Kader Asmal, announced a series of mergers to be implemented as a means of addressing the structural problems affecting Higher Education in South Africa. (The Merger of the Universities of Durban-Westville and Natal:

(www.nu.ac.za/department/data/merger.pdf)

The New Institution

The merger between the universities of Natal and Durban-Westville is among a series of mergers in the tertiary education scene to take place in South Africa.

Some mergers on the card and suggested name changes are:

- RAU & Wits Technikon ➔ University of Johannesburg
- Peninsular Technikon & Cape Technikon ➔ CPUT
- UPE & PE Technikon ➔ Nelson Mandela Metropolitan University
- Some universities not affected by this are:
- UCT
- Wits
- Stellenbosch

As such, it heralds a new era for Higher Education in South Africa, offering an opportunity, recognised by both institutions, to create a South African institution based on academic excellence, critical engagement and demographic representivity. In both culture and form, the new university will be designed to meet the challenges of serving the country and the region in innovative and effective ways. It will be a world-class institution and an active global player while still serving the KwaZulu-Natal and Southern African region.

Vision and Mission of the University of KwaZulu-Natal

To be the premier university of African scholarship

(http://www.ukzn.ac.za/aboutus/mission.asp)

A truly South African university that is academically excellent, innovative in research, critically engaged with society and demographically representative, redressing the disadvantages, inequities and imbalances of the past.

By combining the strengths of both institutions, the new university will be in a position to offer:

- A wider range of academic programmes, across five campuses in KwaZulu- Natal;

- A diverse complement of highly specialized staff and academic experts;
- An institutional ethos and environment to support the intellectual, professional and personal
- development of both students and staff;
- Extensive facilities for all sporting, cultural and academic activities;
- An opportunity to be a part of the transformation of Higher Education in South Africa.
- (http://www.ukzn.ac.za/aboutus/mission.asp)

The leaders of this institution share a vision of an institution that will facilitate the highest levels of teaching, research and community service with the potential to become one of the of South Africa's leading research and innovation centres. It is envisaged that commerce will be placed at the Westville Campus in 2007.

The School of Information Systems and Technology

The School of IS&T at the Pietermaritzburg and Westville campuses belong to the Faculty of Science, while at the Howard College campus it is housed in the Faculty of Commerce. It has been agreed at all three campuses that Information Systems and Technology for the new university will belong to the Faculty of Commerce (see figure 2).

Previously the IS&T module was called Business Information Systems (BIS) at the former University of Natal. During the merger talks in 2003 it was decided to use a generic term IS&T for all three campuses.

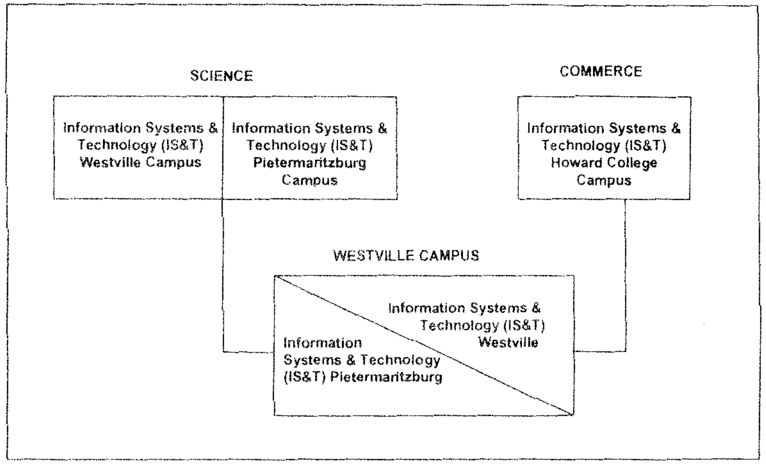


Figure 1: Department of Information Systems and Technology

What is Information Systems and Technology?

Tully as *cited* by Neville and Brayshaw (1996) in his address to the first UK Academy for Information Systems conference raised the position of the Discipline of IS as having relevance to a number of existing research areas. First it has its origins in Information Technology where the supporting hardware/software facilities to animate IS design were developed. The second, Systems Thinking, where some of the underlying organisational and human aspects have been discussed. He also makes reference generically to other areas which have contributed skills, techniques, and ideas all relevant to the development of Information Systems as a discipline. In his paper the authors discuss IS in relation to IT/Computer Science and Systems Thinking and argue why it merits separate from other disciplines. This theme was of central importance to much of the discussions which took place at the first UKAIS Conference held at Cranfield University in 1996. Acknowledging its roots in other disciplines, and indeed their continuing influence upon future developments, we seek to highlight distinguishing features which set IS apart from more well established subject disciplines. A definition of what an information system is:

> *The study of information systems and their development is a multi-disciplinary subject and addresses the range of strategic, managerial and operational activities involved in the gathering, processing, distributing, and the use of information and its associated technologies in society and organisations.*

(Defined by the UK Academy for Information Systems as *cited* by Neville & Brayshaw, 1996)

O'Leary & O'Leary (2004) defines an information system as a "collection of hardware, software, people, data, and procedures that work together to provide information essential to running an organisation" and information technology(IT) as "computer systems, large or microcomputers, that provide understanding to the end user."

According to Whitten, Bentley & Dittman (2004) an information system (IS) as an arrangement of people, data, processes, and information technology that interact to collect, process, store, and provide as output the information needed to support an organisation. According to the same authors information technology (IT) is a contemporary term that describes the combination of computer technology (hardware and software) with telecommunications technology (data, image, and voice networks). For the purpose of this study the first definition will be used.

Central Issues in Delineating Information Systems

Communities currently live in an increasingly information rich world. People are all information users – and must be able to make use

of the facilities offered by government and industry. With the Internet increasingly being used in academia and industry, users now have access to a massive variety of information, images, sounds, and video. It is thus of importance to look for ways of making information accessible and usable, and this involves addressing fundamental problems in searching, navigation, and dealing with information load and complexity. There would seem little point in storing amounts of information if users are unaware of this information existence or whereabouts, or lack the appropriate toolset to retrieve and use that data.

Changing the Agenda of Human Computer Interfaces

Next is the manner of interacting with information systems, how do people design for information access in the new age? Some old approaches for information retrieval may still be appropriate in given sets of instances, but humans are now at the dawn of an era when they use their imaginations and experiment with new ideas about what information is and how humans are to interact with it. With the advent of total immersion Virtual Reality new possibilities open. However, even using new data visualization techniques, people have new ways of exploring and presenting data. Thus HCI shifts to a more visionary approach of designing for the future-not just optimizing the present.

Some Fundamental Problems

Central to issues of designing information spaces are a series of fundamental issues.

These are listed below:

- Conceptualisation – what model of the information system is appropriate for users to have in their head?
- Navigation – seeing the wood for the trees. In particular knowing what's out there and where you are.
- Searching – sorting the wheat from the chaff.
- Interaction – how are users to interact with the information space available to them? What types of toolset are appropriate to enable the previous three activities?

- Appropriateness – what types of models are appropriate for which set of tasks and what sort of users?
- Adaptability/flexibility – to what extent is it necessary or desirable to allow users to modify their ability to interact with the information systems or the conceptual model behind it.

There are a series of possible ways of designing information spaces to explicitly deal with these issues.

Why Information Systems?

There have now emerged application areas where computers are being used as information servers, often for vast quantities of data. For the reasons outlined above, if their use is to be maximized for potential users then fundamental obstacles have to be overcome. For a user base that can frequently comprise more relatively unskilled or casual users, it is important to study and identify ways to maximize their use of our information reserves. Structure is also an important issue, and when one looks at Management Information Systems, not just the information itself needs to be considered, but issues concerning strategic planning, and frequently the type and nature of the business itself. For all these reasons their now seems to emerge a specialized related area which requires treatment in its own right in the light of the unique set of obstacles and challenges that academics now face.

By studying information systems as a distinct entity academia can thus identify and offer generic solutions to many of the fundamental problems that we now face. It is timely and important that academics now do so. The UKAIS was formed by a small group of IS experts who have recognised this need for a distinct area. Without this distinct recognition of a new discipline it becomes increasingly difficult for IS research as a traditional demarcations for funding in the UK Research Councils and Research Assessment Exercise hide IS in with computing. The result is that research which is more directed away from hard systems development paradigms has to seek funding from other bodies

which would tend to classify IS as computing. Many find themselves trapped in a situation where they have no clear profession/academic/funding body to represent them – hence the drive for the formation of the UKAIS. So they are purely pragmatic (as well as academic) arguments which require the clear identity of IS to be established. (Neville & Brayshaw, 1996).

Why not Systems Sciences?

Information Science is the study and development of artifacts; including the development process and management role. System ideas are of direct relevance to the process of development. There have been many attempts to formalise software design approaches (e.g. Whitten *et al.,* 2004). Some of these have been harder approaches, made from an engineering perspective. Others, taking a more socio-technical systems, focus more on process than being by technological determinism. Whilst it clearly is that case that this means that system sciences have an important role to play in IS development; the precise methods by which this takes place are still relatively unclear. Thus whilst system sciences are relevant, it is still a research question about how to formulate these processes in the IS context and thus deserves consideration in its own right.

Why not Computer Science?

For Information System (IS), computers are just a convenient form of delivery. People can conceive that IS does not require machines (e.g. the card indexes of old libraries). As a tool, computers do impact on the implementation of information systems but the area of study is in structure and use of the information, the application layer, and not on the other layers of technology that might underpin this. Thus the computer is interesting in terms of functionality it allows the user to exhibit at an application layer, but in itself is not the focus of attention. An IS may need to incorporate different data centres, corporate system, management systems, marketing systems, manufacturing systems, as well as other information service providers. Its concerns are not limited to those software sciences alone. It thus represents a sizeable application area in its own right and one that deserves explicit treatment. A counter to the above might be to ask 'ah yes, but aren't all software systems information systems, even if the information is just ones and noughts'. The answer to this is yes and no. The goal of the software is often something different, for example solving complex mathematical equations. This is distinct from IS where the explicit point about their existence is to provide an information service. There is however a grey area where things may overlap. An example is scientific visualisation where scientific data (e.g. chemical structure) is replayed back to the scientists via visual models. Here indeed is a case of a scientific software system that is also to some extent an information system. It is to be hoped that many of the ideas contained in these systems can be integrated with general models of information systems, not least when the user wish to display complex scientific data.

Opportunities Provided by Information Sciences

Information Systems, as the name implies, are systems in their own right. However they may differ from other systems in their content, structure, plasticity, and in their virtual makeup. The information can come in a wide variety of forms, for example pictures, video, sound, text, in various formats e.g. video conference, virtual reality, hypertext, and from diverse sources e.g. internal or external networks, hard disks, databases, CD's or the Internet. There is also more to Information Systems than just the raw provision of the desired information. Users are now capable of building up new experiences for users which are unlike anything we have encountered before. For example, physical presence is no longer a prerequisite for many activities, so students can now attend virtual universities (EISE 1996), people can surf around virtual libraries and museums, or even be virtual tourists (e.g. http://www.paris.org) . These new experiences may also be shared with other users, or software posing as other users. Such software may include artificial intelligences that interact or even collaborate with users when they use an information system (e.g. Whitten *et al.,* 2004). These interactions can take place using conventional communication channels e.g. email, phone, synchronous or asynchronous conferencing, or video conferencing, or they could take place in a shared environment like a virtual reality meeting room or office. However users are not limited to basing shared experiences on some software copy of the world as already experienced, but are able to dream up new conceptual structures appropriate to given tasks and goals. An example of such a created reality is a data network like cyberspace (O'Leary & O'Leary, 2004) which is often described as a shared hallucination whereby users perceive as real an information space populated by people, AI's, and large organisations. All these concepts are new and challenging ideas which push us to new limits in such questions as:

- How we access and use machines?
- How we behave as people?
- How we interact with a world which can increasingly be moulded by ourselves?
- How we interact with other people?
- How we perceive seemingly intelligent agents we meet?
- How all this is used and integrated into the society and our existing organisational structures?

The implications and repercussions that information systems are going to have make it important that they are considered explicitly under a separate banner headline. Many of the things that we have to consider challenge assumptions under which traditional disciplines have laboured. The move towards virtual world means that many hard constraints are now soft.

Conclusion

In this paper the authors have argued that information systems deserve to be treated as a separate related discipline. Within the term information systems they have included a large number of different emerging technologies all with some information bearing/bearing content. They think this is justifiable because these rapid changes, and indeed the emergence of totally novel (and sometimes unforeseen) technologies, mean that conventional label sets should not be strictly applied. Rather at this stage it is better to stand back and see what emerges, and be inclusive in definitions rather than exclusive.

References

Neville DE & Brayshaw M 1996. *Establishing a New University Information Systems Department*. Brayford Pool: Lincoln University.

O'Leary TJ & O'Leary LI 2004. *Computing Essentials*. New York: McGraw-Hill Companies, Inc.

The Merger of the Universities of Durban-Westville and Natal http://www.nu.ac.za/department/data/merger.pdf accessed 20th May 2004.

University of KwaZulu-Natal http://www.ukzn.ac.za/aboutus/mission.asp accessed 20th May 2004.

Whitten JL, LD Bentley, & KC Dittman 2004. *Systems Analysis and Design Methods*. New York: McGraw-Hill Companies, Inc.

Authors' Contact Details

Ravina Sewlal (sewlalra@ukzn.ac.za)
Sam Lubbe (slubbe@ukzn.ac.za)
School of Information Systems & Technology
University of KwaZulu-Natal, Durban, South Africa

Factors Affecting Usage of Web Based Learning Tools

Lisa Seymour
Irwin Brown
E Beck
F Khatree
L Morley

Abstract

Several studies on technology adoption have attempted to develop models, such as the technology accepted model, that can be generally applied to any technology. Typical web-based learning technologies such as WebCT, however, are composed of several distinct tool sets, and student motivations for using each may differ, due to the different purposes and characteristics of each tool. In this study, a different set of factors were found to affect usage intent of the WebCT bulletin board and quiz tool and these differences are discussed. Perceived usefulness and Perceived ease of use were found to affect student intentions to use the bulletin board, while Compatibility with learning style; Self efficacy and Long-term consequences all affected intentions to use the quiz tool. The implications of these findings on designing the integration of these educational technology tools with courses are discussed as well as limitations and future research.

Keywords

computer-based learning; web-based learning; WebCT; Bulletin board; Quiz tool; TAM; learning environments

1 Introduction

With the rise of the Internet there has been increased potential for information technology to be integrated with education. Wernet *et al.* (2000) propose that there is a current research trend to explore new and varied methods of teaching, and educational institutions now have the opportunity to work through infrastructures that support student learning both within the classroom and outside of it. Recognised benefits of using educational technology include increased flexibility, interactivity in learning, improved communication, accessibility and availability (Singh & Blewett 2003). Over the past decade many online learning environments have been developed. One of the most successful of these is WebCT (Web Course Tools), which proclaims to be the world's leading provider of e-Learning solutions for higher education and lists eight South African higher education institutions as customers (WebCT n.d.).

This paper aims to identify the factors affecting intentions to use WebCT tools by tertiary level students. The WebCT environment has five main tool sets from which courses can be designed:

- Course Content (e.g. Syllabus, Content Module)
- Communication Tools (e.g. Chat Room, Bulletin Board)
- Evaluation Tools (e.g. Self Test, Quiz Tool)
- Student Tools (e.g. My Progress, My Grades)
- Content Utilities (e.g. Search, Compile)

Most of the literature reviewed has assessed student reactions to WebCT as a general application and has not reported on differences in user perceptions and usage found between the tools. To overcome this weakness, this study looks at two WebCT tools - the WebCT bulletin board, referred to as Discussions within WebCT, and the quiz tool, and analyses the differing influences on usage intent.

In the following sections, the conceptual background to the study will firstly be presented, before the research propositions are outlined. The research methodology follows and the data analysis and results are

then presented. A discussion of the results and implications for future research are reported, leading to the conclusion of the paper.

2 Conceptual Background

Information technology can be used to facilitate and support the learning process, and serve to integrate information. Kendall (2001:1) states that 'rapid developments in information and communications technology have improved opportunities for individuals and groups to communicate and share information directly with each other through community networks.' Knowledge management tools, such as the Internet, intranets, course websites and online library databases are increasingly being used in course delivery.

WebCT (Web Course Tools) is an online learning management system, which like many similar tools provides educators with a web site template into which content is added, and components customised to suit the particular course (McClelland 2001). WebCT provides access to 'a collection of course-related materials, such as syllabus, assignments, readings, lectures, class notes, study guides, selected papers, and general announcements' (Benbunan-Fich 2002:96). It provides added functionality, with communication tools such as bulletin boards and facilities to send out e-mail that promote interactivity between students, and between the lecturer and students. The bulletin board is an area wherein students and lecturers can hold online discussions and post messages to one another. It can thus be described as interactive, social, informational, and supportive.

Within the evaluation module, the quiz tool can be used to post tests and surveys online, and has a high level of academic relevance, for both revision and testing purposes. In a recent study examining WebCT usage, the quiz tool was found to be the most extensively used (Knol & Vincent 2002). In contrast, to the bulletin board, it can best be described as evaluative, individual, performance-related and in some cases, intense. These different characteristics are expected to yield different motivations for usage intent.

In determining what factors influence intentions to use these tools, this study uses an expanded technology accepted model (TAM) proposed by Brown (2003). This model combines different models, including TAM and the decomposed theory of planned behaviour (Taylor & Todd, 1995). The basis of all extended TAM frameworks is that user perceptions of a technology are important predictors of user acceptance of that technology (Brown 2003).

In the Brown (2003) study, the expected influences on intentions to use the Internet as a learning tool were categorised as cognitive instrumental processes, social influence processes, and perceived behavioural control factors (See Figure 1).

Cognitive instrumental processes are defined by Venkatesh and Davis (2000) as the mental representations that are used in order to decide whether to adopt a technology. Of these factors, Perceived ease of use and Result demonstrability were shown to have little influence on usage intent (Brown 2003). However, only Result demonstrability was excluded from this study, as Perceived ease of use, was shown to be extremely relevant in a prior study that examined WebCT usage (Brown 2002). The remaining five factors considered as influences on usage in this study are defined as follows:

- Perceived usefulness (PU): The degree to which a person believes that using a particular system would enhance his or her learning (job) performance (Davis 1989).
- Long-term consequences (LTC): The increased flexibility to change work or increased opportunities to do more meaningful work (Chang & Cheung 2001).
- Compatibility with learning style (CLS): The degree to which an innovation is viewed as being consistent with the existing learning styles (adapted from Agarwal & Prasad 1997).
- Perceived ease of use (PEU): The degree to which a person believes that using a particular system will be free of effort (Davis 1989).

- Perceived enjoyment (PE): The perceived degree of enjoyment with using a system (Venkatesh 2000).

Of the social influence factors, Subjective norm, which had no influence, was dropped from further consideration. The remaining factor, Perceived voluntariness, was shown to have a significant effect on usage intent, was retained and is defined as follows:

- Perceived voluntariness (V): The extent to which users perceive the adoption decision to be voluntary (Agarwal & Prasad 1997).

Both perceived behavioural control factors, Self efficacy and Facilitating conditions, showed no significant influence on usage intent (Brown 2003). However, a previous study by Brown (2002) examining WebCT usage specifically, found this category to be important and therefore, these factors, defined as follows, were retained for further investigation:

- Self efficacy (SE): An individuals' self-confidence in his or her ability to use a technology (Venkatesh 2000).
- Facilitating conditions (FC): The availability of external support needed to use a technology (Venkatesh 2000).
-

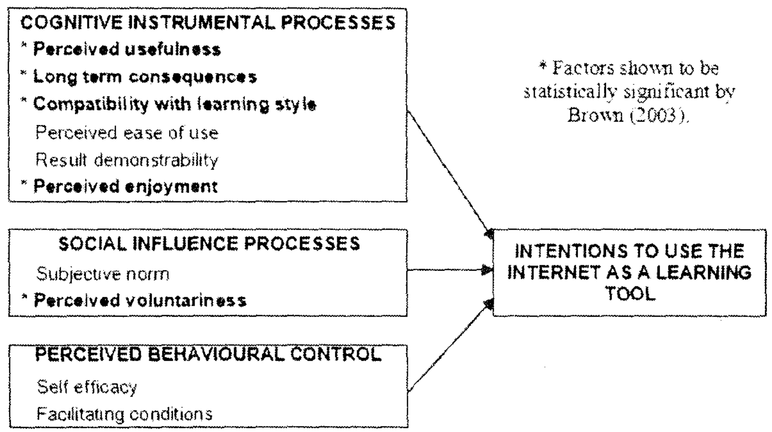


Figure 1: Expanded TAM taken from Brown (2003) and modified

The dependent variable used in extended TAM frameworks to measure technology acceptance, has been either use of the technology or intentions to use the technology, or in some cases both (Venkatesh & Davis 2000). Chang and Cheung (2001:1) claim that 'intention to use a technology is equally important [to usage], not only for promoting a technology but also for encouraging its voluntary continued use.' For this reason, intentions to use rather than actual usage is employed as the dependent variable in this study.

3 Research Propositions

The above set of factors was selected as there was sufficient evidence to expect that they would influence usage intent of learning tools such as WebCT. The contention of this study, however, is that WebCT, and other such learning technologies comprise of a suite of tools, and the relative influence of factors on usage intent for each specific tool may vary, due to their unique purpose and characteristics. Support for this argument comes from Gefen and Straub (2000), who found that the effect of perceived ease of use in e-commerce adoption varied, depending on whether a web site was to be used for a simple enquiry, or for actual purchase of a product. Each of the factors specified in Section 2 will be considered in turn, and their expected relative influence on the bulletin board and quiz tool respectively will be discussed leading to a set of propositions.

3.1 Perceived Usefulness

Perceived usefulness has been shown to be central to technology adoption across a wide variety of technologies and settings, with few exceptions being reported (e.g. Anandarajan *et al.* 2002). This construct has also been described as near-term usefulness, as opposed to the long-term usefulness construct (Chang & Cheung 2001). In the case of the bulletin board, near-term usefulness might be more salient than for the quiz tool. By posting queries and/or reading items already posted to a bulletin board, a learner may be able to acquire information

useful for more immediate concerns relating to the subject at hand. The quiz tool, on the other hand, might provide a testing environment which students would only want to use once they feel prepared enough to do so, possibly at a later stage of the learning process. Thus, its usefulness in the near-term is not as salient as the bulletin board. This leads to the following proposition:

Proposition 1:

Perceived usefulness is a more salient factor for intentions to use the bulletin board, rather than the quiz tool.

3.2 Long-Term Consequences

For the reasons postulated previously, it is expected that conversely, long-term usefulness, or Long-term consequences of use is more salient for the quiz tool, when compared with the bulletin board. In addition, technologies such as the quiz tool would be more likely associated with passing tests and examinations, which in turn students hope, will lead to positive long-term career benefits.

Proposition 2:

Long-term consequences of use is a more salient factor for intentions to use the quiz tool, as compared to the bulletin board.

3.3 Compatibility with Learning Style

The quiz tool is generally associated with testing, and thus may evoke feelings of anxiety amongst students. Students who feel they are not yet ready to be tested, would most likely shy away from the use of such a tool, until such time as they feel prepared. This could lead to postponement of use until absolutely necessary. The bulletin board, on the other hand is not generally associated with such stress, and would be more easily used. As a consequence it is expected that Compatibility with learning style would be a major influence for the quiz tools, but not so much so for the bulletin board.

Proposition 3:

Compatibility with learning style is a more salient factor for intentions to use the quiz tool, as compared to the bulletin board.

3.4 Perceived Ease of Use

Perceived ease of use, like Perceived usefulness, has been found in a wide array of settings to be an influence on usage and intentions to use a technology. A tool that is to be used for social interaction and communication (e.g., bulletin board) is more likely to be influenced by ease of use than a tool that is task-specific and task-oriented (e.g. quiz tool). This follows, as it has been shown that cultures or people that are more social and community-oriented, are more likely to use a tool based in its perceived ease of use than those who are individualistic and task-oriented, whose prime concern is usefulness (Venkatesh & Morris 2000, Anandarajan *et al.* 2002).

Proposition 4:
Perceived ease of use is a more salient factor for intentions to use the bulletin board, as compared to the quiz tool.

3.5 Perceived Enjoyment

Perceived enjoyment has been shown to be closely related to Perceived ease of use, especially as experience with a technology grows (Venkatesh 2000). Thus the arguments that apply to ease of use may also apply to enjoyment. Furthermore, given the generally relaxed informal nature of the bulletin board, its usage is likely to be motivated more by perceived enjoyment than the quiz tool.

Proposition 5:
Perceived enjoyment is a more salient factor for intentions to use the bulletin board, as compared to the quiz tool.

3.6 Perceived Voluntariness

Perceived voluntariness has been shown to be important for usage of learning technologies. Brown (2003), for example, found this factor to be a key influence on intentions to use and usage. Where usage of a technology is mandated for a course, students are more likely to use it, than if usage were left voluntary. It may be expected that quiz tools would be used less, if not mandated, due to the possible anxiety associated with testing, and therefore voluntariness may be a more salient factor for the quiz tool.

Proposition 6:
Perceived voluntariness is a more salient factor for intentions to use the quiz tool, as compared to the bulletin board.

3.7 Self Efficacy

Self efficacy embodies the concept of self-confidence with respect to technology use. High levels of anxiety may reduce self-confidence, and thus lead to reluctance to use a technology (Venkatesh & Morris, 2000). The quiz tool is often used to test student ability, and is more likely to be associated with anxiety, and its negative impact on self efficacy. Thus, this construct is likely to be more salient for the quiz tool as compared to the bulletin board.

Proposition 7:
Self efficacy is a more salient factor for intentions to use the quiz tool, as compared to the bulletin board.

3.8 Facilitating Conditions

Facilitating conditions are those factors in the environment that provide support and assistance with technology usage (Venkatesh, 2000). Given the interactive nature of the bulletin board, where discussion postings can be consulted freely, and the tool itself can be used to request for assistance, this factor would be more salient for the bulletin board. For the quiz tool, on the other hand, students could feel that under test conditions, the type of support and assistance is very restricted, thus facilitating conditions are less relevant to its usage.

Proposition 8:
Facilitating conditions is a more salient factor for intentions to use the bulletin board, as compared to the quiz tool.

4 Research Procedure

The propositions were tested through a survey taken during a lecture in a class of first year students at the University of Cape Town who were reading for Commerce degrees, but none of whom were majoring in Information Systems. Technically focused Information Systems majors were excluded as they could have skewed the data. Not connecting the research sample to a specific major allows broader application of the findings.

The research subjects were specified as first year students who had completed one semester, in the hope that their background influences would be more pronounced than those of students who had been within the university environment for a longer period of time. This was expected to yield more varied and individual responses rather than more homogenous ones. The students had gained exposure to WebCT in a statistics course in their first semester, which was compulsory for all first year Bachelor of Commerce students and had made express use of the WebCT bulletin board and quiz tool.

The questionnaire used in this research was based on three questionnaires used in studies of technology adoption for learning (Brown 2002, Knol and Vincent 2002, Brown 2003). Additional questions were added to establish some of the demographic variables. Each of the independent and dependent variables listed in the propositions were tested for the WebCT bulletin board and quiz tool respectively. Other than for Perceived enjoyment, the constructs consisted of multiple items. Each item was measured using a seven-point Likert scale, ranging from Strongly Disagree (1) to Strongly Agree (7).

Table 1: Respondent Profile

		Number	Percent
Gender	Female	133	55%
	Male	102	42%
	Did not specify	9	4%
Age	Under 20	172	70%
	20 – 23	59	24%
	Over 23	8	3%
	Did not specify	5	2%
Race	Asian	10	4%
	Black	92	38%
	Coloured	45	18%
	Indian	13	5%
	White	66	27%
	Other / Did not specify	18	8%

The students were given a brief explanation of what the research entailed, and were then allowed approximately ten minutes to complete the printed questionnaire. Incomplete questionnaires were rejected. The remaining questionnaires were captured into Microsoft Excel with values being checked to ensure that they were within range.

At total of 244 useable responses were obtained, out of a potential 500 respondents, giving a 49% response rate. Table 1 shows the respondent profile. About 55% of respondents were female, with the majority (70%) under the age of 20. The race classification depicts the diversity of South African culture, with the major groups represented being Black (38%), and White (27%).

4.1 Reliability and Validity Analysis

Cronbach's alpha was used to test the reliability of the questionnaire's constructs. In order for a construct to be deemed reliable, it should have a Cronbach's alpha value of 0.7 or above (Nunnaly 1978). All of the alpha tests on the research constructs gained values above 0.7, except for Perceived enjoyment which could not be tested, since it only consisted of one item (See Table 2).

Table 2: Reliability Analysis

	Bulletin board	Quiz tool	Number of Items
Perceived usefulness	0.93	0.94	8
Long-term consequences	0.88	0.88	5
Compatibility with learning style	0.90	0.92	3
Perceived ease of use	0.91	0.96	4
Perceived enjoyment	-	-	1
Perceived voluntariness	0.77	0.90	3
Self efficacy	0.86	0.89	3
Facilitating conditions	0.70	0.70	4
Intentions to use	0.85	0.9	2

Table 3a: Validity Analysis for the bulletin board

Items	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7
PU1	**0.80**	0.04	0.16	0.04	0.01	0.07	0.09
PU2	**0.79**	0.06	0.16	0.01	0.10	0.25	0.06
PU3	**0.81**	0.06	0.15	-0.02	0.07	0.14	0.18
PU4	**0.69**	0.09	0.11	-0.07	0.13	0.30	0.17
PU5	**0.80**	0.09	0.20	0.00	0.03	0.12	-0.03
PU6	**0.78**	0.09	0.20	0.10	0.11	0.14	0.01
PU7	**0.75**	0.21	0.18	0.03	0.10	0.22	0.16
PU8	**0.72**	0.21	0.22	0.10	0.07	0.03	-0.14
LTC1	0.22	0.01	**0.78**	0.06	0.04	0.17	0.01
LTC2	0.26	0.04	**0.70**	0.02	0.02	0.07	-0.08
LTC3	0.27	0.00	**0.83**	0.02	0.02	0.12	-0.05
LTC4	0.15	-0.01	**0.83**	-0.02	-0.02	0.10	0.01
LTC5	0.13	0.03	**0.79**	0.03	0.11	0.00	0.04
CLS1	0.31	0.07	0.22	-0.05	0.23	**0.76**	0.13
CLS2	0.37	0.08	0.16	0.08	0.10	**0.82**	0.05
CLS3	0.34	0.11	0.13	0.07	0.10	**0.81**	0.09
PEU1	0.12	**0.84**	-0.04	-0.06	0.11	0.03	0.18
PEU2	0.10	**0.84**	-0.01	-0.08	0.13	0.04	0.25
PEU3	0.18	**0.85**	0.10	0.04	0.12	0.12	0.18
PEU4	0.17	**0.82**	0.04	-0.02	0.13	0.05	0.17
V1	0.18	0.19	0.03	**0.74**	0.03	0.09	-0.02
V2	-0.01	-0.09	0.01	**0.90**	0.02	0.03	-0.02

V3	-0.04	-0.23	0.04	**0.82**	-0.07	-0.03	0.01
SE1	0.16	0.24	-0.01	0.02	0.20	0.07	**0.83**
SE2	0.10	0.34	-0.09	0.04	0.12	0.09	**0.81**
SE3	0.06	**0.46**	0.02	-0.11	0.00	0.12	**0.70**
FC1	0.00	0.07	0.13	-0.06	**0.66**	0.09	0.13
FC2	0.12	0.08	-0.06	0.04	**0.78**	0.01	-0.10
FC3	0.09	0.13	-0.02	-0.01	**0.73**	0.10	0.14
FC4	0.31	0.23	0.18	0.04	**0.57**	0.24	0.20

Table 3b: Validity Analysis for the quiz tool

Items	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
PU1	**0.85**	0.10	0.06	-0.01	0.01	0.12
PU2	**0.86**	0.10	0.12	-0.02	0.05	0.09
PU3	**0.84**	0.02	0.09	-0.03	0.11	0.05
PU4	**0.79**	0.19	0.18	-0.09	0.01	0.05
PU5	**0.78**	0.17	0.21	0.01	-0.03	0.05
PU6	**0.77**	0.23	0.21	0.00	0.04	0.01
PU7	**0.77**	0.15	0.15	0.08	-0.03	0.14
PU8	**0.71**	0.11	0.27	0.13	0.00	0.02
LTC1	0.27	-0.01	**0.78**	0.05	0.03	0.03
LTC2	0.20	-0.09	**0.73**	0.10	0.02	-0.02
LTC3	0.23	-0.06	**0.84**	0.12	0.03	0.09
LTC4	0.12	0.06	**0.83**	0.08	0.01	-0.10
LTC5	0.13	0.07	**0.77**	0.06	0.10	-0.02
CLS1	**0.72**	0.01	0.08	0.09	0.32	0.17
CLS2	**0.76**	-0.01	0.10	0.14	0.31	0.13
CLS3	**0.73**	0.00	0.08	0.15	0.28	0.08
PEU1	0.14	**0.87**	-0.03	-0.11	0.13	0.28
PEU2	0.17	**0.88**	0.00	-0.09	0.13	0.27
PEU3	0.20	**0.88**	0.02	-0.07	0.07	0.19
PEU4	0.17	**0.90**	-0.02	-0.08	0.05	0.16
V1	0.08	-0.02	0.15	**0.86**	-0.11	-0.03
V2	0.09	-0.14	0.15	**0.90**	-0.08	-0.09
V3	0.06	-0.13	0.08	**0.88**	-0.07	-0.09
SE1	0.20	0.33	-0.05	-0.03	0.10	**0.82**
SE2	0.20	0.32	-0.03	-0.09	0.16	**0.78**

SE3	0.14	0.37	-0.02	-0.11	0.12	***0.78***
FC1	0.04	0.01	0.17	-0.23	***0.50***	0.31
FC2	0.09	0.04	-0.05	0.05	***0.79***	0.07
FC3	0.14	0.19	0.01	-0.18	***0.74***	-0.03
FC4	0.28	0.18	0.25	-0.10	***0.55***	0.30

Factor analysis was performed on the measurement items to ensure the constructs were valid. Validity is demonstrated when items load at greater than 0.4 on their own factor, and less than 0.4 on all other factors, using varimax normalised rotation, and assuming an eigenvalue of 1. Seven factors were expected to load for each of the bulletin board and Quiz tool, corresponding to the seven constructs having multiple items. Perceived enjoyment consisted of 1 item only, and so was not included in the analysis.

The factor loadings for the bulletin board grouped as expected, except for item 3 of self efficacy, which cross-loaded with the Perceived ease of use construct (value = 0.46). It still loaded higher on its own construct (value = 0.7), and so was retained (See Table 3a).

Factor analysis on the same items for the quiz tool shows that all items loaded as expected, with one minor anomaly - Perceived usefulness and Compatibility with learning style loaded on the same factor, demonstrating the close relationship between these two. The variance inflation factors (VIFs) for these constructs were less than 10, however, an indication that multicollinearity would not pose a problem (Tan & Teo, 2000). . Table 3b shows the results of factor analysis.

5 Results

Table 4 compares the means between the bulletin board and the quiz tool, with regards to prior similar experience, respondent perceptions, and usage intent.

In terms of years of experience, the bulletin board has on average been used for a longer period (1- 2 years), than the quiz tool (about 1 year). There is little difference in terms of frequency of use (few

times/month), and intensity of use (about ½ hour per average day). Perceptions on average do not differ much between the tools, except in the case of Perceived voluntariness, where use of the bulletin board is seen to be slightly more voluntary than the quiz tool. All other means are greater than 4, indicating positive perceptions. Intentions to use the tools are high (both 5, on a scale of 1 to 7), which shows that students in general are appreciative of the tools.

Table 4: Descriptive Statistics

	Bulletin Board		**Quiz Tool**	
	Mean	**Std. Dev.**	**Mean**	**Std. Dev.**
Years of use	2.7	1.9	2.1	1.5
Frequency of use	3.3	1.8	2.9	1.5
Intensity of use (hrs/day)	2.3	1.2	2.3	1.2
Perceived usefulness	4.4	1.1	4.6	1.1
Long term consequences	4.1	1.2	4.1	1.2
Compatibility with learning style	4.4	1.3	4.4	1.3
Perceived ease of use	5.4	1.0	5.5	1.2
Perceived enjoyment	4.5	1.3	4.3	1.5
Perceived voluntariness	3.4	1.5	3.0	1.7
Self efficacy	5.2	1.3	5.4	1.3
Facilitating conditions	5.0	1.0	5.0	1.0
Intentions to use	5.0	1.3	5.0	1.4

5.1 Proposition Testing

In order to test the propositions, multiple linear regression equations were created for each tool separately, with independent variables regressed onto the dependent variable, intentions to use. The beta coefficients obtained are shown in Table 5, with the significant values in bold. The beta coefficients were then compared to ascertain if the expected differences were apparent. This technique is similar to that used by Venkatesh and Morris (2000) in their comparison of technology adoption across genders. Propositions were found to be supported if the differences between the two tools were significant and in the same direction as that proposed. Table 5 also shows the results of the proposition analysis.

For the bulletin board two factors significantly influenced usage intent – Perceived usefulness, and Perceived ease of use. Interestingly, these are the same factors that make up the original TAM (Davis 1989). For the quiz tool, a different set of factors were significant – Long-term consequences, Compatibility with learning style, and Self efficacy.

Comparing the two sets shows that there is support for 5 of the 8 propositions. As proposed, Perceived usefulness (Proposition 1) and Perceived ease of use (Proposition 4) were more salient for the bulletin board, as compared to the quiz tool. Long-term consequences (Proposition 2), Compatibility with learning style (Proposition 3), and Self efficacy (Proposition 7) were more salient for the quiz tool, as compared to the bulletin board.

Three factors, Perceived enjoyment, Perceived voluntariness and Facilitating conditions were shown to have no significant influence on usage intent of either tool. For these factors, no significant differences between the two tools were found and therefore no support could be found for Propositions 5, 6 and 8.

Table 5: Results of Regression Analysis

Factors and Propositions	Bulletin board	Quiz tool	Proposition supported?
	Beta values		
1. Perceived usefulness	******0.30***	0.05	Yes
2. Long term consequences	0.00	****0.16***	Yes
3. Compatibility with learning style	0.12	****0.21***	Yes
4. Perceived ease of use	*****0.19***	0.13	Yes
5. Perceived enjoyment	0.02	0.01	No
6. Perceived voluntariness	-0.01	0.09	No
7. Self efficacy	0.08	*****0.23***	Yes
8. Facilitating conditions	0.08	0.05	No

Beta values marked in bold were significant as follows: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

6 Discussion and Implications

The results show that for the sample group, the students had positive perceptions of and high intentions to use both the bulletin board and the quiz tool. The quiz tool is used in summative assessment and therefore was found by students to be less voluntary than the bulletin board. However, other than for Perceived voluntariness, perceptions between the two tools did not differ significantly. In contrast, these results confirm significantly different influences on usage intent between the two tools.

It is interesting that the original TAM factors, Perceived usefulness and Perceived ease of use, were found to influence usage intentions for the bulletin board and not the quiz tool. The TAM model has generally been developed to explain the adoption of technology in work environments, and has been generalised across a wide variety of technologies (Venkatesh & Davis 2000). The bulletin board, in like manner, is a tool whose use is not restricted only to learning environments, but is applicable across a wide variety of contexts where electronic communications and interaction is required. In fact, students may use it for purposes other than learning (e.g. socialisation and entertainment). Thus, this finding is perhaps not surprising.

The quiz tool, on the other hand, is more specifically used for learning purposes, and thus the basic TAM model, developed to serve as a more general theory of technology adoption, is not entirely adequate to explain variations in usage intent. In such contexts, as the analysis reveals, Compatibility with learning style, Self efficacy and Long-term consequences are more important. Compatibility with learning style specifically has been found to be the main influence on intentions to use the Internet in a degree program (Brown 2003).

The implications for practice are that in order for educators to motivate usage of specific tools, they should be aware that "one size does not fit all". For the bulletin board, enhancing perceptions of usefulness, and employing mechanisms to enhance perceptions of ease of use may well lead to greater usage. For the quiz tool, on the other hand, there needs to be a focus on enhancing compatibility of tool usage with learning styles and development of self efficacy. Emphasising the positive long-term consequences in terms of improving chances of success in tests and examinations will also help.

7 Limitations and Future Research

The research has been limited to undergraduate students of Commerce, and this may limit generalisation across a wider learning context. Future research might therefore involve repeating the study across different faculties to ensure that the results are not biased to one discipline.

Future research might also look at improving on methodological weaknesses, such as the use of a single item to measure Perceived enjoyment. To improve reliability and validity of measuring instruments it is common in social science research to employ multiple items for each construct. However, on that score, an unforeseen area of resistance to the questionnaire came from students' reaction to answering multiple items, which they perceived as being repetitive and indistinguishable in some cases.

Various other factors could also be included in the research framework. For example, the literature points to level of skill, computer anxiety, image and visibility as possible influences on usage intent (Brown 2003). These factors may be responsible for variations in results not explained by the tested constructs.

The demographic data can be used to ascertain the effect of cultural and socio-economic background on adoption of learning technologies, so that the tools can be better used to help in dealing with the wide student diversity present in many tertiary level programs.

Finally, other tools present in WebCT such as the chat room and calendar can also be compared, to ascertain what factors might motivate their usage.

8 Conclusion

Although much research has been conducted on WebCT usage and acceptance, few studies have compared the various tools within the technology. These tools are significantly different from each other, ranging from bulletin boards to quiz tools. Some tools are interactive while others are static, resulting in each tool having unique purposes and characteristics. This study found that a different set of factors affected student intentions to use the bulletin board and the quiz tool respectively, which demonstrates the importance of individual learning tool consideration.

Perceived usefulness and Perceived ease of use were found to affect usage intent for the bulletin board, while Compatibility with learning style, Self efficacy and Long-term consequences affected usage intent for the quiz tool. The findings suggest that different approaches should be used in the introduction of each in a course. It may be necessary for educators to emphasise these differences and the advantages, appropriateness and relevance of the tools for specific tasks.

References

Agarwal, R & J Prasad 1997. The Role of Innovation Characteristics, and Perceived Voluntariness in the Acceptance of Information Technologies. *Decision Sciences* 28, 3: 557 – 582.

Anandarajan, M, M Igbaria, & U Anakwe 2002. IT Acceptance in a Less-Developed Country: a Motivational Factor Perspective, *International Journal of Information Management* 22, 1: 47 – 65.

Benbunan-Finch, R 2002. Improving Education and Training with IT, *Communications of the ACM* 45, 6: 94 – 99.

Brown, I 2003. The Role of Cognitive Instrumental Processes, Social Influence Processes and Perceived Behavioural Control in the Acceptance of the Internet as a Learning Tool, *Alternation* 10, 2: 89 – 121.

Brown, I 2002. Individual and Technological Factors Affecting Perceived Ease of Use of Web-based Learning Technologies in a Developing Country, *Electronic Journal on Information Systems in Developing Countries* 9, 5: 1 – 15.

Chang, M & W Cheung 2001. Determinants of the Intentions to Use the Internet/WWW at Work: A Confirmatory Study, *Information and Management*, 39, 1: 1 – 14.

Davis, F D 1989. Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology *MIS Quarterly*, 13,3: 319 – 340.

Gefen, D & D Straub 2000. The Relative Importance of Perceived Ease of Use in IS Adoption: A Study of E-Commerce Adoption, *Journal of the Association for Information Systems* 1(8): 1 – 28.

Kendall, M 2001. Teaching Online to Campus-based Students: The Experience of Using WebCT for the Community Information Module at Manchester Metropolitan University, *Education for Information* 19, 4: 325 – 346.

Knol, B & S Vincent 2002. *Web-based Learning Tools: The Influence of Cultural Values on Usage Behaviour*, Unpublished Honours Dissertation, UCT.

McClelland, B 2001. Digital Learning and Teaching: Evaluation of Developments for Students in Higher Education. *European Journal of Engineering Education* 26, 2: 107 – 115.

Nunnaly, 1978. *Psychometric Theory*, McGrawHill, New York.

Singh, UG & C Blewett 2003. The Cobbler's Children Have No Shoes: An Investigation into the Usage of Educational Technology Tools within University IT Departments in South Africa, *Proceedings of the 33rd Annual South African Computer Lecturers' Association Conference* 30 Jun - 1 Jul, Pilanesberg, South Africa [CD].

Tan, M & T Teo 2000. Factors Influencing the Adoption of Internet Banking *Journal of the Association for Information Systems*, 1(5): 1 – 42.

Venkatesh, V 2000. Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation and Emotion into the Technology Acceptance Model, *Information Systems Research* 11, 4: 342-365.

Venkatesh, V & F Davis 2000. A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies, *Management Science* 46, 2: 186 - 204.

Venkatesh, V & M Morris 2000. Why Don't Men ever stop for Directions? Gender, Social Influence, and their Role in Technology Acceptance and Usage Behaviour, *MIS Quarterly*, 24, 1: 115 – 139.

WebCT web site n.d. *About Us*. Accessed on 24th February 2004 http://www.webct.com/company.

Wernet, SP, RH Olliges & T. Delicath 2000. Postcourse Evaluations of WebCT (Web Course Tools) Classes by Social Work Students, *Research on Social Work Practice* 10, 4: 487–504.

Authors' Contact Details

Lisa Seymour (corresponding author lseymour@commerce.uct.ac.za)
Irwin Brown (corresponding author ibrown@commerce.uct.ac.za)
E Beck
F Khatree
L Morley
Information Systems Department
University of Cape Town, Cape Town, South Africa

The Functionality of a Requirements Elicitation Procedure Developed for Process Modelling within the Higher Education Application Domain

Alta van der Merwe
Paula Kotzé
Johannes Cronjé

Abstract

Although various application domains use requirements elicitation procedures to gather information and model the application domain, only a few guidelines mention the important characteristics that are essential for such procedures. This paper reports on identifying a set of characteristics for a functional requirements elicitation procedure within the higher education domain.

Introduction

The Internet is no longer a new technology in the higher education institution application domain. In a recent report published by Educause (Educause 2003) an increase in the number of institutions in the USA that use the Internet to provide web-based campus portals was reported to have risen from 21.2 percent in 2002 to 28.4 percent in 2003. Online registration facilities grew from 20.9 percent in 1998 to 70.9 percent in 2003. The same trend is noticeable in South Africa.

Traditional higher education institutions that have already incorporated e-learning into their curricula often claim to have gained an advantage over their competitors because they serve a larger number of students. If they wish to retain such an advantage, these institutions should continuously convert their existing processes so that they pro-

*Alter*nation 12.1b (2005) 489-514 *ISSN 1023-1757*

competitors do (Laurillard 1993; Bates 2000). However, the incorporation of e-learning into the curricula of traditional higher education institutions is a complex task (Luker 2000; Ryan, Scott et al. 2000). The main reason for dissatisfaction with e-learning and its often inefficient implementations is similar to those in most application domains where developments have not been successfully implemented (Pressman 2000; Whitten, Bentley et al. 2001). Reluctance to incorporate electronic innovations often originates in a failure to understand the application domain adequately. Such an understanding requires an expert appreciation of all aspects of e-learning technologies and strategies if it is to be successful.

Requirements elicitation is a technique used by organizations to describe and specify an application domain. Various requirements elicitation procedures are used to gather information and model environments in different application domains. In software development projects, for example, a number of software requirements engineering procedures are in use (McDermid 1993; Pressman 2000; Hickey and Davis 2003), and in the field of business process re-engineering, authors such as Davenport (1993) and Hammer (1990) describe specialized re-engineering elicitation procedures. In software and business environments numerous guidelines describe the characteristics of the procedures concerned. Although requirements elicitation procedures might therefore seem to be the ideal tools for building up an understanding of the higher educational domain, only a very limited number of descriptions of process modelling procedures for this environment exists, and likewise of the characteristics that a requirements elicitation procedure for this domain should adhere to (Bruno, Vrana et al. 1998; Tait 1999; Cloete, Van der Merwe et al. 2003).

The aim of our research is to gather information on the processes involved in creating a learning environment and in modelling the workflow between these processes. The objective of this paper is to identify the output characteristics of a functional requirements elicitation procedure applicable to the higher education domain. The identification of such a set of characteristics is especially beneficial to requirement elicitation procedure *developers* because it will help them to establish a procedure that includes all the important traits that such a procedure must possess. The recognition and inclusion of these vital traits will lead to the development of improved products in the course of re-engineering the current environment to include e-learning technologies. This would naturally increase the rate at which such products are successfully deployed and accepted. *Researchers* may also benefit from this paper because they are not only responsible for establishing and developing new knowledge that benefits society, but they work closely with practitioners to define much-needed standards.

The next section identifies the context of the paper with regard to the modelling of a complex environment, and elaborates on the procedure followed to establish the characteristics of the requirements elicitation and modelling procedures identified. This is followed by a section describing a requirements elicitation procedure that is applied in a higher education environment, and a section showing how this procedure adheres to the suggested characteristics. The penultimate section addresses the issue of the scientific validation of the reported research, while the final section makes a few concluding observations.

Identifying the Characteristics of the Requirements Elicitation and the Modelling of Processes

Modelling a complex environment, such as the changing educational domain, involves two main sub-fields, namely requirements elicitation and the modelling of the information gathered during the requirements elicitation process. *Requirements elicitation* is the systematic extraction and inventory of the requirements of a system (IEEE 1998). If a requirements elicitation procedure is to be considered effective, it should at least produce the initial goal (Rzepka 1989). *Process modelling* presents a technique (comprising several activities) that graphically depicts the series of processes that accomplish a predefined goal (Curtis, Kellner et al. 1992). The *process model* is the *structure* that represents a group of processes and their relationship to one another, which together accomplish a specific goal. These two sub-fields naturally exist within cyclic methodologies that have the aim of developing software or re-engineering current environments (Pressman 2000; Hickey and Davis 2003). Our focus, as illustrated in Figure 1, is on elicitation and modelling activities.

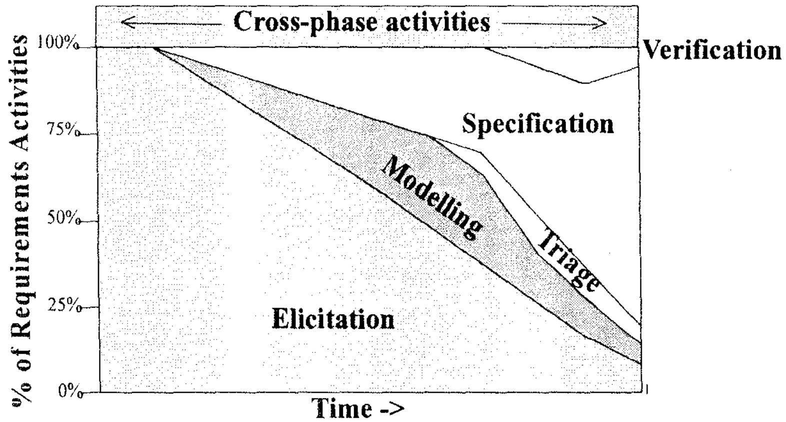


Figure 1: Requirements elicitation activities procedure (adapted from (Hickey and Davis 2003))

In order to identify the required characteristics of an elicitation procedure, we conducted a comprehensive literature review of sources that referred to characteristics within the field of study and described the characteristics of a range of features in requirements elicitation and modelling. Twenty-six of these resources mentioned useful characteristics. After a number of cycles of identification of characteristics and working through references, maturity occurred with fifty-eight identified characteristics.

Table A1 (Appendix A) shows a list of the twenty-six resources (a number has been assigned to each for further reference). Table A2 (Appendix A) specifies the list of characteristics that have been identified from these references, and includes a column with the corresponding references to a specific characteristic.

Although different authors propose different steps in the requirements engineering process, the core of these methodologies always in-

cludes (1) a feasibility study, (2) elicitation, (3) modelling, (4) triage, (5) verification and (6) cross-phase activities (Macaulay 1996; Sommerville and Sawyer 1997; Hickey and Davis 2003). We found that some of the characteristics identified as important for requirements elicitation actually belonged to other activities such as a feasibility stage – and were not relevant to an elicitation stage. By using these core steps and categorizing each of the characteristics into one of these steps, we ensured that we listed only those characteristics that were relevant to our study, namely the requirements elicitation and modelling phase (Appendix A, Table A2). We also merged characteristics with the same meaning and so ended up with a total number of fifty characteristics. In our last step, we grouped characteristics that naturally belong together into sub-phases (Appendix A, Table A2).

For the purpose of this paper, we are interested in only those characteristics that focus on the elicitation and modelling phases and in the ones that are applicable to all phases. Table 1 presents a list of these phases with the relevant characteristics identified. Although we appreciate the importance of the other phases, our focus in this paper will be limited to the phases mentioned.

A Requirements Elicitation Procedure for the Higher Education Environment

We now present an overview of a requirements elicitation procedure for the purpose of modelling a higher education environment. This procedure was developed and tested as part of a research project (Cloete, Van der Merwe et al. 2003) at the University of South Africa, and was also used as the fundamental requirements elicitation tool to determine the core and secondary processes at other institutions. The procedure consists of five phases. Phase 1 establishes objectives, whereas the identification of critical institutional units (Phase 2) and the identification of primary processes (Phase 3) help us to understand the domain. It is also during Phases 2 and 3 that the developers collect stakeholder requirements.

Table 1: List of characteristics

Phase	Sub-phase	Characteristic
All Phases	Support	Provide automated support for the requirements engineering process
	Standards	Provide standardised ways of describing work products
		The precision of definition of its notation
		Process model standards
	Techniques	Select appropriate technique for the problem domain
		Use of use cases to describe related tasks
		Support a systematic step-by-step approach
		Modifiable solutions and be iterative in
	Documentation	Support documentation of requirements
	Maintenance	Procedures for maintaining work products
	Conflict	Conflict negotiation
Elicitation	Specification	Requirement completeness
		Requirement relevance
		Expectations during specification of requirements
		Correctness
		Communication during specification of requirements
		Requirement accuracy
		Importance of necessity: requirements document
		Level of control over specifying requirements
	Boundaries	Specify constraints / boundaries
	Problem analysis	Support analysis
		Degree of understanding of the task and process
	Data gathering	Support data gathering techniques
	Client/customer	Support customer/client involvement
Modelling	Support modelling	Motivation to support modelling
	Goal modelling	Model the purpose by describing behaviour
	User involvement	Reflect the needs of customers / users
	Modelling	Model business rules
		Support modelling of workflows
		Clarity of business process
		Model system services
		Systems architecture modelling

The procedure continues with the organization of the acquired information into a high-level process model (Phase 4), which is refined

in the final step into several sub-process models (Phase 5). Each of these phases will now be described in more detail.

Phase 1: Establish High-Level Objectives

In Phase 1, the requirements engineering team, in cooperation with stakeholders, compiles a detailed description of the high-level purpose of the requirements elicitation exercise. The deliverable of the first phase is a descriptive document that acts as a framework for future reference and verification purposes. A document of this nature includes a short description of the goal(s) as well as a clear specification of the required deliverables.

Phase 2: Identify the Critical Institutional Units

The objective of the procedure is to identify the critical processes in the application domain (i.e. their essential activities and workflow) so that the application domain can be understood. The critical processes can only be identified from a consideration of different operational units within the institution. A *unit* refers to a working segment of the institution that is responsible for specific tasks such as, for example, a financial section, an academic department, a technical division, et cetera. As a first step, all such units within the institution are listed. This list is compiled from documentation and diagrams, such as organizational charts, or from data gathered using interviews. If e-learning is being contemplated, the second step involves extracting those units that actively create and present learning environments. The units that focus on other aspects of the institution are then labelled as *support units* and are deleted from the unit list. For example, while the catering services department prepares food, it has no direct responsibility the learning environment, and will therefore be removed from the unit list. The deliverable of Phase 2 is a list of the *critical* operational units of an institution.

Phase 3: Identify Primary Processes

In the next three phases, the procedure involves a formal approach to identify the relevant processes. The procedure distinguishes between primary and support processes in the application domain. Primary processes are the critical activities responsible for, or involved in, the design and construction of a student's learning environment. Support processes are those that provide sustenance for the primary processes and play a secondary role in accomplishing the defined goal. The purpose of Phase 3 is to identify the primary processes in each of the critical units of the application domain. The procedure suggests an initial list of primary processes for the e-learning domain that includes the registration process (REGISTRATION), development of course material (COURSE DEVELOPMENT), production of course material (PRODUCTION), distribution of course material (DISTRIBUTION), and academic support available to the student (ACADEMIC STUDENT SUPPORT).

The following steps will expand the list and verify its adequacy and completeness. These steps should be applied to each of the institutional units identified in the second phase (compiled into a unit list).

1. List and document the most important processes of a particular unit in order to establish the main duties within the unit. The focus should be on the goals to be achieved rather than on whatever individual activities might realize these goals. A general guideline is to include *what-processes* rather than *how-processes*. A *what-process* is goal-oriented in its description and expresses the objective of the particular process, while a *how-process* is action-oriented and explains the particulars of specific activities that will accomplish the specified goal.
2. Categorize each process as being either a support or a primary process by using the definitions provided earlier.
3. Attempt a mapping for each of the newly identified primary processes to an item on the initial list. A process list is created from items on the initial list that correspond to primary processes through their mappings, while primary processes that cannot be mapped are added to the process list as new items.

The deliverable of Phase 3 is a *process list* consisting of set of the identified primary processes, namely:

$$\{P_k\}_{k=1}^{m} \text{ for } k, m \in N_1$$

where m denotes the total number of processes for all critical operational units.

The procedure recommends that developers should reconsider the list if there are more than ten primary processes included in the list. Eriksson and Penker (2000) also comment that it is unusual, even for a complex environment, to have more than ten primary processes.

Phase 4: Construct the High-Level Process Model

The procedure for constructing a high-level process model employs a standard notation that includes the process itself, process resources and the goal description of the process (Eriksson and Penker 2000). Process resources can be either *input* (I) or *output* (O) resources. An input resource is used to assist in the flow of process activities. For example, in a student registration process, the registration form (input) is used to capture primary information about potential students. An output resource is the resultant output of activities in a specific process. It might in turn serve as an input resource for another process. Each process has at least one input resource and one output resource that is associated with it. The first step towards constructing the high-level process model is to define the goal, input resources, and output resources that are associated with each item on the process list, which had been created in the previous phase. At the end of this step, a set of all resources for primary processes of the application domain can be described as:

$$\{R_j\}_{j=1}^{n} \text{ for } j, n \in N_1$$

with n denoting the total number of input and output resources.

The second step is to indicate the workflow between different primary processes through input and output resources. This task remains simple provided that (1) there are only a small number of primary processes to consider, and (2) they can be accomplished simply by connecting related processes through directed lines on the process model diagram. However, as the number of primary processes increases, the degree of complexity in depicting the workflow increases proportionately. In such cases, the procedure would indicate that a more formal approach to establish relationships between primary processes is required.

The objective is to identify the resources that serve as both input and output resource for the different processes and then eliminate redundant resources (those resources that would appear more than once on the same process model diagram). To identify these resources, determine the association value (say T_{kj}) that a resource R_j has with a process P_k (for all j and all k). These association values may be input (T_{kj} = I), output (T_{kj} = O), or *no* association (T_{kj} = Null). Each T_{kj} is stored as an entry in an association list, which tabulates vertically all processes from top to bottom and tabulates horizontally all resources from left to right.

The following steps assist in indicating the work flow and associations between the different processes and as a result describe the high-level process model:

1. For $k = 1..m$ and $j = 1..n$, describe all the resources in terms of their association values with P_k. This is written as a triplet (P_k, R_j, T_{kj}). (Null values can be ignored.)
2. For $k = 1..m$, graphically depict P_k on a diagram with its associated goal.
3. For $j = 1..n$, add the identified resources, R_j to the diagram.
4. Use the set of triplets (identified in 1), in particular the third coordinate, to add directed lines between processes and resources.

This approach produces a high-level process model for the application domain, as is illustrated in Figure 3.

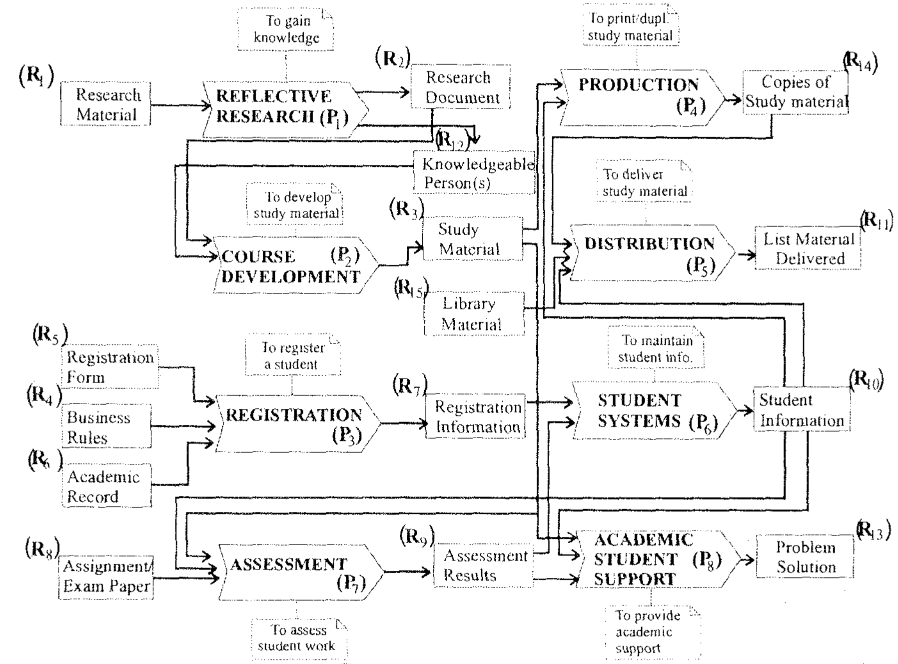


Figure 3: High-level process model for the higher education domain

Phase 5: Refinement

A complete understanding of the application domain is depicted through a single high-level process model with several smaller sub-process models to accomplish the intended goal. The purpose of the refinement phase is to decompose and particularize the individual processes in the high-level process model through iterative steps into a set of sub-processes or atomic activities. An atomic activity is a process that cannot be broken down into further sub-processes.

The activities required to depict the sub-models mentioned above are similar to those described in the previous phase for the high-level process diagram. In summary:

1. For each (primary) process, identify the set of affiliated sub-processes involved in the generation of its output resource(s). For each sub-process, define its associated goal, input and output resources.
2. Associate the sub-processes with one another through input and output resources, as described in Phase 4.
3. Draw the process model. This model graphically depicts the sub-processes and their relationships between one another.
4. Repeat these steps for each of the identified sub-processes until all processes are atomic – or until the requirements engineering team decides against further refinement.

The deliverable of this phase is a set of smaller sub-process models augmenting the high-level process model.

How Does the Requirements Elicitation Procedure Adhere to the Characteristics Identified?

In this section we consider the requirements elicitation procedure described above and portray its scientific soundness in terms of the characteristics listed in Table 1. Using the characteristics as a starting point, we evaluated and rated the requirements elicitation procedure with relation to each characteristic by using the following descriptors:

1. *Not adhere*: The requirements elicitation does not adhere to the characteristic at all.
2. *Partially adhere*: Some aspects of the requirements elicitation adhere to the characteristic.
3. *Strongly adhere*: The requirements elicitation procedure fully adheres to the characteristic.

The result of this rating of the different aspects of the requirements elicitation procedure is presented in Table 2. In the first column, we list the three phases followed by the sub-phases of each phase. In the third column, we include the characteristics identified followed by the rating achieved for each characteristic.

Table 2: Degree to which requirements elicitation procedure adheres to the identified characteristics

	Sub-phase	**Characteristic**	Does not adhere to (NA)	Partially adheres to (PA)	Strongly adheres to (SA)
All Phases	Support	Provide **automated support** for the requirements engineering process	√		
	Standards	Provide **standardised ways** of **describing** work products			√
		The **precision** of definition of its **notation**			√
		Process model standards			√
	Techniques	Select appropriate technique for the **problem domain**		√	
		Use of **use cases** to describe related **tasks**	√		
		Support a **systematic step-by-step approach**			√
		Solutions can easily be modified and are **iterative in nature**			√
	Documentation	Support **documentation of requirements**			√
	Maintenance	Procedures for **maintaining** work products		√	
	Conflict	**Conflict negotiation**	√		
Elicitation	Specification	**Requirement completeness**		√	
		Requirement relevance			√
		Expectations during specification of requirements			√
		Correctness			√
		Communication during specification of requirements			√
		Requirement accuracy			√
		Importance of **necessity:** requirements document			√
		Level of control over specifying requirements			√

	Boundaries	Specify **constraints / boundaries**			√
	Problem analysis	Support **analysis**			√
		Degree of understanding of the **task and process**			√
	Data gathering	Support **data gathering techniques**			√
	Client/customer	Support **customer/client involvement**			√
Modelling	Support modelling	Motivation to **support modelling**			√
	Goal modelling	Model the **purpose** by describing **behaviour**			√
	User involvement	Reflect the **needs of customers / users**		√	
	Modelling	Model **business rules**			√
		Support **modelling of workflows**			√
		Clarity of business process			√
		Model system **services**			√
		Systems **architecture modelling**	√		

The procedure strongly adhered to the use of *standard notation* and existing process model *standards*. It also supports a *step-by-step approach,* which is defined in the original documentation as *iterative*. Because reference is made more than once in the procedure to the output of a phase as being a set of documentation, it therefore also supports the use of *documentation* of the requirements.

Within the elicitation phase of the procedure, the procedure supports *requirement relevance* by excluding units and processes that are not applicable to the goal and modelling only the primary processes that are important in creating a learning environment. This goal and the limitations are expected in the beginning of the procedure. This indicates that the developers support the definition of *expectations* and the specification of *boundaries*.

The procedure suggests a systematic method for *gathering the information* from the different units – information that is *correct, necessary* and *accurate*. The procedure divides the educational environment into units for the purpose of gathering information, and uses *communication* techniques to extract whatever information is necessary from the *employees*.

The goal of the elicitation procedure is to *analyse* the current environment so that a different developer could, with this information and his or her understanding of the environment, identify *tasks* and *processes* within the educational domain.

Three of the five phases in the elicitation procedure are concerned with the modelling task. The procedure therefore strongly adheres to the modelling of *business rules*, *workflows* and different *services*. The procedure gives a *motivation* for using modelling in this application domain and also adheres to the *purpose* by producing the goal, the high-level process model, and sub-process models.

Table 3: Characteristics that the procedure 'does not adhere to'

Phase	Characteristic	Rating	Comment
All Phases	Provide **automated support** for the requirements engineering process	NA	While there is no *automated support* developed for the procedure, it should be possible to use existing tools (such as existing Case Tools) to support the documentation process.
	Select appropriate technique for the **problem domain**	PA	The procedure suggested only one way of gathering information. Other techniques such as questionnaires should also be appropriate for the application domain.
	Use of **use cases** to describe related **tasks**	NA	A few resources mentioned this as being important. The procedure did not include use cases to describe scenarios. Object-oriented notation supports the use of use cases.
	Procedures for **maintaining** work products	PA	While the procedure did not specifically mention the importance of maintenance, they support the use of documentation that is easily maintainable.
	Conflict negotiation	NA	No conflict negotiation is mentioned by the procedure.
Elicitation	**Requirement completeness**	PA	Although the procedure does not specifically define measurements to measure requirements completeness, they do suggest a cyclic system that tries to obtain complete requirements.
Modelling	Reflect the **needs of customers / users**	PA	Because the goal of the procedure is to model the current business processes, no need analysis is involved.
	Systems **architecture modelling**	NA	No system architecture modelling is included. This is important during the re-design of current workflows.

There are only a small number of characteristics that the procedure does 'not adhere to'. Table 3 includes all the characteristics that the procedure 'does not adhere to' or 'partially adheres to', with a comment in the last column on each of the ratings.

One characteristic that needs further investigation is the automated support for the requirements engineering process. As we mentioned above, it should be possible to use existing tools, such as existing case tools, to support the documentation process. Furthermore, although the procedure did not specifically mention the importance of maintenance, it supports the use of documentation that is easily maintainable. Cloete et al. (2003) do not give any guidelines on conflict negotiation although this is an important characteristic and research into this is necessary – especially in the educational domain where a diverse group of people is involved in development. The remainder of the characteristics are self-explanatory.

Validation of Requirements Elicitation Procedure

The objective of this work is to identify the characteristics that are required to render functional outputs for an elicitation procedure that could enable successful e-learning implementations in higher education institutions. In Section 3, we described the elicitation procedure for the application domain that was suggested by Cloete *et al.* (2003). Because of the scarcity of published research in this domain, we used this procedure as a basis for our work. However, scientific validation (and possibly augmentation) of the procedure is still necessary if it is to be rendered suitable and valid as an instrument that can be used by other researchers and practitioners. Such an instrument should be able to produce repeatable, usable and effective outputs that could overcome those obstacles to requirements elicitation that contribute to inadequate e-learning implementations.

In attempting to perform such a validation, we conducted a literature study over a wide set of application domains where requirements elicitation is conducted. We demonstrated earlier how requirements

elicitation and the subsequent modelling procedure are closely related. The literature review presented us with a list of desirable characteristics that the requirements elicitation and modelling phases in general should possess. By focussing on the specific application domain of this paper, we also extracted a similar, associated list of characteristics. Table 4 summarises these, lists the different phases, and lists the characteristics that the procedure adheres to in the specific phase.

Table 4: Phases in the requirements elicitation procedure, which adhere to the identified characteristics

	Characteristic	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
All phases	Provide standardised ways of describing work products			√	√	√
	The precision of definition of its notation			√	√	√
	Process model standards			√	√	√
	Support a systematic step-by-step approach	√	√	√	√	√
	Modifiable solutions and iterative in nature	√	√	√	√	√
	Support documentation of requirements	√	√	√	√	√
Elicitation	Requirement relevance		√	√		
	Expectations during specification of requirements		√			
	Correctness		√	√		
	Communication during specification of requirements		√	√		
	Requirement accuracy		√	√		
	Importance of necessity: requirements document	√	√	√		
	Level of control over specifying requirements		√	√		
	Specify constraints / boundaries	√	√	√		
	Support analysis	√	√	√		
	Degree of understanding of the task and process	√	√	√	√	√
	Support data gathering techniques	√	√	√		
	Support customer / client involvement	√	√	√	√	√

Modelling	Motivation for modelling support	√	√	√	√	√
	Model the purpose by describing behaviour				√	√
	Reflect the needs of customers / users	√	√	√	√	√
	Model business rules				√	√
	Support modelling of workflows				√	√
	Clarity of business process				√	√

From Table 4 it is clear that all the phases in the procedure support a systematic approach. It is iterative in nature (the procedure is cyclical and is only completed after a number of iterations). In all the phases, the information gathered by the developers is documented. This indicates that the procedure supports the documentation of the requirements and the documentation of the different models. Furthermore, in Phases 3 to 5, a notation used by modellers in process modelling environments, is prescribed. The characteristic *provides standardised ways of describing work products* is therefore adhered to. Similarly, the notation is precise and process model standards are used.

The only characteristic supported in only one phase of the procedure is the *expectations during specification of requirements*. This is understandable because this characteristic is only applicable to the specific phase of the procedure.

Conclusion

The main result of this research is a subjective instrument with fifty-eight indicators aimed at the higher education domain. We attempted to retrieve the indicators or characteristics from authors that commented on the characteristics of requirements elicitation and modelling procedures. We also extracted some from domains such as elicitation or modelling within software engineering or within business process re-engineering. This is, as far as we know, the first research effort that has resulted in an instrument of this nature.

The potential applications of our research results can be discussed from both research and practice perspectives. Researchers may use the instrument as a guideline during the development of similar requirements elicitation procedures. Practitioners using procedures that ad-

process re-engineering. This is, as far as we know, the first research effort that has resulted in an instrument of this nature.

The potential applications of our research results can be discussed from both research and practice perspectives. Researchers may use the instrument as a guideline during the development of similar requirements elicitation procedures. Practitioners using procedures that ad-here to a set of clearly defined characteristics can do so with the knowledge that the procedure is well-defined, and that it adheres to standards that are used in different application domains.

In further work, we plan to use the instrument to see how other requirements elicitation procedures within the educational domain adhere to the suggested indicators. According to various sources (Finkelstein, Ryan et al. 1996; Maiden and Ncube 1998) we shall in future see the development of reference models for specifying requirements. If this is so, the effort involved in developing requirements models such as ours from scratch will be reduced. This will help move many projects from being creative design to being typical design, and will facilitate the selection of commercial off-the-shelf (COTS) software. Further research into this domain is also necessary in education studies.

Acknowledgements

We wish to thank Dr Elsabé Cloete for her valuable contribution to earlier research leading to this paper. Also thank you to Prof. Remzi Seker who made valuable comments during the development of the requirements elicitation procedure. The work reported on in this article is based partly on work sponsored by a grant from the National Research Foundation of South Africa under Grant Number GUN: 2053850.

Appendix A

Table A1: Activities in a requirements ELICITATION procedure

Reference Number	Reference	Reference Number	Reference
1	(Macaulay 1996)	2	(Madhavji, Holtje et al. 1994)
3	(Goodrich and Olfman 1990)	4	(Eman and Madhavji 1995)
5	(Dawson 1991)	6	(Cordes and Carver 1989)

7	(Davis 1993)	8	(Zagorsky 1990)
9	(Basili and Weiss 1981)	10	(Farbey 1990)
11	(Kotonya and Sommerville 1995)	12	(Nuseibeh and Easterbrook 2000)
13	(Maiden and Rugg 1996)	14	(Johnson 1992)
15	(Schneider and Winters 1998)	16	(Jarke and Kurki-Suonio 1998)
17	(Macaulay, Fowler et al. 1990)	18	(Nuseibeh and Robertson 1997)
19	(Sommerville and Sawyer 1997)	20	(Viller and Sommerville 1999)
21	(Loucopoulos and Kavakli 1995)	22	(Yu 1997)
23	(Greenspan and Feblowitz 1993)	24	(Dardenne, Lamsweerde et al. 1993)
25	(Lamsweerde 2000)	26	(Young 2002)

Table A2: Characteristics

Phase	Sub-phase	Characteristic	No of Refs	References
All phases	Automated support	Provide **automated support** for the requirements engineering process	8	1 2 8 10 11 17 19 26
	Standards	Provide **standardised ways of describing** work products	6	1 2 11 19 26
		The **precision** of definition of its **notation**	2	11 19
		Process model standards	7	4 6 7 12 19 20 25
	Appropriate techniques	Select appropriate technique for the **problem domain**	6	2 12 13 19 26
		Use of **use cases** to describe related **tasks**	4	15 16 19 26
		Support a **systematic step-by-step approach**	3	1 19 26
		Solutions can easily be modified and are iterative	3	2 17 26
	Documentation	Support **documentation of requirements**	4	1 10 19 26
	Maintenance	Provide procedures for **maintaining** work products	1	1
	Conflict	**Conflict negotiation**	1	19
Feasibility	Goal Description	Define the **goal** of the modelling	4	3 22 24 25
	Management involvement	**Management consent** with solution	2	2 11
		Management attitude towards **change**	2	9 19
	Feasibility	Support **feasibility studies**	6	1 2 4 11 17 19
		Predictions about the system	1	3
		Scope for integration with existing systems	1	11
		Scope for **evolution**	1	11
	Cost-benefit	Do **cost-benefit analysis** of options	8	1 4 5 6 7 8 10 17
Elicitation	Requirements Specification	**Requirement completeness**	5	3 4 18 19 26
		Requirement relevance	4	2 3 18 26
		Expectations during specification of requirements	4	3 4 21 26
		Correctness	4	6 11 17 18
		Communication during specification of requirements	3	3 11 26
		Requirement accuracy	2	3 26
		Importance of **necessity**: requirements document	2	6 26
		Level of control over specifying requirements	1	3
	Constraints / Boundaries	Specify **constraints / boundaries**	5	2 11 12 19 26
	Problem analysis	Support **analysis**	7	1 2 11 12 19 25 26
		Degree of understanding of the **task and process**	3	2 3 14
	Use data gathering techniques	Support **data gathering techniques**	4	2 12 19 26
	Client involvement	Support **customer/client involvement**	2	3 26
Modelling	Motivation for modelling	**Support modelling**	11	1 2 3 7 11 12 17 19 20 25 26
	Goal modelling	Model the **purpose** by describing **behaviour**	2	21 26
	User involvement	Reflect the **needs of customers / users**	5	2 4 11 17 26
	Model environment	Model **business rules**	3	23 25 26
		Support **modelling of workflows**	3	2 23 25
		Clarity of business process	2	4 18
		Model system **services**	2	23 25
		Systems **architecture modelling**	1	19
Triage		Support **articulation / coherence** of the product concept	3	1 12 18
Verification	Id of Measurement tools	Provide ways of **assessing** the quality of work products	6	1 2 9 17 18 26
		Enable **identification of measures** of the requirements engineering process	4	1 2 18 26
		Support descriptions of **product effectiveness**	2	1 4
	Measures	**Quality** of the **product**	1	4
		Process effectiveness	1	5
		Cycle time	1	5
		Trace-ability	1	18
		User/customer satisfaction	5	2 4 9 11 17
		Requirements maturity (number of changes made	3	4 9 10

Bibliography

Basili, V & D Weiss 1981. Evaluation of a Software Requirements Document by analysis of Change Data. *SEKE93: Fifth International Conference on Software Engineering*. San Fransisco Bay, USA.

Bates, AW 2000. *Managing Technological Change, Strategies for College and University Leaders*. San Francisco: Jossey-Bass Publishers.

Bruno, K, B Vrana & L Welz 1998. *Practical Process Engineering for Higher Education. CAUSE98*. Seattle, Washington: CAUSE.

Cloete, E, A Van der Merwe & L Pretorius 2003. A Process Modelling Approach to Requirements Elicitation to Incorporate e-Learning as a Core Learning Strategy. *Seventh World Conference on Integrated Design and Process Technology (IDPT 2003)*. Austin, Texas.

Cordes, D & D Carver 1989. Evaluation Method for User Requirements Documents. *Information and Software Technology* 31(4): 181-188.

Curtis, B, MI Kellner & J Over 1992. Process Modeling. *Communication of the ACM* 35(9): 75-90.

Dardenne, A, A Lamsweerde & S Fickas 1993. Goal-Directed Requirements Acquisition. *Science of Computer Programming* 20: 3-50.

Davis, A 1993. *Software Requirements: Objects, Functions and States*. Englewood Cliffs, NJ: Prentice Hall.

Dawson, J 1991. Toronto Laboratory Requirements Process Reference Guide. Technical Report (Unpublished). Toronto, Canada, IBM Canada Ltd. Laboratory.

Educause 2003. The 2003 *National Survey of Information Technology in US Higher Education*. Washington, DC: Educause.

Eman, KE & NH Madhavji 1995. Requirements Engineering Processes. *International Symposium on Requirements Engineering*. York, England.

Eriksson, H-E & M Penker 2000. *Business Modeling with UML*. New York. John Wiley & Sons.

Farbey, B 1990. Software Quality Metrics: Considerations about Requirements and Requirements Specifications. *Information and Software Technology* 21(1): 60-64.

Finkelstein, A, M Ryan & G Spanoudakis 1996. Software Package Requirements and Procurement. *8th International Workshop on Software Specification and Design*. Schloss Velen, Germany.

Goodrich, V & L Olfman 1990. An Experimental Evaluation of Task and Methodology Variables for Requirements Definition Phase Success. *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences*. Hawaii. IEEE Computer Society.

Greenspan, S & M Feblowitz 1993. Requirements Engineering Using the SOS Paradigm. *1st International SYmposium on Requirements Engineering (RE'93)*. San Diego, USA, 4-6 January 1993.

Hickey, AM & AM Davis 2003. Requirements Elicitations and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes. 36th Hawaii International Conference on System Sciences. Hawaii. IEEE.

IEEE 1998. IEEE Guide for Developing System Requirements Specifications. New York, USA.

Jarke, M & R Kurki-Suonio 1998. Guest Editorial - Special Issue Scenario Management. *IEEE Transactions on Software Engineering* 24(12).

Johnson, P, 1992. Human-Computer Interaction: psychology, Task Analysis and Software Engineering. Maidenhead: McGraw-Hill.

Kotonya, G & I Sommerville 1995. *Requirements Engineering with Viewpoints*. Lancaster, UK: Cooperative Systems Engineering Group: CSEG/10/1995.

Lamsweerde, AV 2000. Requirements Engineering in the Year 00: A Research Perspective. *International Conference on Software Engineering*. Lumerick, Ireland.

Laurillard, D 1993. *Rethinking University Teaching*. London: Routledge.

Loucopoulos, P & E Kavakli 1995. Enterprise Modelling and the Teleological Approach to Requirements Engineering. *International Journal of Intelligent and Cooperative Information Systems* 4(1): 45-79.

Luker, MA 2000. *Preparing Your Campus for a Networked Future*. San Fransisco. Jossey-Bass Inc.

Macaulay, L 1996. *Requirements Engineering*. Great Britain: Springer-Verlag.

Macaulay, L, C Fowler, M Kirby & A Hutt 1990. USTM: A New Approach to Requirements Specification. *Interacting with Computers* 2(1): 92-118.

Madhavji, N, D Holtje, W Hong & T Bruckhaus 1994. Elicit: A Method for Eliciting Process Models. *3rd International Conference on Software Process*. Reston, Virginia.

Maiden, N & G Rugg 1996. ACRE: Selecting Methods for Requirements Acquisition. *Software Engineering Journal* 11(3): 183-192.

Maiden, N A M & C Ncube 1998. Acquiring Requirements for Commercial Off-The-Shelf Package Selection. *IEEE Software* 15(2): 45-56.

McDermid, J 1993. Software Engineer's Reference Book. Oxford. Butterworth-Heinemann.

Nuseibeh, B & S Easterbrook, 2000. Requirements Engineering: A Roadmap. *International Conference on Software Engineering*. Lumerick, Ireland.

Nuseibeh, B & S Robertson 1997. Making Requirements Measurable. *International Conference on Software Engineering* (ICSE 1997). Boston, MA USA.

Pressman, RS 2000. *Software Engineering, a Practitioner's Approach*. United Kingdom: McGraw-Hill International.

Ryan, S, B Scott, H Freeman & D Patel 2000. *The Virtual University - The Internet and Resource-Based Learning*. London: Kogan Page.

Rzepka, WE 1989. A Requirements Engineering Testbed: Concept Status, and First Results. *Twenty-Second Annual Hawaii International Conference on System Sciences*. Hawaii. IEEE Computer Society.

Schneider, G & J Winters, 1998. Applying Use Cases: a practical guide. Boston, MA: Addison-Wesley.

Sommerville, I & P Sawyer 1997. *Requirements Engineering: A Good Practice Guide*. UK: John Wiley & Sons.

Tait, F 1999. Enterprise Process Engineering: A template tailored for higher education. *CAUSE/EFFECT* 22(1).

Viller, S & I Sommerville 1999. Social Analysis in the Requirements Engineering Process: from Ethnography to Method. *4th International Symposium on Requirements Engineering*. Limerick, Ireland, 7-11th June 1999.

Whitten, JL, L D Bentley & K C Dittman 2001. *System Analysis and Design Methods*. New York: McGraw-Hill Higher Education.

Young, R, 2002. Recommended Requirements Gathering Practices. *Crosstalk: The Journal of Defense Software Engineering* April 2002: 9-12.

Yu, E, 1997. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *3rd IEEE International Symposium on Requirements Engineering (RE'97)*. Annapolis, USA, 6-10 January 1997.

Zagorsky, C, 1990. Case Study: Managing the Change to CASE. *Journal of Information Systems Management*. September 1990: 24-32.

Authors' Contact Details

Alta van der Merwe (vdmeraj@unisa.ac.za)
Paula Kotzé (kotzep@unisa.ac.za)
School of Computing, University of South Africa
Johannes Cronjé (jcronje@up.ac.za)
Department of Teaching and Training Studies, University of Pretoria, Pretoria, South Africa

Using Cluster Analysis to Investigate the Role of Culture in the Adoption of Web-Based Learning Tools

Irwin Brown
Belinda Knol
Simon Vincent

Introduction

Web-based learning technologies are now more or less pervasive in higher education institutions, but in many instances they are not used to their full potential. There may be many reasons for this, including inadequate technology skills on the part of lecturers, increased workload as a result of implementing such tools, and difficulties with integrating the technology into traditional teaching approaches (Singh & Blewett, 2004). Many studies have examined the issues of adoption from the perspective of the academic staff, with fewer examining the impact of student perceptions on acceptance (Grandon *et al*., 2005). In this study, a student-centred perspective is therefore taken, whereby students' perceptions of web-based learning technology are solicited, and how these influence usage is determined. The impact of cultural values on this process is furthermore elucidated.

To assess the factors influencing usage, the well-known technology acceptance model (TAM) is employed (Lee *et al*. 2003), whilst to assess culture the equally well-known Hofstede (1980) dimensions are used. The effect of culture is examined by employing a statistical technique known as cluster analysis (Segars & Grover, 1999). Cluster analysis allows groups (in this case cultural groups) to be formed on

*Alter*nation 12.1b (2005) 515-542 *ISSN 1023-175*

the basis of multiple criteria (the cultural dimensions), and in the process differences within groups are minimised whilst differences between groups are maximised.

In the next section, web-based learning technologies, technology adoption, and culture are discussed. This leads to the development of hypotheses, before the research procedure is outlined. The data is then analysed, and the results reported. Following this is a discussion on the implications of the findings, before ideas for future research are suggested and the paper concluded.

Conceptual Background

The theoretical foundations of this study are web-based learning technologies, technology adoption (TAM specifically), and culture, respectively. This section will therefore briefly examine past literature and research in these areas, as well as their relevance to the South African context.

Web-Based Learning Technologies

There exists a wide variety of web-based learning tools that make use of Internet technologies to enable and enhance the teaching and learning process (Mioduser *et al.*, 2000). One such tool is WebCT (Web Course Tools). This is a course management web-based tool that was developed for academic purposes at the University of British Columbia, and is now used at educational institutions across the globe (Wernet *et al.*, 2000). The tools enable educators to develop and maintain a structured web site that can be used for an entirely online web-based course or to supplement classroom-based activities (McClelland, 2001). A typical WebCT site in general may have four main modules:

- Course Material module (*e.g.*, course syllabus template, course contents)
- Communication Tools (e.g., bulletin board, chat room, email, and whiteboard)

- Evaluation Tools (e.g. quizz, self-tests, assignments, results)
- Study Tools (e.g. student web pages, presentations)

This tool could be particularly useful in institutions with highly diverse student bodies, such as are found in South Africa. For example, schools in formerly disadvantaged areas are generally crowded, and under-resourced, and may not prepare students sufficiently for tertiary-level studies (Hall, 2001). Students coming from these backgrounds may therefore face enormous challenges when entering universities that have previously catered for students from better-resourced, advantaged schools (Brown, 2002). Educators are faced with the dilemma of having students from both advantaged and disadvantaged backgrounds in the same class, with vastly different prior knowledge and experiences. Web-based learning technologies offer opportunities for managing such diversity, as their flexibility allows for students to work at their own pace, and use the technology in a manner consistent with their learning styles and prior knowledge and experience (Lanham & Zhou, 2003).

Technology Adoption

In assessing technology adoption the popular technology acceptance model (TAM) is often employed (Lee *et al.*, 2003). This predicts that usage of a technology is influenced in the main by two interrelated variables – perceived usefulness, and perceived ease of use respectively. To take into account the nature of the learning environment, an additional variable can be included – that of perceived voluntariness (Brown, 2003). This is defined as the extent to which users perceive the adoption decision to be voluntary (Agarwal & Prasad 1997). The rationale for including this variable is that very often the attitude of the lecturer or instructor determines the extent to which a technology is used. If it is mandated for the course, or if it is perceived to be, then for certain individuals, they will more likely use it, sometimes even if they perceive it to be not useful.

Cultural Values and Measurement Issues

The subject of national culture and its impact on technology adoption has received fairly widespread research attention in information systems (IS) (McCoy *et al.*, 2005). Much of this interest has been fuelled by the diffusion of information technology (IT) into countries having very different cultures to that from where they were first launched. Thus, conventional models of adoption, such as the TAM (Lee *et al.*, 2003), developed and tested in USA, have been questioned as to their utility in other cultures. Straub *et al.* (1997), for example, found the model to be suitable for explaining adoption of email in the USA and Switzerland, but not in Japan. In the context of learning, Grandon *et al.* (2005) found there to be differences too between American and Korean students with regards to the factors influencing adoption of online classes.

National culture has very often been defined in terms of Hofstede's (1980) dimensions. He defined culture as being "*the collective programming of the mind which distinguishes the members of one group or category of people from another*" (Hofstede, 1991, p. 5), and operationalised it along four dimensions:

- *Power-distance* – Degree of inequality among people, which the population of a culture considers normal.
- *Uncertainty avoidance* - Degree to which people in a culture feel uncomfortable with uncertainty and ambiguity.
- *Individualism* - Degree to which people in a culture prefer to act as individuals, rather than as members of groups.
- *Masculinity*- Degree to which values like assertiveness, performance, success and competition prevail among people of a culture over gentler qualities like the quality of life, maintaining warm personal relationships, service, care for the weak, etc.

Hofstede later added a fifth dimension – long-term orientation (Hofstede, 1991). However, many studies still use the original 4 dimensions when examining national culture. Many of the studies on culture and technology adoption work from this basis – i.e., they take Hofstede's (1980) findings to be reflective of current reality (McCoy *et al.*, 2005). Several problems have been noted with this approach.

Firstly, Hofstede conducted his study of national culture about two decades ago. Thus, no recognition is taken of the dynamic nature of culture. His measures may not accurately reflect the current reality (McCoy *et al.*, 2005). Secondly, his sample for the study were IBM employees, and thus may not be reflective of the demographics of a country. South Africa is a case in point. The cultural profile of South Africa as captured by Hofstede (1980) reflected that of IBM employees and managers at that time – mainly White. Thus generalisation of his profile to the nation at large is problematic. Hofstede (1998) responds to this critique (although not referring to South Africa specifically) by asserting that differences between countries in terms of values remain more or less stable over time, and that in order to compare across nations, the sample group must be similar, in as many ways as possible. Only then is it possible to compare groups across nations, and focus only on the differences between national cultures.

Some researchers have attempted to overcome these problems by including the Hofstede measuring instrument in their studies. However, in several of these cases, it has been found that the measures do not exhibit statistical validity and reliability (Spector *et al.*, 2001). Concerning lack of validity and reliability, Hofstede (2002) responds that his measures were designed to assess the values of a multitude, and not an individual. Thus, standard statistical tests for reliability and validity are not always appropriate.

McCoy *et al.*, (2005) contend that "the assumption of homogeneity is not appropriate, particularly if the national culture construct are to be integrated into IS models that reflect individual behaviour..." (p. 214). They therefore argue for assessing the cultural values of individuals rather than a multitude, in recognition of the fact that people from the same nation or ethnic group may have different values (Srite *et al.*, 2003).

In the South African context, whilst apartheid sought to deliberately keep ethnic groups separate, with the new South Africa, freedom of association is guaranteed. Thus cultural values will not be entirely based on ethnicity or race. Indeed in a recent study, Thomas & Bendixen (2000) found there to be little difference between ethnic groups in South Africa in terms of Hofstede's cultural dimensions. Ethnicity as a surrogate for cultural values also has other connotations. For example, socio-economic differences between ethnic groups at a macro level is still very much a reality in South Africa (StatsSA, 2003), and may also explain differences found in technology adoption between ethnic groups (Brown & Licker, 2003).

This article reports on a study which employed Hofstede's measures to assess cultural values of students, and in so doing investigate what impact these have on the adoption process for web-based learning tools. Thus, unlike with other culture studies, which had national profiles or ethnic groups as the unit of analysis, the focus here was on groups of students, regardless of race or ethnicity.

Initial study findings were reported in Brown *et al.* (2003). However, in that analysis each dimension of culture was analysed separately for its effect on technology adoption. The intent of Hofstede was for culture to be described by the profile across all four dimensions (5 if long-term orientation is included). In this article, cluster analysis was therefore employed to re-analyse the data and generate profiles (Segars & Grover, 1999). Groups were thus formed such that within groups, differences across a set of criteria were minimised, whilst between groups differences were maximised. The criteria in this instance were the four original cultural dimensions proposed by Hofstede (1980).

Development of Hypotheses

The WebCT technology investigated in this study consisted of four main modules. Usage behaviour for each module is affected differently by cultural values, therefore only a single module was selected for further analysis. Preliminary data analysis showed that the evaluation module was used to the greatest extent, and so was the focus of further attention. The effects of cultural values (uncertainty avoidance, masculinity, power-distance and individualism) on the strength of relationship between perceived ease of use and usage, perceived usefulness and usage, and perceived voluntariness and usage were investigated, leading to the hypotheses outlined next.

Anandarajan *et al.* (2002) posit that where there is high uncertainty avoidance, usage of a technology will be significantly influenced by its perceived ease of use, as this attribute reduces ambiguity of use. Thus, the hypothesis is:

- H1A: The influence of perceived ease of use on usage of web-based learning evaluation tools is greater for those with high uncertainty avoidance than those with low uncertainty avoidance.

Where there are high levels of uncertainty, a strong motive exists to want to reduce it amongst those with high uncertainty avoidance traits (Straub *et al.*, 1997). Any technology that is perceived as supporting this goal will be perceived as useful, and may subsequently be used quite extensively. The online evaluation module in WebCT provides facilities for learners to reduce uncertainty in performance by providing quick feedback on quizzes, *etc.* The hypothesis therefore is:

- H1B: The influence of perceived usefulness on usage of web-based learning evaluation tools is greater for those with high uncertainty avoidance than those with low uncertainty avoidance.

High uncertainty avoidance cultures have been shown to prefer clear written rules and regulations concerning matters (Milberg *et al.*, 1995) and are more likely to comply with these than low uncertainty avoidance cultures. Thus, it follows that those who score high on uncertainty avoidance will be more likely to use a technology if it is mandated for a course, as they would not want to take the risk of not following the advice and requirements. The hypothesis is:

- H1C: The influence of perceived voluntariness (inverted) on usage of web-based learning evaluation tools is greater for those with high uncertainty avoidance than those with low uncertainty avoidance.

For those who score low on masculinity (high on femininity), usage of the tool will be influenced more by the comfort and ease of use of a technology, than for those who score high on masculinity (Hofstede, 1980). The hypothesis is therefore:

- H2A: The influence of perceived ease of use on usage of web-based learning evaluation tools is greater for those with low masculinity than those with high masculinity.

For high masculinity individuals the online evaluation tools offer an opportunity to assess performance, success, and competitiveness, all of which are of value to them (Hofstede, 1980). Thus, evaluation tools will be seen as useful, which in turn will motivate their usage to a greater extent than those lower in masculinity. The hypothesis is therefore:

- H2B: The influence of perceived usefulness on usage of web-based learning evaluation tools is greater for those with high masculinity than those with low masculinity.

Those with high masculinity traits have a greater focus on task accomplishment (Hofstede, 1980). Thus, they will more likely use a tool, if it is perceived as mandatory and required for completing a task. Those who score low in masculinity will, however, also be

more likely to use a technology if it is perceived as mandatory, but the motive in this case will be due to social pressure, to which they are more susceptible (Hofstede, 1994). Thus the hypothesis:

- H2C: The influence of perceived voluntariness (inverted) on usage of web-based learning evaluation tools does not differ between those with high masculinity, and those with low masculinity.

Those with high individualism scores will more likely perceive evaluation tools as being easy to use, as the tools are compatible with their preference for working independently rather than collectively (Veiga *et al.*, 2001). The hypothesis supported is:

- H3A: The influence of perceived ease of use on usage of web-based learning evaluation tools is greater for those with high individualism than those with low individualism.

Those scoring high on individualism will perceive evaluation tools to be more useful than those who are collectivist, as it is more compatible with their style of working independently (Veiga *et al.*, 2001).

- H3B: The influence of perceived usefulness on usage of web-based learning evaluation tools is greater for those with high individualism than those with low individualism.

Those who score low on individualism (highly collectivist) will be more influenced by social pressures from peers and superiors in usage of tools, as they are more conformity-oriented (Steenkamp *et al.*, 1999), thus the hypotheses is:

- H3C: The influence of perceived voluntariness (inverted) on usage of web-based learning technologies is greater for those with low individualism than those with high individualism.

In high power-distance cultures, "*subordinates defer to superiors and do not question their authority*" (Lim, 2004, p. 32). In the learning context, students thus accept the lecturer authority almost without question. Usage of a technology for those high in power-distance cultures will be driven to a great extent by whether the supervisor/lecturer encourages or mandates usage. The hypothesis supported is:

- H4: The influence of perceived voluntariness (inverted) on usage of web-based learning evaluation tools is greater for those with high power-distance than those with low power-distance.

Research Procedure

In this study cluster analysis is to be used to generate groups of students having similar cultural profiles, based on the dimensions of masculinity, individualism, uncertainty avoidance and power-distance. Differences in technology adoption between the groups will then be examined by drawing from the above hypotheses. The approach to the research is positivistic, hypothetico-deductive and quantitative.

Questionnaire Design

The questionnaire consisted of three sections. In the first section demographic information regarding the respondents' degree program, year of study, home language, race, family income, gender and age were gathered. In the second section respondents' cultural values were assessed using a 20-item abridged and modified version of Hofstede's (1980) original measures taken from Hepburn *et al.* (2000). The wording of the 20 items were modified so as to be understood by undergraduate students. For each cultural value, participants were asked to rate its importance to them as individuals on a scale of 1 to 5. The third section of the questionnaire related to respondents' perceptions and usage of WebCT. These were all assessed using a 7-point Lickert scale, anchored by Strongly disagree at one end to Strongly agree at the other. Perceived Usefulness was operationalised with 5 items identified from Davis (1989) and Teo *et al.*

(1999), and modified for the particular context. Perceived Ease of Use, too, was measured with 4 items modified from Teo *et al.* (1999). Perceived voluntariness was operationalised with 3 items derived from Agarwal & Prasad (1997). Finally, usage of WebCT was assessed according to respondents' self-assessment of the extent to which they used the standard features available in each of the four modules of a typical WebCT site. For the Course content module, there were 6 items, for the Communications module 4 items, the Evaluation tools, 4 items, and the Study module, 4 items. Each item was assessed on a scale of 1 to 7, anchored by Never used at one end to a Great Extent at the other.

A pilot questionnaire was handed out to ten students who were asked to fill in the questionnaire and make suggestions where necessary. Problem areas were identified and questions reworded to improve understanding. See Appendix 1 for the final item measures used.

Data Collection Procedure

A list of courses that make use of WebCT was obtained from the WebCT systems administrator at the University of Cape Town. Emails were sent to a number of course lecturers requesting permission to distribute questionnaires during their lectures. Arrangements were finalised for four courses, although respondents from any other course that used WebCT were not excluded from participating. The four courses were:

- A first year Statistics course, in which the WebCT Evaluation tools were mainly used.
- A first year Information Systems course, in which WebCT was made available as a learning tool. Usage was encouraged, but not made compulsory.
- A first year Biology course, in which the WebCT evaluation tools were mainly used.

- A third year Economic course, in which Chat room and other communications tools were used extensively.

A total of 250 questionnaires were distributed. 178 questionnaires were returned. 32 questionnaires were returned incomplete and were therefore discarded from any further analysis. This resulted in 146 responses that were usable.

Demographic Profile of Respondents

The majority (77%) of students were in first year and thus under 21 (88%). Most were studying Business (BCom and BBusSci) (71%), with the remainder mainly studying for a BSc (27%). There was an even gender mix, and a majority of Black students (40%), followed by White (37%), Coloured (12%), then Indian and Asian (6%) students. In essence, therefore, the spectrum of South African race-based cultures were represented.

The demographic profile of respondents is shown in Table 1 below.

Table 1: Demographic Profile

Variable	Items	Frequency	Percentage
Year of Study	First	113	77.4%
	Second	17	11.6%
	Third	11	7.5%
	Fourth	5	3.4%
Degree	BCom	60	41.1%
	BBusSci	44	30.1%
	BSocSci	2	1.4%
	BSc	40	27.4%
Gender	Male	73	50%
	Female	73	50%
Age	Under 21	128	88%
	21 to 30	18	12%
Race	Black	59	40.4%
	White	54	37%
	Coloured	17	11.6%
	Indian	7	4.8%
	Asian	2	1.4%
	Did not answer	7	4.8%

Cluster Analysis

Cluster analysis was employed, in order to generate groups (profiles) based on individualism, masculinity, uncertainty avoidance and power distance collectively. This resulted in two groups having the cultural profiles shown in Figure 1 below.

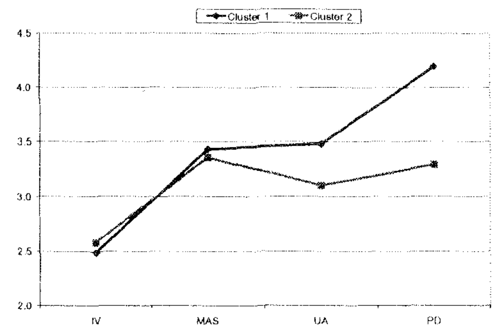


Figure 1: Cultural Profiles arising from Cluster Analysis

Cluster 1 was much higher in power-distance (PD) and uncertainty avoidance (UA) than Cluster 2, with minimal difference on the individualism (IV) and masculinity (MAS) dimensions.

T-tests were used to further examine differences between the clusters, as shown in Table 2. There were significant differences on the uncertainty avoidance, power-distance, and individualism dimensions, but not the masculinity dimension. The difference on the individualism dimension was furthermore minimal. There were no significant differences in terms of perceptions of WebCT and usage of evaluation tools, indicating minimal effect of culture on these variables.

Table 2: t-tests for Clusters

	Cluster 1 Mean	Cluster 2 Mean	t-value	p	Cluster 1 Size	Cluster 2 Size
Cultural Values						
Power-Distance	***4.2***	***3.3***	***-13.7***	***0.0000***	***76***	***70***
Uncertainty Avoidance	***3.5***	***3.1***	***-4.5***	***0.0000***	***76***	***70***
Individualism	***2.5***	***2.6***	***2.0***	***0.0507***	***76***	***70***
Masculinity	3.4	3.4	-1.7	0.1003	76	70
Perceptions of WebCT						
Perceived Ease of Use	5.3	5.4	0.6	0.5217	75	70
Perceived Usefulness	4.7	4.4	-1.0	0.3020	76	70
Perceived Voluntariness	2.8	2.7	-0.3	0.7937	76	70
WebCT Usage						
Evaluation Tools Usage	3.9	4.4	1.8	0.0795	76	70

In order to examine the impact of the cultural profiles on the relationships between variables, two separate regression models can be developed for each cluster, and the regression coefficients compared. This is similar to the technique employed by Venkatesh & Morris (2000) when testing the effect of gender on technology adoption. Before developing regression models, both sets of data must exhibit validity and reliability. The data from the 76 respondents belonging to Cluster 1 were therefore subjected to validity and reliability tests, and then separately the data from the 70 respondents in Cluster 2.

Validity Tests

To test validity, factor analysis with varimax rotation was employed, with eigenvalue set to 1. If items load at greater than 0.5 on their own factor, and less than 0.4 on all other factors validity is demonstrated (*e.g.* see Tan & Teo, 2000).

For the both clusters, one of the perceived usefulness items (PU3 – I find WebCT to be useful), cross-loaded on the ease of use factor, and so was dropped. The factor analysis was repeated without this item, resulting in validity being demonstrated in both cases – *i.e.*,

items for perceived ease of use, perceived usefulness, and perceived voluntariness all loaded as separate factors (see Table 4a and 4b)

Table 4a: Factor Analysis for Cluster 1

	Ease of Use	Usefulness	Voluntariness
EOU1	0.82	0.11	0.00
EOU2	0.87	0.09	0.08
EOU3	0.89	0.07	0.13
EOU4	0.87	0.07	0.04
PU1	0.11	0.85	-0.05
PU2	0.07	0.87	-0.02
PU4	0.12	0.85	0.06
PU5	0.05	0.92	-0.09
Vol1	-0.02	0.01	-0.86
Vol2	-0.12	0.03	-0.89
Vol3	-0.08	0.05	-0.89

Table 4b: Factor Analysis for Cluster 2

	Ease of Use	Usefulness	Voluntariness
EOU1	0.79	0.21	0.02
EOU2	0.87	0.26	-0.14
EOU3	0.74	0.18	-0.02
EOU4	0.83	0.23	0.05
PU1	0.28	0.87	-0.03
PU2	0.25	0.88	-0.05
PU4	0.16	0.92	-0.04
PU5	0.39	0.70	-0.01
Vol1	-0.01	0.27	0.73

Vol2	-0.02	-0.11	0.91
Vol3	-0.02	-0.25	0.84

Reliability Tests

To test for reliability, the two sets of data were assessed for their Cronbach alpha. If for each construct, the alpha is greater than 0.7, then reliability is proven (e.g., see Teo *et al.*, 1999). Table 5 shows the results of reliability analysis for each cluster.

Table 5: Reliability Analysis

	Cluster 1	Cluster 2
Perceived Ease of Use	0.89	0.86
Perceived Usefulness	0.90	0.92
Perceived Voluntariness	0.85	0.77
Evaluation Tool Usage	0.77	0.89

Multiple Linear Regression

Multiple linear regression was now possible for each of the clusters, where the three independent variables, perceived usefulness, perceived ease of use, and perceived voluntariness were regressed on to evaluation tool usage. The results of this test are shown in Table 6. For Cluster 1, only perceived mandatoriness (the inverse of voluntariness) was a significant influence on usage, whereas for Cluster 2 it was perceived usefulness and perceived mandatoriness. The values of coefficients for perceived ease of use, and perceived voluntariness in Cluster 1 were furthermore greater than for Cluster 2.

Table 6: Multiple Linear Regression with Evaluation Tool Usage as Dependent Variable (**p < 0.01; *p < 0.05)

Independent Variable	Cluster 1	Cluster 2	Relevant Hypotheses
Perceived Ease of Use	0.12	-0.01	H1A
Perceived Usefulness	0.21	*0.27	H1B (H2B)
Perceived Voluntariness	**-0.43	*-0.23	H1C, H4

In order to interpret these findings the hypotheses previously generated can be revisited. Of specific interest are the hypotheses concerning uncertainty avoidance (H1A, H1B, H1C) and power distance (H4), since there were major differences between the clusters based on these two dimensions.

For Cluster 1, which is higher in uncertainty avoidance and power distance than Cluster 2, it is expected from H1A that perceived ease of use will be a greater influence; from H1B that perceived usefulness will be a greater influence, and from H1C that perceived voluntariness will be a greater influence. From H4, it is expected again that perceived voluntariness would be a greater influence. The regression coefficients in Table 6 confirm the hypotheses H1A, H1C and H4, but not H1B.

The reasons for hypothesis H1B not being supported may be because Cluster 2 exhibited a small but significantly higher score for individualism, than Cluster 1. The WebCT tool allows students to work more or less independently. Thus perceived usefulness becomes a more salient factor for Cluster 2 (higher individualism) than cluster 1, based on hypothesis H2B. This influence may be stronger than that caused by high uncertainty avoidance (H1B).

Discussion and Implications

The findings point to the need for there to be a balance when introducing web-based learning technologies into a classroom. The technology in itself is not an end, and its introduction must be considered within the social context of student learning. Thus student cultural values, as well as their perceptions of the technology must be taken into account, as all of these ultimately impact on levels of usage.

In the South African context, where at the tertiary level, there is student diversity in terms of culture, socio-economic background, and levels of preparedness for higher education (Hall, 2001), these are issues that cannot be ignored. Lecturers and instructors must be conscious of the differing expectations and interpretations of students, and how these will impact on issues such as usage of learning technology. So, for example, this study has shown there to be two major groups – one high in power-distance and uncertainty avoidance, and the other lower on these dimensions. Each group is motivated by different factors. In terms of using learning technology, the one is motivated by perceived mandatoriness (the opposite of voluntariness), the other by perceived usefulness, as well as mandatoriness. Thus, to ensure both groups are catered for, a useful strategy would be to ensure that the web-based learning tools are perceived as useful, by making sure the tools are aligned, integrated, and relevant to the course of interest. Instead of merely encouraging use, usage could be made a requirement by, for example, including as part of the course assessment, a test or tests that must be completed through the web-based evaluation tools. Since perceived ease of use is not a factor for everyone, an optional training session could be provided for those who feel they need it before using the tools independently.

From another perspective, lecturers may want to develop in students the ability to work independently and think critically. Those who are using the tool primarily because they see it as a requirement, or because it is easy to use, may have to develop a more critical mindset, whereby they can independently assess and evaluate the tool, in terms of the value it adds. If it is perceived as not contributing to learning, they ought to challenge the lecturer/instructor on

this. In that way it can be ascertained whether they have critically evaluated the tool. It may be that they have not fully explored its features, or that the functionality has been difficult to use, in which case these issues can be addressed first.

Limitations and Future Research

The study has been limited to examining the impacts of three variables on usage of web-based evaluation tools – perceived usefulness, perceived ease of use, and perceived voluntariness of WebCT respectively. The perceptions measured were of WebCT in general. WebCT, however, consists of several different modules, inclusive of evaluation tools. Future research ought to be more specific, and investigate both perceptions and usage with regards to a particular tool or tools within WebCT.

It is perhaps appropriate that qualitative, interpretive studies be now conducted in this area to get a richer understanding of student perceptions, attitudes, and experiences with these technologies. Alternatively, a mixture of both quantitative and qualitative techniques can be employed, whereby a study such as this one can initially be carried out, and then follow up interviews conducted with a set of students who responded to the questionnaire.

The subject of culture is both interesting and controversial, with Hofstede's (1980) study attracting widespread attention and critique (McCoy *et al.*, 2005). It is thus a fertile area for research in the context of education and technology use.

Conclusion

Web-based learning technologies are used in a diverse array of courses in South African tertiary institutions. Much research has focused on the technologies and their capabilities. However, for these to make an impact on learning, it is important that they be firstly accepted and used by students. This research, therefore, took a student-centred view, and examined their perceptions and usage, as well as the effect of social factors such as cultural values.

Culture was examined by assessing the importance to students of four sets of cultural values – individualism, masculinity, power-distance, and uncertainty avoidance. Cultural groupings independent of ethnicity and language were then generated using the statistical technique of cluster analysis. This analysis generated two clusters (groups). The one cluster was higher in power-distance and uncertainty avoidance than the other, with little difference between the clusters on the masculinity and individualism dimensions.

Factors influencing usage of web-based learning technologies were then compared between the groups. Consistent with expectations, for the group higher in power-distance and uncertainty avoidance, usage was influenced mainly by whether such usage was perceived as being mandatory, and to a lesser extent by its perceived usefulness and ease of use. For the group lower in uncertainty avoidance and power-distance on the other hand, usage was motivated primarily by perceived usefulness, and to a lesser extent by whether its use was perceived as being mandatory, with perceived ease of use playing no part.

The findings point to the need for educators to understand not only the technology and its capabilities, but to also consider what might motivate student usage, and how these motivations may differ depending on cultural values. In other words, there is a need to view web-based learning tools not simply as technical innovations to improve educational practice, but more holistically as a socio-technical innovation that may have positive consequences for student learning.

References

Agarwal, R & J Prasad 1997. The Role of Innovation Characteristics, and Perceived Voluntariness in the Acceptance of Information Technologies. *Decision Sciences* 28, 3: 557-582.

Anandarajan, M, M Igbaria, & U Anakwe 2002. IT Acceptance in a Less-Developed Country: A Motivational Factor Perspective. *International Journal of Information Management* 22: 47 – 65.

Brown, I 2002. Individual and Technological Factors Affecting the Perceived Ease of Use of Web-based Learning Technologies in a Developing Country. *Electronic Journal of Information Systems in Developing Countries* 9, 5: 1 – 15. Accessed on 6th May, 2005 at http://www.ejisdc.org.

Brown, I 2003. The Role of Cognitive Instrumental Processes, Social Influence Processes and Perceived Behavioural Control in the Acceptance of the Internet as a Learning Tool. *Alternation* 10, 2: 89 - 121.

Brown, I & P Licker 2003. Exploring Differences in Internet Adoption and Usage between Historically Advantaged and Disadvantaged Groups in South Africa. *Journal of Global Information Technology Management* 6, 4: 6 - 26.

Brown, I, P Licker, B Knol & S Vincent 2003. Web-based Learning Tools: The Influence of Cultural Values on Usage Behaviour. *Proceedings of the 4th Annual Global Information Technology Management Association Conference* Calgary, June 8 – 10: 11 – 14.

Davis, F 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly* 13, 3: 319-340.

Grandon, EE, K Alshare & O Kwun 2005. Factors Influencing Student Intention to Adopt Online Classes: A Cross-Cultural Study. *Journal of Computing Sciences in Colleges* 20, 4: 46 – 56.

Hall, M 2001. Access to Higher Education: Race, Resources and Social Exclusion. *The Journal of South African and American Comparative Studies,* 5, Accessed on 6th March, 2002 at http://www.safundi.com/.

Hepburn, A, D Laing & K Mentor 2000. *The Influence of Culture and Time on the Effectiveness of GDSS: A Pilot Study.* University of Cape Town Unpublished Honours Empirical Research Report (Mentor: P Licker), 1 – 90.

Hofstede, G 1980. *Culture's Consequences, International Differences in Work-related Values*. Beverly Hills: Sage.

Hofstede, G 1991. *Cultures and Organisations: Software of the Mind.* London: McGraw-Hill.

Hofstede, G 1994. Management Scientists are Human. *Management Science* 40, 1: 4-13.

Hofstede, G 1998. A Case for Comparing Apples with Oranges. International differences in Values. *International Journal of Comparative Sociology* 39, 1: 16 – 31.

Hofstede, G 2002. The Pitfalls of Cross-National Survey Research: A Reply to the Article by Spector *et al.* On the Psychometric Properties of the Hofstede Values Survey Module 1994: A Comparison of Individual and Country/Province Level Results. *Applied Psychology: An International Review* 51, 1: 170 – 173.

Lanham, E & W Zhou 2003. Cultural Issues in Online Learning – Is Blended Learning a Possible Solution? *International Journal of Computer Processing of Oriental Languages* 16, 4: 275 – 292.

Lee, Y, KA Kozar & KR Larsen 2003. The Technology Acceptance Model: Past, Present and Future. *Communications of the Association for Information Systems* 12, 50: 752 – 780.

Lim, J 2004. The Role of Power Distance and Explanation Facility in Online Bargaining Utilizing Software Agents. *Journal of Global Information Management* 12, 2: 27 – 43.

McClelland, B 2001. Digital Learning and Teaching: Evaluation of Developments for Students in Higher Education. *European Journal of Engineering Education*, 26, 2: 107 – 115.

McCoy, S, DF Galletta & WR King 2005. Integrating National Culture into IS Research: The Need for Current Individual-Level Measures. *Communications of the Association for Information Systems* 15: 211 – 224.

Milberg, SJ, SJ Burke, HJ Smith & EA Kallman 1995. Values, Personal Information Privacy and Regulatory Approaches. *Communications of the ACM* 38, 12: 65 – 74.

Mioduser, D, R Nachmias, O Lahav, & A Oren 2000. Web-based Learning Environments: Current Pedagogical and Technological State. *Journal of Research on Computing in Education* 33, 1: 55-76.

Segars, A & V Grover 1999. Profiles of Strategic Information Systems Planning. *Information Systems Research* 10, 3: 199-232.

Singh, UG & CN Blewett 2004. Potholes on the Educational Superhighway – An Identification of the Factors Hindering the Usage of Educational Technology Tools by University IT Departments in South Africa, *Proceedings of the 34th Annual South African Computer Lecturers' Association (SACLA) Conference* Durban, Jul 5 – 6 [CD]: 820 – 831.

Spector, P, C Cooper & K Sparks 2001. An International Study of the Psychometric Properties of the Hofstede Values Survey Module 1994: A Comparison of Individual and Country/Province Level Results. *Applied Psychology: An International Review* 50, 2: 269 – 281.

Srite, M, D Straub, K Loch, KR Evaristo & E Karahanna 2003. Inquiry into Definitions of Culture in IT Studies. *Advanced Topics in Global Information Management.* Hershey: Idea Group Publishing.

StatsSA: Statistics South Africa 2003. *Census 2001, South Africa.* Accessed on 24th July, 2003

http://www.statssa.gov.za/SpecialProjects/Census2001/Census2001.htm.

Steenkamp, JE, F Hofstede & M Wedel 1999. A Cross-National Investigation into the Individual and National Cultural Antecedents of Consumer Innovativeness. *Journal of Marketing* 63: 55 – 69.

Straub, D, M Keil & W Brenner 1997. Testing the Technology Acceptance Model Across Cultures: A Three Country Study. *Information & Management* 33: 1 – 11.

Tan, M & T Teo 2000. Factors Influencing the Adoption of Internet Banking. *Journal of the Association for Information Systems* 1, 5:1-42.

Teo, T, V Lim & R Lai 1999. Intrinsic and Extrinsic Motivation in Internet Usage. *Omega, International Journal of Management Science* 27: 25-37.

Thomas, A & M Bendixen 2000. The Management Implications of Ethnicity in South Africa. *Journal of International Business Studies* 31, 3: 507 – 519.

Veiga, JF, S Floyd & K Dechant 2001. Towards Modelling the Effects of National Culture on IT Implementation and Acceptance. *Journal of Information Technology* 16: 145 – 158.

Venkatesh, V & M Morris 2000. Why Don't Men Ever Stop for Directions? Gender, Social Influence, and Their Role in Technology Acceptance and Usage Behaviour. *MIS Quarterly* 24, 1: 115 – 139.

Wernet, S, R Olliges & T Delicath 2000. Postcourse Evaluations of WebCT (Web Course Tools) Classes by Social Work Students. *Research on Social Work Practice* 10, 4: 487.

Appendix: Questionnaire Measures

Cultural Values Importance (Scale 1 - 5)

Individualism:

IV1. Having sufficient time for personal or family life.

IV2. Having considerable freedom to adopt one's own approach to any task.

IV3/M9. Having challenging things to do, from which to get a personal sense of accomplishment.

IV4 (-). Having good physical working conditions.

IV5(-)/M10. Fully using your skills and abilities on any task.

IV6 (-). Having training opportunities to improve your skills or to learn new skills.

Masculinity:

M1 (-). Living in a desirable district.

M2 (-). Working with people who co-operate well with one another.

M3 (-). Having a good working relationship with those in authority.

M4 (-). Having security of employment.

M5. Having an opportunity to earn large rewards.

M6. Getting the recognition for a task well done.

M7 (-). Being helpful to others.

M8. Having an opportunity to advance to high-level jobs.

M9/IV3. Having challenging things to do, from which to get a personal sense of accomplishment.

M10/IV5(-). Fully using your skills and abilities on any task.

Uncertainty Avoidance:

UA1. Not feeling nervous or tense when working.

UA2. Sticking with an employer for as long as possible and changing jobs only when absolutely necessary.

UA3. Keeping to the rules of a group – even when one thinks breaking them ought to be in the group's best interests.

Power-Distance:

PD1 (-). Having leaders who consult with everyone before making decisions.

PD2 (-). Not feeling afraid to express disagreement with those in authority.

PD3. Accepting that some people are more powerful than others.

Perceptions about WebCT (Scale 1 – 7)

Perceived Ease of Use:

EOU1. WebCT is easy to use.

EOU2. WebCT is easy to learn.

EOU3. WebCT is user friendly.

EOU4. WebCT is easy to master.

Perceived Usefulness:

PU1. Using WebCT would improve my understanding of the subject.

PU2. Using WebCT would increase my productivity.

PU3. I find WebCT to be useful.

PU4. Using WebCT would improve my results.

PU5. Using WebCT would assist with my learning/study.

Perceived Voluntariness:

Vol1. My use of WebCT is voluntary.

Vol2. Although it might be helpful, using WebCT is certainly not compulsory in my course.

Vol3. My lecturers do not require me to use WebCT.

WebCT Extent of Usage (Scale 1 – 7)

Course Content Module:

Calendar

Course Syllabus

Course Content Module

Glossary

Search

Compile

Communications Module:

Discussion

Chat

Email
Whiteboard

Evaluation Module:
Quizz/Survey
My Grades module
Self-tests
Assignments module

Study Tools Module:
My Progress module
Student Home Pages
Student Presentations
Student Tips

Authors' Contact Details

Irwin Brown (corresponding author: ibrown@commerce.uct.ac.za)
Belinda Knol
Simon Vincent
Department of Information Systems,
University of Cape Town, Cape Town, South Africa

The Application of the Cognitive Dimension Framework for Notations as an Instrument for the Usability Analysis of an Introductory Programming Tool

Charmain Cilliers
André Calitz
Jéan Greyling

1. Introduction

In recent years, South African tertiary education institutions have experienced increasing pressure from national and provincial government to improve student throughput rates. The consequent expectations of higher throughput rates in introductory programming courses have resulted in the identification and investigation of effective methods and strategies that assist students in overcoming difficulties experienced with computer programming.

A successful learning environment for introductory programming students has been described as satisfying the following constraints (Brusilovsky *et al.* 1994):

- the learning environment should support a notation that consists of a small, simple subset of the programming constructs generally available in a programming notation;
- the visual appearance of the program structure should enable an introductory programming student to comprehend the semantics of the programming constructs supported; and

*Alter*nation 12.1b (2005) 543-576 *ISSN 1023-1757*

- the environment should shield introductory programming students from misinterpretations and misunderstandings.

The programming environment most commonly used by students of introductory programming courses is categorised as being a commercial programming environment, examples being Delphi™ Enterprise and Visual Studio (De Raadt *et al.* 2002). Commercial programming environments typically support textual programming notations and not the alternative visual programming notations.

Further, commercial programming environments have been partly blamed for the fact that introductory programming courses are often perceived by students as being difficult (Hilburn 1993; Calloni & Bagert 1997; Warren 2000). Commercial programming environments have also typically been designed for use by experienced programmers who are developing large programs (Ziegler & Crews 1999). The debugging tools supported by the traditional programming environments are complex to initiate and use and are at times by choice avoided by more advanced programmers. These kinds of tools are thus inappropriate for use by students of introductory programming courses.

Iconic programming notations, a subset of visual programming languages, have been proposed as an alternative programming notation specifically aimed at introductory programming students (Burnett & Baker 1994; Calloni & Bagert 1994, 1995, 1997). An iconic programming notation is one in which each visual sentence is a spatial arrangement of icons, with each icon having a distinct meaning (Chang *et al.* 1994). Iconic programming notations attempt to simplify the programming task by reducing the level of precision and the incidence of manual typing typical of textual programming notations (Blackwell 1996).

In response to the challenge of increasing throughput in introductory programming courses, the Department of Computer Science and Information Systems (CS&IS) at the Nelson Mandela Metropolitan University (the former University of Port Elizabeth (UPE)) identified the need for the development of an experimental iconic programming

notation, B# (Brown 2001; Thomas 2002; Cilliers *et al.* 2003; Yeh 2003; Greyling *et al.* 2004). B# was deliberately designed to be a short term visual programming notation providing initial technological support in the learning environment of an introductory programming course.

One factor that has a bearing on the success of B# as technological support in the learning environment of an introductory programming course is the level of usability supported. The usability of computer software is typically measured in terms of the way that users interact with the software package. A well known technique that is often used in the measurement of usability is Nielsen's ten usability heuristics (Nielsen 1994b). An alternative technique is that of the cognitive dimensions framework for notations (Green & Petre 1996). The latter technique consists of fourteen individual cognitive dimensions and is primarily aimed at measuring the usability of programming tools.

This paper reports on an investigation into the usability of B# within the context of B# being classified as a successful learning environment, as attributed earlier to Brusilovsky (1994). The criteria used in the usability analysis of B# are a set of usability criteria for programming tools known as the cognitive dimensions framework for notations. The cognitive dimensions framework is used to assess the usability of B# at two levels, namely at the software design and student programmer levels. The criteria used in the usability assessment deviates slightly from Nielsen's well documented and familiar usability principles as defined by the Heuristic Evaluation Usability Engineering method.

The paper proposes a mapping that illustrates the correspondence of the fourteen cognitive dimensions to Nielsen's ten heuristics. Thereafter, each cognitive dimension is individually discussed in terms of an assessment from a design perspective of the way in which B# supports it. A quantitative and qualitative data analysis of a cognitive dimension questionnaire administered to students of an introductory programming course using B# follows. The investigation concludes that B# provides an integrated visual environment that attempts to enhance the learning experience of the introductory programming course student by supporting the cognitive dimensions of notation framework for programming languages, with a view to ultimately increase the throughput in introductory programming courses.

2. Background

Transformations in the South African political and educational scenario over the past few years have resulted in increasing pressure from national and provincial government to improve student throughput rates at national tertiary institutions (Department of Education 2001). The problem of sustaining recommended satisfactory throughput rates in tertiary level courses is further compounded by the fact that currently larger numbers of under-prepared students are entering South African tertiary education institutions (Warren 2001; Monare 2004). The resulting higher incidence of under-prepared students in South African tertiary education institutions has a particular significance for introductory programming courses which rely heavily on the use of technological tools as components of the teaching model. The prevalence of technologically under-prepared students in introductory programming courses consequently impacts on the group profile of the students and overall throughput rate of these courses.

Maintaining satisfactory group and individual performance rates in introductory programming courses is not constrained to South African tertiary education institutions. The sustaining of acceptable levels of performance remains an issue that is constantly being addressed by tertiary education institutions worldwide (Lister & Leaney 2003). Acknowledged as being of great importance in efforts to elevate the throughput rate in an introductory programming course at tertiary level are effective methods and strategies that assist students to overcome difficulties associated with computer programming (Carbone *et al.* 2001).

Typical difficulties experienced by students in introductory programming courses include deficiencies in problem-solving strategies, misconceptions related to programming notation constructs and the use of traditional programming environments (Studer *et al.* 1995; Proulx *et al.* 1996; Deek 1999; AC Nielsen Research Services 2000; McCracken *et al.* 2001; Satratzemi *et al.* 2001). The resulting increased demands on lecturing and computing resources as a consequence of attempts to address these difficulties creates an urgent need for methods to raise the successful completion percentage of candidates of already oversubscribed introductory programming courses without reducing the quality of the course (UCAS 2000; Boyle *et al.* 2002).

One approach to this problem is the modification of the introductory programming course teaching model (Wilson & Braun 1985; Austin 1987). This strategy incorporates the modification of course presentation techniques to support students at a technological level. One such type of technological support is a class of programming languages known as visual programming languages, of which iconic programming notations is a category.

International quantitative research in the use of an iconic programming language to encourage satisfactory performance achievement in introductory programming students has prompted related research at UPE (Calloni & Bagert 1994, 1995, 1997). An iconic programming notation, B#, has consequently been developed in the Department of CS&IS at UPE for use as the technological support in the learning environment of the university's introductory programming course.

Concerns exist as to the cost/benefit ratios when using technological support in learning environments, specifically regarding the maintenance of the balance between learning about the supporting software and learning about the content contained therein (Rader *et al.* 1998). It has been observed by numerous researchers that implementation issues evident in the software provided by traditional commercial textual programming environments can distract students of introductory programming courses so that they do not comprehend the programming abstractions required for the correct implementation thereof (Reek 1995; Lidtke & Zhou 1998; Ziegler & Crews 1999; Proulx 2000; Warren 2000, 2001).

Although conventional textual programming environments concurrently display many programming constructs on the screen, they tend to under-determine the student by providing no guidance as to the textual symbols required to be entered, resulting in a large gap between the plan of the desired program solution and the supported programming notation (LaLiberte 1994; Wright & Cockburn 2000). The student is thus forced to provide precisely correct notation syntax before receiving any response to the solution plan and implementation thereof (Crews & Ziegler 1998). Further, the lack of sufficient visual feedback in the use of such programming tools makes the comprehension of notation semantics more difficult for a student (Satratzemi *et al.* 2001). Features of conventional programming development environments include complex hierarchical menu structures and intricate user interfaces. These properties are often experienced by students as distractions from the task of programming (Reek 1995; Lidtke & Zhou 1998; Ziegler & Crews 1999; Proulx 2000; Warren 2000, 2001).

Important factors to consider when making a choice of programming notation for use by students of introductory programming courses is how easily they will learn the chosen notation, the existence of any notation features that might interfere with the understanding of the fundamental programming concepts, and any notation features that ease the transformation of the beginner programmer to one who is competent (Dingle & Zander 2001).

Against the background described, the criteria applied in the analysis of the usability of B# as technological support in the learning environment of an introductory programming course are presented in the following section.

3. Usability Analysis Criteria

The usability analysis of software typically deals with the analysis of the way in which users interact with computer software. One familiar and experimentally verified technique is that of the application of the ten usability heuristics presented by Nielsen (1993; 1994a; 1994b). The application of Nielsen's usability heuristics are typically restricted to deal with user interfaces.

An evaluation technique proposed specifically for visual programming language notations and their associated development environments is that of the cognitive dimensions framework for notations (Green & Petre 1996). The cognitive dimensions framework focuses on the actions and procedures being performed by a programmer while using a programming notation and its associated environment.

Each of the two sets of usability criteria is individually overviewed in this section. The section concludes with a discussion on how the two techniques presented compare with one another.

3.1. Nielsen's Heuristics

Nielsen's usability heuristics, listed and defined in Table 1, deal with the user interfaces of software systems. The heuristics have, however, recently been applied to conventional textual programming environments in the context of the interaction of student programmers with a computer (Warren 2003).

The application of the technique to conventional textual programming environments provides insight into the problems that student programmers experience with traditional commercial textual programming notations and development environments.

Heuristic	Description
N1: Visibility of system status	The system should always keep users informed about what is going on through appropriate feedback within reasonable time.
N2: Match between system and the real world	The system should speak the users' language rather than system-oriented terms.
N3: User control and freedom	The system should clearly assist users in exiting from an undesired state.
N4: Consistency and standards	Users should not have to wonder whether different words, situations or actions mean the same thing.
N5: Error prevention	Even better than good error messages is a design which prevents a problem from occurring in the first place.
N6: Recognition rather than recall	The user should not have to remember information from one part of the dialogue to another.
N7: Flexibility and efficiency of use	The system should cater to both inexperienced and experienced users.
N8: Aesthetic and minimalist design	Every extra unit of information competes with the relevant units of information and diminishes their relative visibility.
N9: Help users recognise, diagnose and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
N10: Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focussed on the user's task, list concrete steps to be carried out, and not be too large.

Table 1: Nielsen's Ten Usability Heuristics

Warren (2003) concludes that conventional textual programming notations like C++, C#, Java, Delphi and Visual Basic set in their respective development environments are large intricate systems that fail to satisfy the majority of Nielsen's usability heuristics to some level. According to Warren, the specific usability heuristics not satisfied by conventional textual programming notations and their associated development environments are those listed in Table 1, with the exception

of the heuristic of consistency and standards (N4) that is only partially satisfied. In contrast, Warren's recommendation in terms of Nielsen's heuristics is that the use of spreadsheet software followed by a scripting language such as JavaScript, together with an HTML editor with integrated browser capabilities as the technological support in the learning environment of an introductory programming course, adheres to the usability heuristics listed in Table 1 more closely.

An alternative technique for assessing the usability of a programming tool, namely the cognitive dimensions framework for notations, is discussed in the next section.

3.2. Cognitive Dimensions Framework for Notations

The cognitive dimensions framework for notations is an evaluation technique for interactive devices and non-interactive notations that has evolved over the past 15 years (Green 1989; Green & Petre 1996; Green & Blackwell 1998). This technique is task-specific and concentrates on the processes and activities being performed by programmers while using the software system rather than on the software deliverable itself.

In the case of the iconic programming notation B#, cognitive dimensions are the descriptions of the system-student relationship and are intended as a measurement instrument at a high level of abstraction. The cognitive dimensions framework for notations was initially intended for use during the early stages of the design process of a programming tool due to its structural assessment characteristic.

The definitions of the individual components of the cognitive dimensions framework appear in Table 2 (Green & Petre 1996; Blackwell & Green 2000). The framework has also been used in the design of questionnaires aimed at programmers to assess the usability of the programming tools used (Kadoda *et al.* 1999; Blackwell & Green 2000). The cognitive dimensions framework emphasises that programming tools include both a notation and a development environment, and that usability is a function of the two.

Cognitive Dimension	Description
CD1: Abstraction management	The system provides facilities for the definition of new concepts or constructs within the notation.
CD2: Closeness of mapping	The notation supported by the system closely resemblances the program solution being described.
CD3: Consistency	The system supports similarity in different parts of the notation that have like meanings.
CD4: Diffuseness	The system supports brevity in the description of solutions within the provided notation.
CD5: Error-proneness	The system permits the making of unnecessary mistakes which are a hindrance to the programming task.
CD6: Hard mental operations	The system has features that require a large amount of mental effort to use effectively.
CD7: Hidden dependencies	The system enforces consistency and a high level of visibility between closely related components of the notation.
CD8: Premature commitment	The system places a restriction on the ordering of subtasks within the programming task.
CD9: Progressive evaluation	The system supports the execution of partially completed versions of the solution.
CD10: Role-expressiveness	The system supports the easy identification of constructs within a program solution.
CD11: Secondary notation	The system supports annotations that convey meaning to the student.
CD12: Viscosity	The system supports the simplification of modifications to existing programs.
CD13: Visibility and juxtaposibility	The system supports the easy location of the various parts of the notation, and if corresponding representations are required to be compared, then the student is able to view them at the same time, preferably alongside one another.
CD14: Provisionality	The system supports the interactive modification of a solution and permits the determination of the effect of programming decisions.

Table 2: Cognitive Dimensions Framework Components

Since both Nielsen's heuristics and the cognitive dimensions for notations framework are usability analysis techniques used in the assessment of programming tools, a mapping of the 14 cognitive dimensions to Nielsen's 10 usability heuristics is discussed next.

3.3. Equivalence of Usability Analysis Criteria

Both Nielsen's usability heuristics and the cognitive dimensions framework have been used by various researchers as measurement instruments of the usability of programming tools (Kadoda *et al.* 1999; Blackwell & Green 2000; Warren 2003). Table 3 illustrates the correspondence and overlap of the two sets of usability analysis criteria.

Nielsen's Usability Heuristic	Cognitive Dimension
N1: Visibility of system status	CD7: Hidden dependencies CD9: Progressive evaluation CD11: Secondary notation CD13: Visibility and juxtaposibility CD14: Provisionality
N2: Match between system and the real world	CD2: Closeness of mapping
N3: User control and freedom	CD1: Abstraction management CD6: Hard mental operations CD8: Premature commitment CD9: Progressive evaluation CD11: Secondary notation CD12: Viscosity
N4: Consistency and standards	CD3: Consistency
N5: Error prevention	CD5: Error-proneness
N6: Recognition rather than recall	CD10: Role-expressiveness
N7: Flexibility and efficiency of use	CD1: Abstraction management CD6: Hard mental operations CD8: Premature commitment CD12: Viscosity
N8: Aesthetic and minimalist design	CD4: Diffuseness
N9: Help users recognise, diagnose and recover from errors	CD5: Error-proneness
N10: Help and documentation	CD5: Error-proneness

Table 3: Correspondence between Nielsen's Usability Heuristics and the Cognitive Dimensions Framework

All 14 cognitive dimensions (CD1 – CD14) can be equated to Nielsen's 10 usability heuristics (N1 – N10) based on the definitions of each technique's components as discussed previously. In many instances, the same cognitive dimension is mapped to multiple heuristics, and the same heuristic is mapped to multiple cognitive dimensions.

As mentioned previously, for the purposes of the study reported on in this paper, the usability of B# is evaluated according to the cognitive dimensions framework. Since this measurement instrument is task-specific, the tasks relevant to creating a program solution in B# are thus the focus of the following section.

4. Development of a Program Solution in B#

In order for a program solution to be developed using B#, the student is first required to either locate and identify an existing program solution, or provide identification for a new program solution. Thereafter, the B# program is constructed in the form of a control-flow solution that closely resembles that of a flowchart. Once this has been completed, the student may execute and debug the program solution and save it for future use. An overview of the task model for developing a program solution in B# is shown in Figure 1.

Each B# program solution is constructed in the form of a flowchart of icons, with each icon representing a distinct programming construct. The flowchart is a top-down single-sequence structure of icons connected by lines, forming a box-and-line graph which is typical of visual programming languages (Green & Petre 1996; Materson & Meyer 2001). The student selects an appropriate icon from the icon palette and drags-and-drops it in the correct position on the flowchart. An example of a program solution in the B# programming notation and associated development environment is illustrated by Figure 2. The flowchart representation of the program solution appears in the left hand pane of the window.

On attachment of the icon to the flowchart, a dialogue box is opened to guide the student in the specification of the properties required by the particular programming construct being manipulated. Figure 3 illustrates an example of a programming construct dialogue. The dialogue box in Figure 3 is that applicable to the counter iteration programming construct, which typically corresponds to a **FOR** textual programming statement. The student is required and guided to correctly complete the dialogue before an icon can be successfully attached to the flowchart representation of the programming construct.

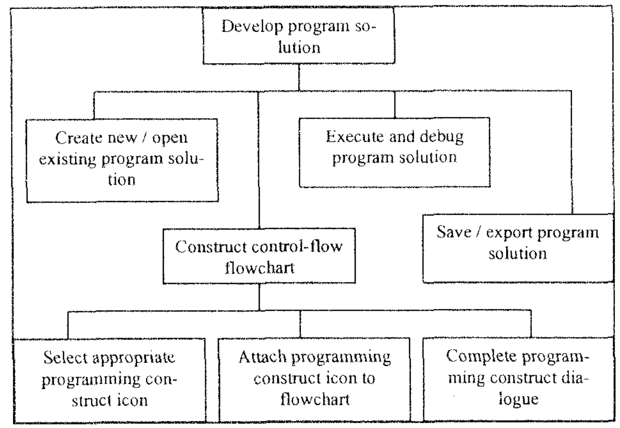


Figure 1: Task Model for the Development of a Program Solution in B#

Icons can be edited, repositioned and removed from the flowchart representation of the program solution. During the construction of a flowchart program solution, B# automatically and immediately displays the correct textual counterpart for the program solution. An example of this display is evident in the bottom right hand pane of the window illustrated in Figure 2. Figure 1 illustrates that once a B# program solution has been constructed, the student may test and debug it. Execution of program solutions is supported in two ways. The first technique is one whereby only the output from the program solution is displayed. The second technique permits the student to control the speed of the execution of the program solution. In this way, the student can trace the execution of the program solution, programming construct by programming construct.

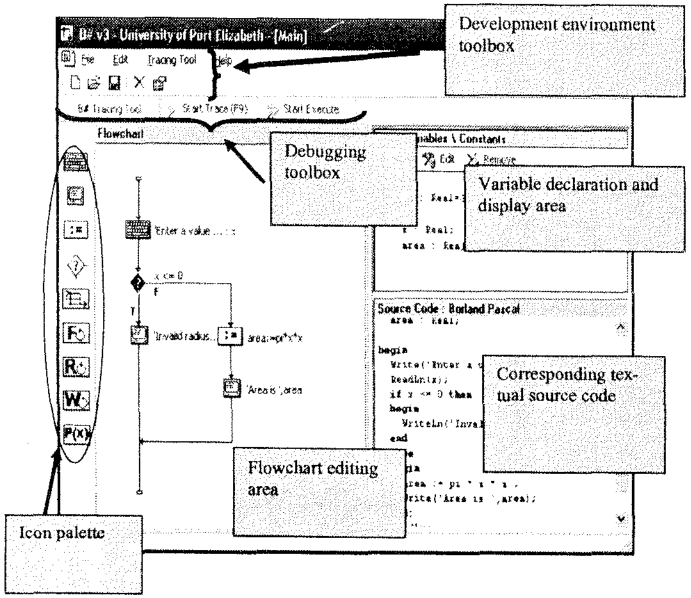


Figure 2: Sample B# Program Solution

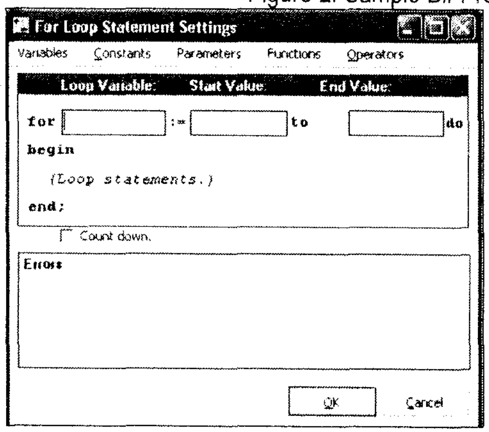


Figure 3: Dialogue Box for Counter Iteration Programming Construct

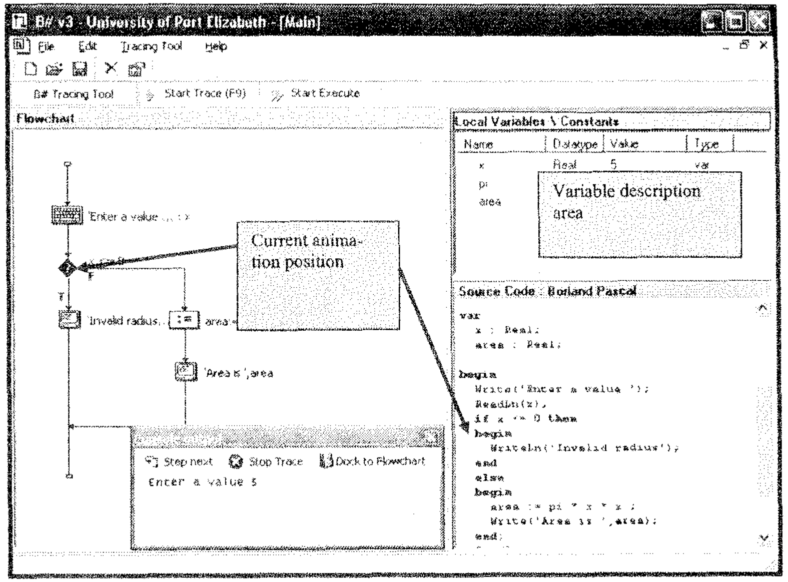


Figure 4: B# Tracing and Debugging Feature

As the student controls the tracing process, both the flowchart and textual representations of the program solution are simultaneously animated. An example of the animation appears in Figure 4 in the form of blue highlighting in each of the program solution representations. Any changes in variable values are highlighted in the variable description area to focus the student's attention on them. This area appears in the top right hand pane of the window illustrated in Figure 4.

The following section analyses the usability of B# as an introductory programming tool in terms of the cognitive dimensions overviewed in Section 3, and the task model described in this section.

5. Usability Analysis of B#

The cognitive dimensions framework, as described previously, is an evaluation technique that is task-specific and intended for use with respect to the design process of programming tools. The framework can also be used to determine the usability of a programming tool from the student programmers' point of view. The following sections report on the usability analysis of B# as a programming tool for student programmers in an introductory programming course at UPE in terms of each of these approaches.

5.1. Design Perspective

In terms of its design, B# supports the cognitive dimension of abstraction management (CD1) by allowing a student to define new programming operations using the notation provided, specifically permitting the definition and use of subroutines. The cognitive dimension of closeness of mapping (CD2) is enforced by a notation that closely resembles the solution being described. B# adopts the use of a visual flowchart which closely mirrors the control-flow or procedural paradigm initially required by the students in the context of UPE's introductory programming course. An example of the visual flowchart notation is illustrated in Figure 2 (shown in section 4).

B# maximises support for the cognitive dimension of diffuseness (CD4) by providing a small number of powerful, non-overlapping programming constructs. The number of programming constructs supported is minimised since more constructs imply more notational syntax and unnecessary complexity which could result in a student programmer experiencing confusion. The programming constructs supported by the notation of B# are shown in Table 4.

Construct	Icon	Construct	Icon
Assignment	:=	Input	
Simple conditional	?	Output	
Multiple conditional		Post-test conditional iteration	R
Counter iteration	F	Return to calling function	
Pre-test conditional iteration	W	Procedure call	P(x)

Table 4: Programming Constructs supported by B#'s notation

Also shown in Table 4 is the fact that the cognitive dimension of consistency (CD3) is supported by the notation of B#. All looping or iteration constructs (counter iteration, pre-test conditional iteration and post-test conditional iteration) have similar metaphorical images, yet remain visually distinct.

B# minimises the incidence of unnecessary errors which hinder the programming task by means of its context sensitive views. This property is evident specifically in the case of the customised dialogues implemented for programming constructs, an example of which is illustrated by Figure 3 in the previous section. Further evidence of the prevention of unnecessary errors is the fact that B# hides from the student all mundane syntactical issues (for example, the correct positioning and inclusion of semi-colons within the corresponding textual language). In these ways the cognitive dimension of error proneness (CD5) is minimised.

During the prototype development of B#, it was observed that the feature of providing the facility for the construction of student defined programming operations required the most mental effort to comprehend and successfully implement. Consequently, it was determined that B# exhibited the cognitive dimension of hard mental operations (CD6). Support for the identified offending feature was thus modified in the most recent version of B# (Yeh 2003) in order to minimise support for the cognitive dimension of hard mental operations.

B# maximises the visibility of the inter-dependence of components by the concurrent display of the flowchart program solution and the corresponding syntactically correct textual version of the program solution. This feature is illustrated in Figure 2 in the previous section. In this way, B# minimises incidences of the cognitive dimension of hidden dependencies (CD7).

An example of the manner in which B# minimises the incidence of the cognitive dimension of premature commitment (CD8) is the way in which variables required by the program solution only need to be defined as they are required. This enhances the student control and freedom of the programming environment. Figure 5 illustrates that the declaration of a variable is required only after the selection and attachment of the assignment programming construct has been initiated by the student. The student is thus not required to declare variables prior to deciding that the assignment operation is the programming construct required to be implemented. B# does, however, provide for a facility whereby more experienced students may pre-declare variables should they so wish. In this way, B# refrains from enforcing a particular order of performing programming tasks during the development of a program solution.

Any B# program solution representation (both flowchart and corresponding textual) is always syntactically correct and the student may consequently execute and debug a program solution using the tracing facility at any point during program solution construction. This feature promotes the cognitive dimension of progressive evaluation (CD9) and is illustrated in Figure 4 of the previous section.

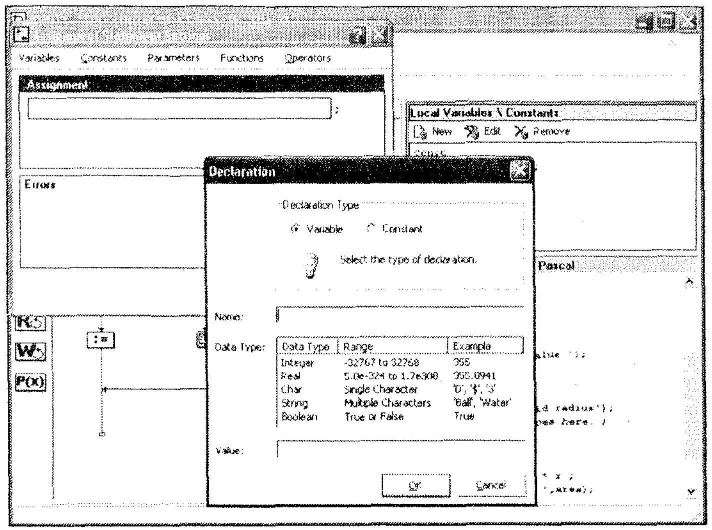


Figure 5: Declaration of variable required only when necessary

The cognitive dimension of role expressiveness (CD10) is supported in B# by means of the use of distinct metaphorical icons for each programming construct found in the flowchart representation for a program solution. Support for the cognitive dimension of role expressiveness is illustrated by the easy visual identification of distinct programming construct images displayed in the flowchart representation of the program solution in Figure 2 (shown in section 4).

Examples of the manner in which B# exhibits the cognitive dimension of secondary notation (CD11) is by means of the vertical and horizontal arrangement of programming construct icons in relation to one another in the flowchart representation of a program solution. A sample of this feature is illustrated in Figure 2.

Vertical arrangement of icons in the flowchart representation of a program solution is an indication of the flow of control, whereas horizontal arrangement is an indication of mutual exclusive selection. B# supports the cognitive dimension of secondary notation in the textual representation of a program solution by means of signalling (or code highlighting), also illustrated in Figure 2.

B# further treats nested programming constructs as a group and the student, by means of a single action, can, for example, successfully and easily reposition a group of programming constructs as a single unit within the flowchart representation of a program solution. In this way B# provides for the support of the simplification of modifications to existing program solutions thereby supporting the cognitive dimension of viscosity (CD12).

Support for the cognitive dimension of visibility and juxtaposibility (CD13) is supported by B# in that the development environment provided by B# minimises the effort required on the part of the student to search for related information. This is evident in the way in which B# displays corresponding representations of a program solution alongside one other (Figure 2). B# also ensures that the student is able to manage the screen display with the minimal amount of windows. Further, minimal effort on the part of the student is required to determine the current status of the system.

B# supports the displaying of the current status of the system by means of context sensitive views, one example being where the user is confronted with a blank display and is thereby encouraged to either create a new program solution or open an existing one. Figure 6 illustrates this system status.

Other examples exhibited by B# in support of the cognitive dimension of visibility and juxtaposibility is the display of an appropriate dialogue for any specific programming construct being added to the flowchart (Figure 3 in section 4) as well as the tracing facility which provides simultaneous animation of corresponding flowchart and textual program solutions (Figure 4 in section 4).

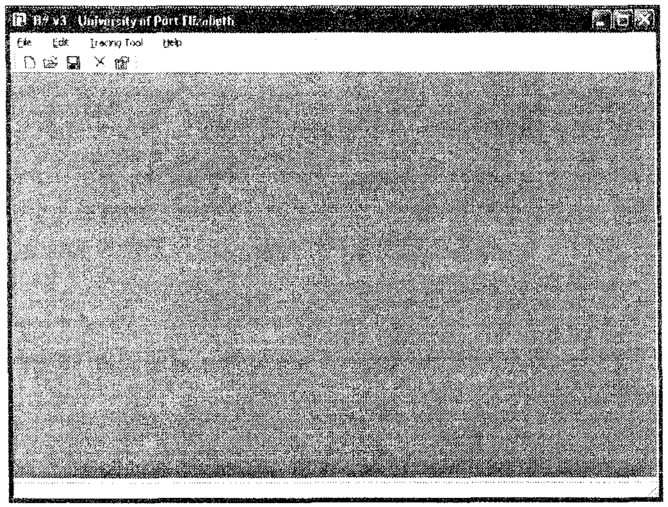


Figure 6: Illustration of Initial System Status

B# provides support for the final cognitive dimension of provisionality (CD14) by facilitating the easy repositioning of nested groups

of programming constructs represented by icons as well as by means of the tracing facility (Figure 4).

In addition to evidence of B# support for the 14 cognitive dimensions at a design level, an evaluation of the system by means of a student questionnaire derived from the set of cognitive dimensions was conducted, the results of the analysis thereof being discussed in the following section.

5.2. Student Programmer Perspective

A questionnaire (Appendix A) customised and adapted from the generic questionnaire proposed by Blackwell & Green (2000) was administered to students using B# as technological support in an introductory programming learning environment. The aim of the survey was to collect data for the examination and testing of the following hypothesis for significance at the 95% percentile ($\alpha = 0.05$) (Berenson & Levine 1999):

> H_0: *An equal number of positive and negative responses for each cognitive dimension are received from the student assessment of the usability of B#.*
>
> H_1: *An unequal number of positive and negative responses for each cognitive dimension are received from the student assessment of the usability of B#.*

The test statistic applicable to the quantitative data analysis is a computed proportion based on the number of positive responses observed for each cognitive dimension on administering the questionnaire. The statistical technique thus appropriate to the analysis of the data collected is the χ^2-test for the homogeneity of proportions using a contingency table to test the equality of the number of positive and negative responses for each of the 14 cognitive dimensions as defined in section 3. STATISTICA (StatSoft Inc. 2001) is the data analysis tool used in the computations required for the χ^2-test.

The cognitive dimensions questionnaire was administered to a group of 18 (of a possible 25) introductory programming students at UPE during 2004. The course was one of the smaller introductory programming courses and was the only course where B# was extensively and exclusively used. The subjects of the study had been using B# as technological support in the learning environment of the introductory programming course for a period of 10 weeks. Each weekly exposure to B# consisted of a single session of at least 75 minutes.

The portion of the questionnaire that is dedicated to the main notation of B# is designed to provide at least one numbered item relevant to each of the 14 cognitive dimensions. The purpose of each numbered item is to provide the opportunity for the respondents to recognise the existence of features in B# that are relevant to each cognitive dimension in terms of the programming task performed by the students. Table 5 maps each of the items appearing as numbered questions in the questionnaire in Appendix A to the appropriate cognitive dimension.

The cognitive dimension of abstraction management (CD1) is not included due to the fact that at the time of the administering of the questionnaire, the curriculum being followed by the introductory programming students had not yet progressed sufficiently to the point where the students were able to effectively assess the particular features supported by B# that are relevant to this particular cognitive dimension.

Cognitive Dimension	Question number	Positive response values	Negative response values
CD2: Closeness of mapping	7	3, 4, 5	1, 2
CD3: Consistency	13	3, 4, 5	1, 2
CD4: Diffuseness	4	3, 4, 5	1, 2
CD5: Error-proneness	6	1, 2, 3	4, 5
CD6: Hard mental operations	5	1, 2, 3	4, 5
CD7: Hidden dependencies	9	3, 4, 5	1, 2
CD8: Premature commitment	12	1, 2, 3	4, 5
CD9: Progressive evaluation	10	3, 4, 5	1, 2
CD10: Role-expressiveness	8	3, 4, 5	1, 2
CD11: Secondary notation	14	3, 4, 5	1, 2
CD12: Viscosity	3	3, 4, 5	1, 2
CD13: Visibility and juxtaposibility	1 and 2	3, 4, 5	1, 2
CD14: Provisionality	11	3, 4, 5	1, 2

Table 5: Correspondence of Questionnaire Questions to Cognitive Dimensions

Results of the statistical analysis of the responses to each of the numbered items in the questionnaire are presented in Table 6. The null hypothesis is clearly rejected for 10 of the 13 cognitive dimensions at the 99% level of confidence, and rejected for the remaining 3 cognitive dimensions at the 95% level of confidence. By inspection, it can be interpreted that in the case of the majority of the cognitive dimensions, a significantly greater proportion of positive responses were observed. Only in the cases of the cognitive dimensions of error-proneness (CD5), hard mental operations (CD6) and premature commitment (CD8) were a significantly greater proportion of negative responses observed.

The questionnaire administered to the students also attempts to solicit the respondents' characterisation of the type of activity for which B# is used in an introductory programming course. Qualitative analysis through the technique of thematic analysis was applied to the data collected for the purpose of this characterisation (Ely *et al.* 1995; Ely *et al.* 1999; Dee Medley 2001).

Cognitive Dimension	Number of Positive observations	Φ-test statistic	p-value
CD2: Closeness of mapping	13	20.350	0.000**
CD3: Consistency	11	9.750	0.002**
CD4: Diffuseness	12	4.000	0.046*
CD5: Error-proneness	5	5.460	0.019*
CD6: Hard mental operations	6	4.000	0.046*
CD7: Hidden dependencies	13	13.830	0.000**
CD8: Premature commitment	5	7.110	0.008**
CD9: Progressive evaluation	16	28.800	0.000**

CD10: Role-expressiveness	17	28.440	0.000**
CD11: Secondary notation	14	16.200	0.000**
CD12: Viscosity	16	21.780	0.000**
CD13: Visibility and juxtaposibility	17 and 12	28.440 and 11.690	0.000** and 0.001**
CD14: Provisionality	16	25.080	0.000**

*α = 0.05 **α = 0.01

Table 6: Results of Usability Evaluation of Main Notation of B# with respect to Cognitive Dimensions of Notations

Thematic analysis of the responses to this question determined that students interpreted the main task/activity for which B# is used to be the creation of program solutions for problems in the form of a flowchart. Typical responses to this question are:

"Solving problems similar to flowcharts"
"Creation of flowcharts"
"Creating programs"

The questionnaire further attempts to determine the existence of any problems related to the usability of B# that are not specifically addressed by the cognitive dimensions. Analysis of the responses determined that the task that took the most time in B# was the reorganisation and restructuring of a B# program solution. The tasks that occupied the least amount of time were searching for the correct programming icon to use and the unproductive experimentation of programming construct icons within program solutions.

Thematic analysis of the responses also indicated that additional assistance was required to be provided by B#. Typical responses indicative of this are:

"By putting a Help function"
"Have back going arrows like in flowcharts. Not only forward arrows"
"Being able to indicate location of the error within the program"

As a result of the quantitative and qualitative analysis of the student programmer assessment of the usability of B#, it is noticeable that additional system support for on-line help is required.

6. Discussion and Conclusions

The paper reported on an investigation into the usability of B#, an iconic programming notation and development environment developed by the Department of CS&IS at UPE, as an appropriate technological support tool in the learning environment of an introductory programming course. The results of the analysis are presented on two levels, namely in terms of the design of B# and with respect to the experience of students using B# to construct program solutions.

At the design level, B# is shown to positively provide support for all 14 cognitive dimensions. Application of a mapping between the technique of the cognitive dimensions framework for notations and Nielsen's heuristics implies that in satisfying the 14 cognitive dimensions, B# satisfies the latter technique of Nielsen's 10 heuristics in their entirety.

In order to confirm the level of support for the cognitive dimensions, a survey was administered to introductory programming students using B# and the responses quantitatively and qualitatively analysed.

The analysis of the responses observed by means of the survey determined that 77% of the cognitive dimensions are positively supported by B#. The analysis, however, provides evidence of usability problems in B# that are not specifically addressed by the cognitive dimensions. These identified usability problems have a relationship with the remaining 23% of cognitive dimensions not directly positively supported by B# in terms of the student assessments.

It is interesting to note that all of the cognitive dimensions assessed by student programmers as not being positively supported by B# are without exception the only questionnaire items that were negatively phrased.

In terms of the cognitive dimensions for notations assessment of the usability of B#, the experimental programming notation and environment can be classified as a successful learning environment for introductory programming students for the following reasons:

- B# supports a notation that consists of a small, simple subset of the programming constructs generally available in a programming notation. This property is evident in B# maximising support for the cognitive dimension of diffuseness (CD4).
- The visual appearance of B#'s program structure enables an introductory programming student to comprehend the semantics of the programming constructs supported. This property is evident in B# maximising support for the cognitive dimension of closeness of mapping (CD2).
- B# attempts to shield introductory programming students from misinterpretations and misunderstandings. This property is supported in B# by maximising support for the cognitive dimension of consistency (CD3). In spite of this observed support, there seems to be evidence that the property is insufficiently supported in that support for the cognitive dimension of error-proneness (CD5) requires further consideration.

Although the paper argues that in terms of support for the cognitive dimensions framework for programming languages, B# provides an integrated visual environment that attempts to enhance the learning experience of the introductory programming course student, it is clear that further research in the usability of B# is necessary.

7. Acknowledgements

The material contained in this paper is based upon work supported by the National Research Foundation (NRF) under grant number 2054293. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF does not accept any liability in regard thereto.

References

AC Nielsen Research Services 2000. *Employer Satisfaction with Graduate Skills*.

Austin, HS 1987. Predictors of Pascal Programming Achievement for Community College Students. *ACM SIGCSE Bulletin* 19,1:161 - 164.

Berenson, ML & DM Levine 1999: *Basic Business Statistics: Concepts and Applications*. 7th Edition, Prentice-Hall International, Inc.

Blackwell, AF 1996. Metacognitive Theories of Visual Programming: What do we think we are doing? *Proceedings of IEEE Symposium on Visual Languages*: 240 - 246.

Blackwell, AF & TRG Green 2000. A Cognitive Dimensions questionnaire optimised for users. AF Blackwell & E Bilotta (eds.) *Proceedings of 12th Annual Workshop of the Psychology of Programming Interest Group*, Corigliano Calaboro, Cosenza, Italy: 137 - 154.

Boyle, R, J Carter & M Clark 2002. What Makes Them Succeed? Entry, Progression and Graduation in Computer Science. *Journal of Further and Higher Education* 26,1:3 - 18.

Brown, D 2001. B#: *A Visual Programming Tool*. Honours Treatise. Department of Computer Science and Information Systems, University of Port Elizabeth. Port Elizabeth, South Africa.

Brusilovsky, P, A Kouchnirenko, P Miller & I Tomek 1994. Teaching Programming to Novices: A Review of Approaches and Tools. *Proceedings of ED-MEDIA 94-World Conference on Educational Multimedia and Hypermedia*, Vancouver, British Columbia, Canada.

Burnett, MM & MJ Baker 1994. A Classification System for Visual Programming Languages. *Technical Report 93-60-14*. Department of Computer Science, Oregon State University, Corvallis.

Calloni, B A & D J Bagert 1994. Iconic Programming in BACCII© vs. Textual Programming: Which is a Better Learning Environment? *ACM SIGCSE Bulletin* 26,1:188 - 192.
Calloni, B A & D J Bagert 1995. Iconic Programming for Teaching the First Year Programming Sequence
http://fie.engrng.pitt.edu/fie95/2a5/2a53/2a53.htm. Accessed on 26 September 2002.
Calloni, B A & D J Bagert 1997. Iconic Programming Proves Effective for Teaching First Year Programming Sequence. *ACM SIGCSE Bulletin* 28,1:262 - 266.
Carbone, A, J Hurst, I Mitchell & D Gunctone 2001. Characteristics of Programming Exercises that lead to Poor Learning Tendencies: Part II. *ACM SIGCSE Bulletin* 33,3:93 - 96.
Chang, S K, G Polese, S Orefice & M Tucci 1994. A Methodology and Interactive Environment for Iconic Language Design. *International Journal of Human-Computer Studies* 41,5:683 - 716.
Cilliers, CB, JH Greyling & AP Calitz 2003. The Development and Evaluation of Introductory Programming Tools. *Proceedings of 3rd International Conference on Science Maths Technology Education*, East London, South Africa.
Crews, T & U Ziegler 1998. The Flowchart Interpreter for Introductory Programming Courses. *Proceedings of 28th Annual Frontiers in Education Conference* 1: 307 - 312.
De Raadt, M, R Watson & M Toleman 2002. Language Trends in Introductory Programming Courses. *Proceedings of Informing Science + IT Education Joint Conference (I'SITE)*: 329 - 337.
Dee Medley, M 2001. Using qualitative research software for CS Education Research. *ACM SIGCSE Bulletin*, 33,3:141 - 144.
Deek, F P 1999. A Framework for an Automated Problem Solving and Program Development Environment. *Transactions of the Society for Design and Process Science* 3,3:1 - 13.
Department of Education 2001. National Plan for Higher Education. South African Department of Education

http://education.pwv.gov.za/DoE_Sites/Higher_Education/HE_Plan/section_2.htm . Accessed on 8 March 2004,
Dingle, A & C Zander 2001. Assessing the Ripple Effect of CS1 Language Choice. *Journal of Computing in Small Colleges* 16,2:85 - 93.
Ely, M, M Anzul, T Friedman, D Garner & A MacCormack- Steinmetz 1995: *Doing Qualitative Research: Circles within Circles*. Falmer Press.
Ely, M, R Vinz, M Downing & M Anzul 1999: *On Writing Qualitative Research: Living by Words*. Falmer Press.
Green, T R G 1989. The Cognitive Dimensions of Notations. *People and Computers V*: 443 - 460. A Sutcliffe & L Macaulay (eds.). Cambridge: Cambridge University Press.
Green, TRG & AF Blackwell 1998. Cognitive Dimensions of Information Artefacts: A Tutorial
http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/
Accessed on 2 September 2002.
Green, TRG & M Petre 1996. Usability Analysis of Visual Progamming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages and Computing* 7:131 - 174.
Greyling, JH, CB Cilliers & AP Calitz 2004. The Development and Assessment of an Iconic Programming Tool. Submitted for Publication in the Special eLearning Issue of the Scholarly Journal *Alternation*: Knowledge Management and Technology in Education.
Hilburn, TB 1993. A Top-Down Approach to Teaching an Introductory Computer Science Course. *ACM SIGCSE Bulletin* 25,1:58 - 62.
Kadoda, G, R Stone & D Diaper 1999. Desirable Features of Educational Theorem Provers - A Cognitive Dimensions Viewpoint. TRG Green, R Abdullah & P Brna (eds.) *Proceedings of 11th Annual Workshop of the Psychology of Programming Interest Group*: 18 - 23.
LaLiberte, D 1994. Visual Languages

http://www.hypernews.org/~liberte/computing/visual.html Accessed on 8 August 2002.
Lidtke, D K & H H Zhou 1998. A Top-Down Collaborative Teaching Approach to Introductory Courses in Computer Sciences. *ACM SIGCSE Bulletin* 30,3:291.
Lister, R & J Leaney 2003. Introductory Programming, Criterion-Referencing, and Bloom. *Proceedings of 34th SIGCSE Technical Symposium on Computer Science Education*: 143 - 147.
Materson, TF & RM Meyer 2001. SIVIL: A True Visual Programming Language for Students. *Journal of Computing in Small Colleges* 16,4:74 - 86.
McCracken, M, V Almstrum, D Diaz, M Guzdial, D Hagan, YB-D Kolikant, C Laxer, L Thomas, I Utting & T Wilusz 2001. A Multi-National, Multi-Institutional Study of Assessment of Programming Skills of First-Year CS Students. *ACM SIGCSE Bulletin*, 33,4:125 - 140.
Monare, M 2004. Now Matric Exams Face Official Probe. *Sunday Times* 4 January 2004.
Nielsen, J 1993: *Usability Engineering*. Boston, MA: Academic Press.
Nielsen, J 1994a. Enhancing the Explanatory Power of Usability Heuristics. *Proceedings of ACM CHI'94 Conference*: 152 - 158.
Nielsen, J 1994b: *Heuristic Evaluation*. J Nielsen & RL Mack (eds.) Usability Inspection Methods: 25 - 62.
Proulx, V K 2000. Programing Patterns and Design Patterns in the Introductory Computer Science Course. *ACM SIGCSE Bulletin* 32,1:80 - 84.
Proulx, V K, R Rasala & H Fell 1996. Foundations of Computer Science: What are they and how do we teach them? *ACM SIGCSE Bulletin*, 28,S1:42 - 48.
Rader, C, G Cherry, C Brand, A Repenning & C Lewis 1998. Designing Mixed Textual and Iconic Programming Languages for Novice Users. *Proceedings of IEEE Symposium on Visual Languages*: 187 - 194.

Reek, MM 1995. A Top-Down Approach to Teaching Programming. *ACM SIGCSE Bulletin* 27,1:6 - 9.
Satratzemi, M, V Dagdilelis & G Evagelidis 2001. A System for Program Visualization and Problem-Solving Path Assessment of Novice Programmers. *ACM SIGCSE Bulletin* 33,3:137 - 140.
StatSoft Inc. 2001. STATISTICA (Data Analysis Software System) Ver. 6
Studer, SD, J Taylor & K Macie 1995. Youngster: A Simplified Introduction to Computing: Removing the Details so that a Child May Program. *ACM SIGCSE Bulletin*, 27,1:102 - 105.
Thomas, J 2002. B#: Version 2. Honours Treatise. Department of Computer Science and Information Systems, University of Port Elizabeth. Port Elizabeth, South Africa.
UCAS 2000. Universities and Colleges Admissions Service for the UK. Technical Report http://www.ucas.ac.uk/figures/index.html. Accessed on 12 February 2004.
Warren, P R 2000. Using JavaScript to Teach an Introduction to Programming http://saturn.cs.unp.ac.za/~peterw/JavaScript/calculator.html. Accessed on 5 June 2003.
Warren, PR 2001. Teaching Programming Using Scripting Languages. *Journal of Computing in Small Colleges* 17,2:205 - 216.
Warren, PR 2003. Learning to Program: Spreadsheets, Scripting and HCI. *Proceedings of Southern African Computer Lecturers Association (SACLA)*.
Wilson, JD & GF Braun 1985. Psychological Differences in University Computer Student populations. *ACM SIGCSE Bulletin* 17,1:166 - 177.
Wright, T & A Cockburn 2000. Writing, Reading, Watching: A Task-Based Analysis and Review of Learners' Programming Environments. *Proceedings of International Workshop on Advanced Learning Technologies*. IEEE Computer Society Press.

Yeh, CL 2003. Tracing Programs in B#. Honours Treatise. Department of Computer Science and Information Systems, University of Port Elizabeth. Port Elizabeth, South Africa.

Ziegler, U & T Crews 1999. An Integrated Program Development Tool for Teaching and Learning How to Program. *ACM SIGCSE Bulletin* 31,1:276 - 280.

Appendix A

This questionnaire collects your thoughts on how easy/difficult it is to use B#. The series of questions presented encourage you to think about the ways you used B# and whether B# helped you to do the things that you were required to do.

Place a 1 next to the task that took the most time in B#, a 2 next to the task that took the next most time, etc ..., with a 4 next to the task that took the least time.

Searching for the correct programming icon to use	
Translating information from pseudocode/flowcharts into a B# program	
Reorganising and restructuring a B# program	
Playing around with the different programming icons in B# without being sure of the result/purpose of each	

Please answer the following questions to the best of your ability by marking the appropriate number (1, 2, 3, 4, 5 or 6). If you are able to, please provide additional details for each point.	**Never**	**Seldom**	**Some of the time**	**Often**	**Always**	**Unsure**
1. I find it easy to locate and use any B# programming icon.	1	2	3	4	5	6
If there are any constructs that are difficult to use and/or locate, please identify them.						
2. The windows of B# that are depend-	1	2	3	4	5	6

ent upon each other are easily visible and are always consistent.						
Please identify any windows of B# that fall into this category. Why is it necessary for these windows to be dependent upon one another?						
3. I find it easy to make changes to a B# program.	1	2	3	4	5	6
If there are any things that are difficult to change in a program, please identify them.						
4. I can state a solution to a problem reasonably briefly in B#.	1	2	3	4	5	6
Is there anything specific that you think could be simpler? How could it be simplified?						
5. I find that it requires a lot of thinking to create a solution in B#.	1	2	3	4	5	6
If there are any things in B# that require a lot of thought, please identify them and describe why you find them difficult.						
6. I find that it is easy to make irritating mistakes in B#.	1	2	3	4	5	6
Describe the kind of irritating mistakes that B# allowed you to make.						
7. I find that the way in which B# programs are created and displayed closely matches the types of problems that I must solve in WRA131.	1	2	3	4	5	6
Why?						
8. B# programs are easy to follow and understand.	1	2	3	4	5	6
Why?						
9. If I make a change to a B# program, the effect is always reflected in the parts that dependent on the change.	1	2	3	4	5	6
Describe the occasion(s) when this did not occur.						
10. It is easy to stop at anytime during a B# program creation and test my work so far.	1	2	3	4	5	6
Describe the times when this was not possible.						

11. B# encourages me to experiment with a solution.	1	2	3	4	5	6
How?						
12. B# forces me to think ahead and make certain decisions about a solution first.	1	2	3	4	5	6
How?						
13. The different parts of B# that mean similar things are clearly similar from the way that they appear.	1	2	3	4	5	6
Please identify anything which you consider to be similar in B#, and describe how you identified them as being similar.						
14. The arrangement of the icons in a B# program helps me to understand the purpose of the program better.	1	2	3	4	5	6
Describe exactly how the arrangement of the icons in a B# programs affects your understanding of the program task.						

What task/activity do you use B# for? ____________________

What, in your opinion, is the end product that B# produces? _____

How do you interact with B#? ____________________

Can you think of any obvious ways that could improve the B# system?

____

Authors' Contact Details

Charmain Cilliers, (Charmain.Cilliers@nmmu.ac.za)
André Calitz, (Andre.Calitz@nmmu.ac.za)
Jéan Greyling (Jean.Greyling@nmmu.ac.za)

Department of Computer Science and Information Systems,

Nelson Mandela Metropolitan University, Port Elizabeth, South Africa

Student Perceptions of Learning Object-Oriented Programming

Leila Goosen
Vreda Pieterse

Abstract

This study was undertaken in order to determine how difficult (or not) CS1 students perceive learning to program in an object-oriented style to be, how well they actually learn object-oriented programming (OOP) and how well they retain their understanding of OOP. By analysing this information and ideas of best practices provided in related literature, suggestions are formulated for improving instruction of OO concepts.

Introduction

Students' acquisition of basic object-oriented (OO) concepts appears to be a major source of difficulty (Doube, 1996), because understanding these concepts requires them to be familiar with some potentially troublesome terminology (Ross, 1996). Knowing which OO concepts students find difficult to understand can allow lecturers to structure instruction in such a way that understanding is improved (Wiedenbeck & Ramalingam, 1999).

In the next section of this paper the research design will be elaborated upon by describing the problem statement, population and sampling, as well as how data was collected. The following section provide an elaboration on the theory of OO concepts, including objects and classes, state and behaviour, encapsulation and interfaces, inheritance and polymorphism.

*Alter*nation 12.1b (2005) 577-591 *ISSN 1023-1757*

Our research results are reported in three sections: The influence of prior learning and experience for learning to program using OO concepts; the understanding of OOP; and the identification of easy and hard topics, both generally and specifically with regard to OO topics. Understanding is interpreted in terms of the perceived and actual difficulty for students of understanding OOP, how well students retain their understanding of OO concepts, and the consistency of the latter with students' course marks.

In the last section we conclude with specific teaching guidelines for improving instruction of OO concepts.

Research Design

Problem Statement

The perception seems to exist that students have difficulty in learning to program in an object-oriented style (cf. SIGCSE, 2001). This study was undertaken in order to determine how difficult (or not) students perceive learning to program in an object-oriented style to be, how well they actually learn object-oriented programming (OOP) and how well they retain their understanding of OOP.

Population and Sampling

Our study was conducted using students who completed an OOP course at the University of Pretoria (UP) during the first semester of 2002. The following table shows the courses presented by the Department of Computer Science of UP and their prerequisites as it was presented in 2002 when this survey was conducted.

Table 1: First year programming courses at UP

Course Code	Content	Prerequisite
COS 110	Programming design principles and practices with emphasis on the object-oriented paradigm using Java as vehicle	Grade 12 Computer Studies (HG) D or an equivalent
COS 283	The course is an introduction to networking principles using Java for www and network programming.	COS110

Students who participated in our survey were enrolled for COS283 at the time of our survey, and therefore had completed COS110. It can thus be assumed that they are schooled in the concepts of object-orientation. Due to the prerequisites set for COS110, all students had programming experience in at least one programming language prior to their exposure to OOP in COS110. It should be noted that 25 students (12%) had not yet passed COS 110 at the time of the survey, but had obtained special permission to do COS 283 in parallel while redoing COS 110.

Data Collection

On October 7, 2002 all students attending class in COS 283 were invited to participate in a questionnaire. The attendance register of that day had 246 students, of whom 178 (72%) completed the questionnaire. As the questionnaire was completed anonymously, the final marks for COS 110 of all attendees of the day were obtained, together with the period (semester and year) when each student had completed the course. Of these students, 197 (80%) had passed COS 110 in June 2002, meaning the majority of students completed the questionnaire roughly four months after completing a course teaching them Object Oriented Programming.

Object Orientation Concepts

For the purpose of this research, understanding of the concepts of object-oriented programming described below is considered to be crucial in the mastering of OOP.

Objects and Classes

An *object can represent any abstract* or real world entity (Finch, 1998) that is relevant to the system, while "classes are a means for describing the properties and capabilities of the objects in real life that a program has to deal with" (Bishop, 1998:23).

State and Behaviour

An object's internal state is defined by the values of its attributes, which can be thought of as a collection of variables and data. The concept of attributes is central to object-oriented programming

(Barrow *et al.*, 2002:3). The behaviour of an object is defined it its methods. Methods specify the operations an object can perform. Among other things, it defines the way in which an object's data can be manipulated (Martin & Odell, 1992:17) to change its state.

Encapsulation and Interfaces

Objects are self-contained entities that encapsulate (hide) both their own data and functions (Barrow *et al.*, 2002:466) as described in its class definition. The interface of an object presents the visible surface of the object for other objects to communicate with it (Chakravarty & Lock, 1997:122). An object thus has control over the way in which other objects can access its data and methods, and can hide anything that it wants to keep private from other objects. The details of an object code belong to the class itself, and this code may be modified in any way desired, as long as its interface remains unchanged (Schneider & Gersting, 2004:391).

Inheritance

Use of OO design permits class definitions to be hierarchically arranged (Barrow *et al.*, 2002:476), leading to the powerful inheritance programming technique. Wirfs-Brock *et al.* (1990:24) define inheritance as "the ability of one class to define the behaviour and data structure of its instances as a superset of the definition of another class or classes." Programmers make use of the process of inheritance by abstracting all the common features into a high-level member class that represents the characteristics that are shared by all its descendants. Inheritance allows code and data structure definitions to be shared among classes with similar structure (Coggins, 1996), leading to the possibility of existing code being reused.

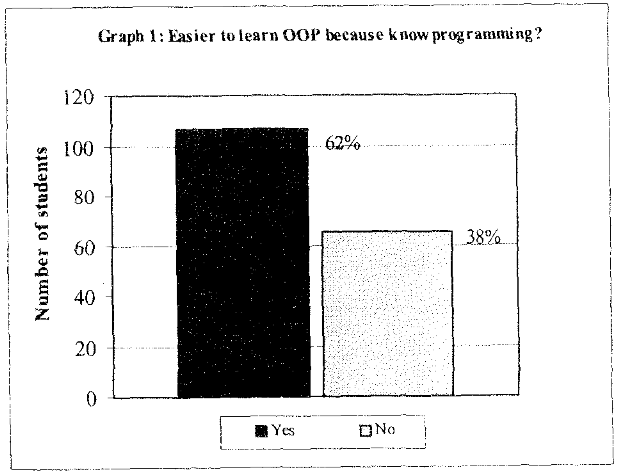
Polymorphism

The term polymorphism originates from the Greek words 'poly morph' and suggests the capacity to appear in many forms (Jupitermedia Corporation, 2003). In the context of object-oriented programming, the concept of polymorphism refers to a programming language's ability to allow different objects to react to the same stimuli

(i.e. message) differently, depending on their data type or class, by redefining methods for derived classes. An object can operate interchangeably with an ancestor in relation to the attributes and methods it inherits from the ancestor (Barrow *et al.*, 2002:465).

Influence of Prior Learning and Experience

In order to establish the influence of prior learning and experience for learning to program using OO concepts, students were asked to indicate whether it was easier to learn OOP, because they already knew programming before starting this particular course. Almost of students' prior learning experiences influenced their learning of OOP positively (see graph 1).



Students were also asked to qualify their responses. Analysis of the responses of students' who indicated that their prior learning helped them, revealed that for most students, their previous programming experience and basic background knowledge of programming concepts help them mainly in a sense that there were less new concepts to deal with.

"A bicycle is a bicycle; some just have more bells and whistles than others"...This quote is from a student who is of the opinion that it was easier to learn OOP due to previous programming experience captures the feeling that the transition were more like riding a new bicycle with some new features rather than learning to ride a totally different vehicle.

Other students, despite indicating that their previous programming indeed helped them in the course under discussion, however, felt that learning the new paradigm required some unlearning of old concepts to be able to grasp OOP at first.

The words "*Even though the jump to OOP was a bit of a ...*" used by a student to describe his/her feeling, represents a general perception that they had to overcome a considerable gap between their prior experience and the new concepts.

Although the majority felt that their prior experience helped them, one student aired the opinion that the fact that he already knew programming helped him "*very little*" during this course, while another's pre-knowledge only helped "*for about the first practical lesson and then it was impossible again*".

The reasons for not gaining from prior programming experience can be attributed to the feeling that learning OOP is very different from what students were used to.

"*very different from everything ever learnt before*"

"*a whole different approach to programming*"

"*just as hard to learn as it was to learn programming originally*"

The above quotes from respondents express the frustration experienced by some students despite their prior knowledge. Their inability to utilise their previous experience can partly be attributed to inadequate depth of their prior knowledge, but is mainly seen to be a

consequence of the vast difference between *structured* or *procedural programming* and *object orientation.* The following quote not only expresses the magnitude of the difference between these paradigms, but also indicates that the student has not yet adopted OOP: "*There is a big step from normal programming to OO programming*"

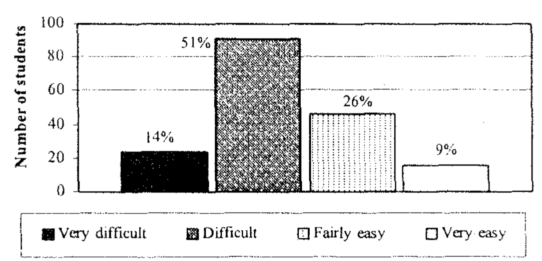
Many students still struggle with the ideas and find OOP to be "*still really, really difficult!*" and "*not easy to learn at all*". From these comments qualifying the students' choice whether their prior knowledge helped them to understand OOP, we conclude that although most of the students indicated that it had a positive impact, the extent of the impact was not profound in all cases. The major reason for students not being able to gain from their prior knowledge is twofold. Firstly they struggle to change their mindset to the new paradigm and secondly after obtaining the OOP mindset, they are unable to apply their previous knowledge in the new environment.

Understanding Object-Oriented Programming

Perceived Difficulty of Understanding OOP

Subjects were asked to select how difficult they perceived the understanding of object-oriented programming to be from the available options of "Very difficult", "Difficult", "Fairly easy" and "Easy". Almost 2/3 of the group indicated that they found the concept to be difficult or very difficult to understand (see graph 2).

Graph 2: How difficult do students find it to understand OOP?



Actual Understanding of OOP Concepts

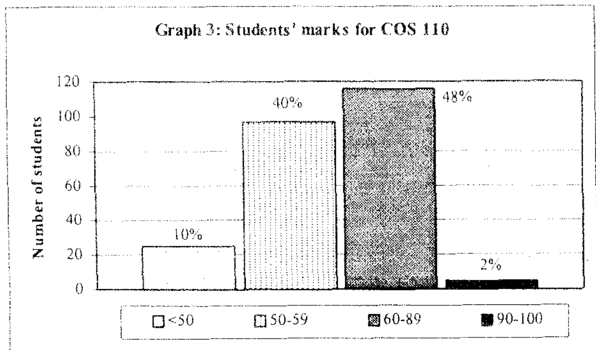
Students' formal examination marks were used as an indication of their level of understanding of the basic concepts of object-oriented programming (see graph 3). The average for students included in this study was 60 (marks out of 100) with a standard deviation of 11.

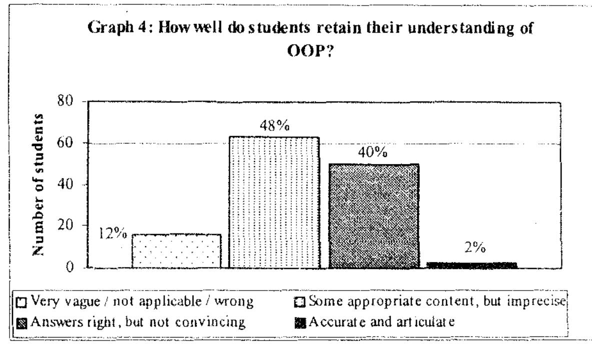
The average quoted for COS 110 might seem high. This can be attributed to the fact that COS 110 is a prerequisite for the course the students were doing when they participated in the questionnaire.

Retention of understanding of OO concepts

In order to determine students' level of retention of understanding of OO concepts, they were asked to explain in their own words what they understood by OOP. Their answers were then rated according to the following scale:

1. Very vague / not applicable / wrong.
2. Some appropriate content, but imprecise.
3. Answers are right, but don't convince the reader that the student knows exactly what the concept is about.
4. Accurate and articulate; student seem to understand the concept very well.





Classifications were made and checked by the researchers, and also verified by an external judicator. Results obtained are represented in graph 4. To illustrate the range of answers and give an indication of the way in which responses were classified, figures 1 and 2 show some excerpts.

Figure 1: Very vague / not applicable / wrong responses

One student replied to this question with a "?", one claims that he "*didn't understand*" and two more didn't "*know how to explain these*". It is "*difficult programming*" that is "*extremely difficult to define*", involving "*programs which are built by objects*" (2) / "*action objects*". Programming like this is "*useful*" and "*uses an interface*" for the "*look and feel of the program*" in a "*basic common language*". "*It is not mainly user-based*".

Figure 2: Accurate, articulate responses

Object-oriented programming refers to "*a design style that models real world objects as bundles of code, and furthermore describes the preferred method of interaction between these software objects*". Everything is viewed "*as an object, which has attributes and ways (functions/methods) to access and modify the attributes*", as well as information hiding. "*These objects can be treated as single entities*": "*data structures containing methods and properties can be treated as a template from which objects can be instantiated.*"

Consistency between Course Marks and Retention of Understanding

A comparison of graphs 3 and 4 show consistency for course marks and retention of understanding in terms of percentages for the extreme cases: the percentage of students who obtained marks of more than 90% in the course retained their knowledge to provide accurate and articulate responses in the questionnaire, while the percentage of students who had not yet passed the course closely matches the percentage of students who were only able to provide very vague descriptions of OOP. The inversion of percentages for the middle columns of these graphs suggest that students who had obtained marks for the course in the lower sixties had retained their understanding of OO concepts (or lack thereof) to the same extent as students with marks in the fifties.

Perceived Difficulty Level

General

The answers given by students responding to the question "*Which part(s) of Java was the easiest?*" resulted in 4% indicating that nothing was easy and roughly 74% of the respondents referring to concerns not specifically related to OO concepts. Topics mentioned ranged from use of specific data structures such as arrays or Strings or known control structures such as if statements and loops, to file handling and GUI design. Some referred to external issues regarding the quality of notes or lecture behaviour. Similar to the answers to what was considered easy, 63% of the responses to the question "*Which part(s) of Java was the hardest?*" referred to issues regarding programming in general, external factors, or topics not confined to OO programming.

When asked to identify easy topics, 4% indicated that nothing was easy, and consequently that everything was hard. In contrast, 12% of the respondents admitted that they found everything hard when asked to identify hard topics. Similarly where only 2.8% indicated that they experienced no difficulties at all when asked about what they find

hard, 7.8% were willing to state that everything was easy when asked about what they found easy. This observation illustrates that the way a question is asked can have a profound impact on the statistical result.

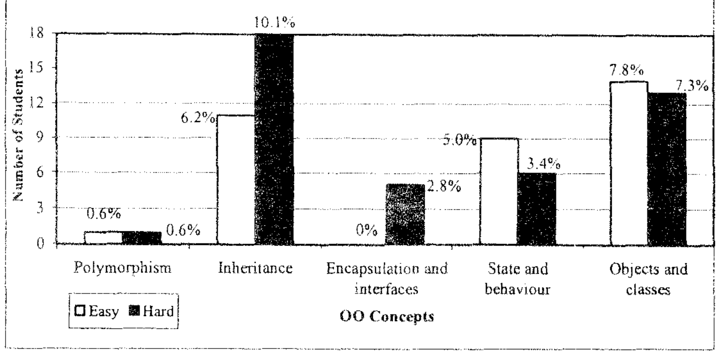
Specific Object Oriented Topics

20% of the respondents identified topics specific to OO that they found easy to understand, while 24% singled out specific OO concepts as being hard to understand. The number of students perceiving the identified OO concepts as being easy or hard to understand is shown in graph 5.

As this data was drawn from a small subset of the sample, no significant conclusions can be drawn from it. In the case of basic concepts such as *Objects and classes* and *state and behaviour* where more students found it specifically easy than students indicating it to be hard, it can be argued that there might be more students who found it hard, but could not even identify their inability to grasp the concept, as the origin of his difficulty in mastering the course content.

The case of *polymorphism* that was mentioned only twice (representing 1.2% of the sample) is merely an indication that the concept of polymorphism was not significantly covered or required in the presentation of the course to get an indication if the students perceived it as easy or hard. The fact that *encapsulation* and *inheritance* is perceived as harder to assimilate by students correlates with our experience that these concepts are more advanced. It is also comforting to realise that these topics was emphasised adequately enough in presentation of the course to lead the students to recognise the cognition of these concepts as crucial.

Graph 5: Perceived Difficulty of OO Concepts



Conclusions

Our first conclusion is that although the majority of students felt that prior experience in programming helped them to understand OOP, the extent of the impact was not profound in all cases. When students are not able to gain from their prior knowledge, it is either because they struggle to change their mindsets to the new paradigm, or after obtaining the OOP mindset, are unable to apply their previous knowledge in the new environment.

The majority of test subjects claim to find understanding object-oriented programming difficult. Traditional lecture methods can be supplemented to enhance student understanding of difficult concepts by steadily replacing these by a learner-centred approach where students have more responsibility for their own learning. A pervasive idea seems to be to present concepts in broad strokes first and add details later (Ross, 1996). Students first have to be allowed to master basic facts, features and rules and gain insight into how these relate to existing knowledge (Sharp *et al.* 2003). They find it easier to understand when they familiarize themselves with high-quality, clear examples of a new OO concept, which are contrasted with non-examples. Instead of only having purely descriptive introductory

material, "hands-on" programming exercises should be utilized (Doube, 1996).

This study shows a clear relationship between students' exam marks and their retention of understanding OOP concepts. As a projected outcome for students learning OO concepts is that they will eventually use these concepts in practical implementations in their place of employment, it is important that they retain their knowledge of these concepts. Exam marks should be a good indication to employers of how well a student would be able to apply OO concepts in actual projects in the work place.

Encapsulation, inheritance and polymorphism were identified as concepts within OOP with pivotal importance in terms of difficulty of understanding. As it is difficult to understand abstract concepts such as encapsulation and information hiding, it is important that the related advantages of these concepts be emphasized (Sharp *et al.*, 2003). Lecturers should also note that their attitudes help to determine whether students view inheritance as a difficult topic (Schaller *et al.*, 1997).

SIGCSE (2001) cautions that certain didactical problems can be exacerbated when an objects-based model is used, as many of the languages used for object-oriented programming in the industry involve significantly more detail complexity than classical languages. Unless lecturers take special care to introduce the material in a way that limits this complexity, such details can easily overwhelm students.

References

ACM Special Interest Group in Computer Science Education (SIGCSE) 2001. Chapter 7: Introductory Courses. *Computing Curricula 2001: Computer Science Volume.* http://www.acm.org/sigcse/cc2001/index.html [Date of access: 2 April 2003].

Barrow, J, JH Gelderblom & MG Miller 2002. *Introducing Delphi Programming: Theory through Practice*. 3rd Ed. Cape Town: Oxford University Press.

Bishop, JM (jbishop@cs.up.ac.za) 2000. Interface 2000. [E-mail to:] Southern African Computer Lecturers' Association. (SACLA@wwg3.uovs.ac.za) February 14.

Chakravarty, MM & HCR. Lock 1997. Towards the Uniform Implementation of Declarative Languages. *Computer languages* 23,2-4:121-160.

Coggins, JM 1996. Subject-Oriented Programming http://iraf.noao.edu/iraf/web/ADASS/adass_proc/adass_95/cogginsj/cogginsj.html [Date of access: 27 January 2003].

Doube, W 2000. The Impact on Student Performance of a Change of Language in Successive Introductory Computer Programming Subjects. *Proceedings of the Australasian Conference on Computing Education (ACE 2000)* (Melbourne, Australia, December 2000). New York, NY: ACM Press.

Finch, L 1998. So Much OO, So Little Reuse. http://www.ddj.com/documents/s=909/ddj9875g/9875g.htm [Date of access: 27 January 2003].

Jupitermedia Corporation. 2003. Polymorphism. http://webopedia.internet.com/TERM/P/polymorphism.html [Date of access: 23 September 2003].

Martin, J & JJ Odell, 1992. *Object-Oriented Analysis and Design*. Englewood Cliffs, N.J.: Prentice-Hall.

Ross, JM 1996. Learning to teach C++. *SIGCSE Bulletin* 28, 2: 25-30.

Schaller, NC, M Berman, J Bishop, P Nixon, E Rozanski, & P Welch 1997. Panel Report: Using Java in Computer Science Education. *The supplemental Proceedings of the Conference on Integrating Technology into Computer Science Education: Working Group Reports and Supplemental Proceedings (ITiCSE '97)* (Uppsala, Sweden, June 1-5 1997). New York, NY: ACM Press.

Sharp, H, ML Manns, & J Eckstein, 2003. Evolving Pedagogical Patterns: The Work of the Pedagogical Patterns Project. *Computer Science Education* 13, 4: 315-330.

Wiedenbeck, S, & V Ramalingam 1999. Novice Comprehension of Small Programs Written in the Procedural and Object-Oriented Styles. *Inernational Journal of Human-Computer Studies* 51: 71-87.

Wirfs-Brock, R, B Wilkerson, & L Weiner 1990. *Designing Object-Oriented Software*. Englewood Cliffs, N.J.: Prentice-Hall.

Authors' Contact Details

Leila Goosen (lgoosen@gk.up.ac.za)
Vreda Pieterse (vpieterse@cs.up.ac.za)

Department of Computer Science

University of Pretoria, Pretoria, South Africa

Determining Suitable Programming Language(s) for the B Tech (IT) Degree

JO Dehinbo

Abstract

There are various programming languages such as Basic, Fortran, Pascal, Cobol, C, C++, Visual Basic (VB) and Java, which are being taught within the Information Technology curriculum of tertiary institutions in South Africa. Due to differing features as in normal languages, some programming languages which are not easier to learn may lead to high failure rate. Some that are simpler to learn may not offer the flexibilities required for future works. The current practice of teaching many of these languages leads to repetition. Adequate consideration need to be given to the choice of a programming language that will be easy to learn even for first year students, and yet lead to increased productivity in future tasks.

The study involves evaluating four programming languages namely C++, Visual Basic, Java and Pascal, in terms of their ease of learning, ease of use under pressure, Line of Codes (LOC) and overall rating. Survey and Experimental approaches were used. Questionnaires were administered to respondents using any of the four languages: C++, Java, Visual Basic, and other structured languages (like Pascal or Basic) in different study groups. In addition, a simple programming exercise was given to respondents to solve and the estimated Lines of Codes (LOC) for the solution in each programming language was measured.

The study concluded that unlike Pascal, there is no significant difference in the factors studied for C++, Visual Basic and Java. The findings therefore show that a careful combination of the languages can achieve the desired result.

Keywords

Programming, languages, ICT, teaching, learning

*Alter*nation 12.1b (2005) 593-614 *ISSN 1023-1757*

1. Introduction

Various programming languages such as Basic, Fortran, Pascal, Cobol, C, C++, and Visual Basic have been used in the past as the programming language of choice for the various computer-based instructional offering for beginners in Tertiary institutions. In the Soshanguve Campus of the Tshwane University of Technology (formally Technikon Northern Gauteng - TNG), Pascal was used in 1999, C was used in 2000, and C++ has been used since 2001. The pass rate remains less that 50%.

1.1 The Context of the Problem

The use of various programming languages in computer-based training led to situations which necessitate some considerations. Adequate consideration will lead to the choice of a programming language that will lead to increased productivity. On the other hand, inadequate consideration could make programming problematic now and even in future. Below are some of the situations:

Basic is considered as a very simple language for beginners but cannot be used for complex programming. So after the first year, the students have to start with another programming language, which may lead to confusion of the statements in both languages. The new language may also not be mastered to the point where it can be used effectively for industrial work and research. Similar arguments apply for FORTRAN and Pascal.

Basic, FORTRAN, Pascal and C are structured programming languages that need a complete re-orientation to adjust to the more recent object-oriented programming style portrayed by C++, Java and event-oriented style of Delphi and Visual Basic. C++ is a hybrid language capable of the structured programming style in C as well as the object-oriented programming style compatible with Java.

However, C++ is considered by some people as being difficult, saying it seems too complex for first year students especially those with no prior knowledge of computing. Another consequence of the

complexity of C++ is that many postgraduate students do not enjoy using it for further complex programming thus abandoning it for Java.

Internet programming at present offers challenging opportunities for research especially in areas such as Servlets, CORBA, Java Server Pages, HTML, XML, COM/DCOM, and Enterprise Java Beans (EJB). These however need a good background in Java and not Basic, Fortran, Pascal or Cobol.

Many programmers, when given a choice of languages for a new project, continue to use the language with which they are most familiar, even if it is poorly suited to the new project. It is therefore very important for student to be trained using a programming language that will be most suitable for their future tasks.

1.2 The Problem Statement

The pass rate for technical programming language subjects for the past few years in TNG averaged less than 50%. This, in addition to various comments from students such as "C++ is too difficult" indicates that the various languages used so far, are possibly not easy to learn for beginners. There are some other languages which could be very good for beginners but do not have enough complexity for the student to become rigorous in programming at the end of the 4^{th} year of a B Tech study. This leads to half-baked students with problems in securing employment and in undergoing postgraduate studies and conducting research.

1.3 The Aim of the Study

The aim of the study was to identify a suitable programming language for the new B Tech (Information Technology) degree in the Soshanguve campus of the Tshwane University of Technology and other tertiary institutions. This language should be relatively easy for beginners to write useful programs, to incorporate other current and relevant IT technologies, and broad enough to fill the 3 to 4 year curriculum while at the same time offer some prospects for future industrial use and research.

1.4 Objectives of the Study

The objectives of the research are:

- Investigate the preferred or 'easy to learn' language among undergraduate students at different levels.
- Investigate the language that is considered most flexible.
- Investigate the language that is considered most suitable for complex works.
- Compute the Line of Codes (LOC) for a solution to a simple problem in each programming language.

2. Types of Programming Languages

There are basically hundreds of programming languages on the market today. According to Sebesta [7], it is widely believed that the depth at which we can think is influenced by the expressive power of the language in which we can communicate our thoughts. Those with a limited grasp of natural language are limited in their complexity of their thoughts, particularly in depth of abstraction. It is difficult for people to conceptualize structures that they cannot describe. Programmers in the process of developing software are similarly constrained. The language in which they develop software places limits on the kinds of control structures, data structures, and abstractions they can use; thus the form of algorithms they can construct are also limited.

Because of the great diversity in computer use, programming languages with very different goals have been developed. From widely accepted views in various programming texts such as [7], some of the areas of computer application and their associated languages are discussed below:

2.1 Scientific Applications

Typically, scientific applications have simple data structures but require large number of floating-point arithmetic computations. Efficiency is a primary concern. The first language for scientific applications was FORTRAN.

2.2 Business Applications

Business languages are characterized according to the needs of the application by elaborate input and output facilities and decimal data types. The first and the most successful high level language for business was COBOL which appeared in 1960.

2.3 Systems Programming Languages

The operating system and all of the programming support tools of a computer system are collectively known as its system software. A language for this domain must have low-level features that allow the software interfaces to external devices to be written. Examples of such language are Assembly Language and Extended ALGOL.

2.4 Very High-Level Languages (VHLLs)

The fourth-generation languages that were developed in the 1970s are sometimes considered to be VHLLs. These languages are used in the commercial or business area of computer applications. They have commands that are commonly programmed in languages close to English. Examples of such languages include DBASE IV, FOXPRO, and ORACLE.

2.5 General-Purpose Structured Languages

These are simple languages like BASIC, PASCAL and C that can be used for both scientific commercial applications. The central theme for these languages includes "top-down" problem decomposition and modularization. Modern programs are usually complex and being developed as a team effort. Top-down decomposition makes it easy to decompose the problem into manageable components or modules. Modularization makes it easy for different members of a team to work on different parts of the application being developed. These parts could be developed as modules or functions.

This saves development time and enhances reuse of codes. Instead of rewriting the codes in the functions in different parts of the program, they are simply called to do the required task and return result values. Compile time is also saved because, no matter how many times the functions are called, they are compiled only once. Functions also improve maintainability, extensibility and reliability. This is achieved by allowing one to fix errors or bugs in one place, rather than every where a task is performed. Readability is thereby improved by isolating code that performs specific tasks [6], making it easier to maintain and extend the system, even in a reliable way.

2.6 Object-Oriented Programming Languages

According to Lerdorf & Tatroe [6], Object-oriented programming opens the door to cleaner designs, easier maintenance, and greater code reuse. Real life objects are used to model problem domain objects as illustrated by the following statements:

> "Unlike the procedure-oriented method of programming, the object-oriented method allows the programmer to use familiar objects to solve problems. The ability to use objects that model things found in the real world makes problem solving much easier [8]".

This object-oriented method covers basic topics such as Classes, Objects, Properties and methods, Encapsulation, Information hiding and Abstract data types, Inheritance, and Polymorphism.

Zak [8] defines a class as a pattern or blueprint used to create an object. When one designs the program, one has to decide the fields for each data item, and then come up with the functions that operate on those data items. In object-oriented terms, one is designing the class by so doing [8]. Most real life objects can be grouped into some classes based on their similar properties. For example, various types of cars can have properties such as "number_of_tyres = 4", "number_of_passengers = 5", etc.

Recent examples of object-oriented programming languages used in educational institutions are C++ and Java. C++ is a hybrid language, combining the features of C with object-oriented features of Smalltalk. In Java, every program is a class, thus being completely object-oriented, though you can still use structured logic within each class.

2.7 Event-Driven Programming Languages

Event-driven programming languages include the likes of Visual Basic and Delphi. They are good for interface design involving such events as clicking a button to perform actions.

2.8 The Shift to Net-Centric Computing

The Internet and the World Wide Web are revolutionizing conventional business models and in some cases producing new ones [4]. Recently, most organizations started to adopt an approach of "write once, run anywhere", where programs are designed and installed on a web site and can be run from any computer connected to the internet. Central to this approach is the Java language, used in conjunction with middlewares such as Servlets, which can also access remote database systems. Dehinbo [3] observed that a very important feature that promotes the use of the Java language is the integration of Java to web pages through HTML.

According to Hamilton [5], Java is an object-oriented programming language with syntax similar to C and C++, only simpler. Because Java is an interpreted language, the typical C or C++ compile-link-load-test-debug cycle is reduced. The main attraction is the fact that Java applications are completely portable. Thus, you write your code once, and you never even need to port or recompile it. The Java runtime environment, or virtual machine translate the bytecode into actual machine specific instructions. Users are assured that applications are safe, even if downloaded from the internet, because the Java runtime environment, or virtual machine, has security mechanisms that protect against tampering.

According to Hamilton [5], Java originated in early 1990 with James Gosling, a software developer at Sun Inc., who was part of a team investigating advanced software techniques for a variety of networked devices and embedded systems. The team's goal was to simplify the development of secure, high performance, and highly robust applications on multiple platforms in heterogeneous distributed networks. They first considered using C++, but because of the many difficulties encountered with C++, the team began to examine other languages, including Eiffel, Smalltalk and Cedar/Mesa. In the end they decided to develop an entire new language, drawing from the best features of each of these languages, with simplicity as one of the overriding design goals.

The designers of Java describe the language as C++ without guns, knives or clubs. They mean that the difficult parts of the language such as pointers and operator overloading that bedevil programmers, as well as learners, have been removed from the language itself and are implemented within the underlying layer [1].

2.9 Languages for Training Purposes

For training purposes in tertiary institutions, not all programming languages are suitable due to factors such as cost, complexity and availability of trainers. However, the four programming languages used in this study (C++, VB, Java and other structured programming languages like Pascal) serve as good representation of the most commonly used languages in educational institutions and industries today.

In a study conducted by Bergin [1] in 1996 to compare C++ with Java, it was concluded that Java is an improved version of C++, by eliminating some of the difficult concepts of C++ such as Pointers.

3. Research Methodology

3.1 Project Design

The approach that was used is a survey and experimental design. The stratified survey approach was used. For the survey part, the ques-

tionnaires contain closed questions, while for the experimental design, there were open questions whose answers are the program listing for the questions. Questionnaires were administered to respondents using any of the four languages: C++, Java, Visual Basic, and other structured languages (like Pascal or Basic) in different study groups. This is necessary in order to solicit information from a range of users of these programming languages.

The required information includes: ease of learning, ease of use under pressure, suitability for complex jobs, problems commonly encountered, programming language preferences, flexibility and efficiency.

For the experimental part, respondents were given a small problem to be solved using each of the programming languages; C++, Java, Visual Basic, and any other structured languages (like Pascal or Basic). The Line of Codes (LOC) for each solution in each language was measured.

3.2 Population

The population for the survey is all programmers and students using the four programming languages in their works and studies. Due to financial and time constraints, this was limited to respondents in software organizations within the Gauteng region as well as web lecturers, programmers, researchers in major higher institutions in the Gauteng region in South Africa.

At an estimated value of 3 programmers per organization, and 100 software organizations in the Gauteng region, we have about 300 subjects in software organizations. At an estimated value of 10 lecturers with 40 students per institution, and 10 higher institutions in the region, we have about 250 subjects in South African higher institutions. These give a total of about 800 subjects for the population.

The intended population is stratified into 8 groups from the Gauteng region, consisting of students currently using C++, Java, Visual

Basic, structured Programming languages like Pascal, BASIC or C, as well as experienced or Post Graduate users of these languages.

This is necessary in order to have an adequate representation of students and experienced users of the various programming languages.

3.3 Sampling Method

As stated by Corbetta [2], sampling is the procedure through which we pick out, from a set of units that make up the object of survey (the population), a limited number of cases (sample) chosen according to criteria that enable the results obtained by studying the sample to be extrapolated to the whole population. One approach to counter human error is to randomly split the participants into groups and treat all groups exactly the same in all other respects. The population is divided into strata and samples taken randomly.

According to Corbetta [2], for a 95% confidence level, a population size of about 800 requires a sample size of 300. However, given the scarcity and the busy schedules of programmers in industries, we can reduce the confidence level to about 80%. Therefore a sample size of about 100 would suffice for our population size of about 800.

3.4 Data Collection

A total of 160 questionnaires were distributed. Since each questionnaire contains questions pertaining to the four programming languages, if each questionnaire were fully completed, a total of 100 completed questionnaires are required. However, most respondents are familiar with 2 or 3 programming languages. In all, a total of 110 questionnaires were duly completed and returned.

3.5 Hypothesis

The null hypothesis is that the low pass rate for C++ is due to C++ being an exceptionally difficult and complex language, as there is likelihood of significant difference in the 'Ease of learning', 'Ease of use', and 'Suitability for complex task' features for C++ as compared to other programming subjects such as Pascal, Visual Basic and Java.

3.6 Limitations

Due to the time frame and the cost of traveling, this study is limited to respondents from the Gauteng region.

4. Results

4.1 Ease of Learning

From the data in Table 1, twenty three percent of the respondents found Visual Basic very easy to learn. From usage experience in my personal view, this is likely to be true because VB has the "drag and drop program generating" facility in which you simply drag an object such as a Combo box unto the desktop, and then VB translate to equivalent program lines.

Next to VB, people found Pascal very easy to learn. This is true as Pascal has simple structured programming constructs.

Nine percent of respondents found C++ and Java similar in terms of ease of learning. This could be true as they have similar language constructs.

Table 1: Ease of learning the programming languages

	C++	Java	Visual Basic	Pascal
Very Easy	11	10	25	15
Relatively easy	10	45	35	5
Okay	54	20	35	25
Difficult	15	5	5	5
Very difficult	0	0	0	0
Don't know	10	20	10	50
No response	10	10	0	10

Forty one percent of the respondents found Java relatively easy. This could be explained due to the fact that Java is considered as "C++ without the pointer problem". Fifty percent of the respondents found C++ just okay. This fact is supported by the fact that for simple programs, C++ has few lines of code (LOC).

Fourteen percent of the respondents found C++ difficult. This is basically supported by the low previous pass rates at TNG which necessitated this study.

Forty five percent of the respondents don't know about Java. This is because Java is relatively new and only few programmers in "well-paid jobs" can afford specialized training in Java.

4.2 Ease of Use under Pressure

From the data in Table 2, thirty six percent of the respondents found Visual Basic very suitable for use under pressure. This is true because from experience, Visual Basic also has simple programming constructs as well as the "drag and drop program generating" facility in which you simply drag an object such as a Combo box unto the desktop, and then VD translate to equivalent program lines. This is followed by C++ which has fewer LOC than C++.

Table 2: Ease of use under pressure

	C++	Java	Visual Basic	Pascal
Not suitable	10	0	0	10
Manageable	10	25	10	10
Suitable	30	25	30	35
Good	15	30	20	10
Very suitable	25	20	40	0
No response	20	10	10	45

4.3 Suitability for Complex Jobs

From the data in Table 3, fifty percent of the respondents considered Java to be very suitable for complex jobs. This is true as Java is more flexible in linking to other current technologies such as the web-based tools like HTML and XML, and also web-based databases. This is the reason why there is increasing demand for Java programmers today. It is interesting to note that even 9% of respondents who gave no response on Java in terms of ease of learning above, still consider Java very suitable for complex jobs as they would at least have seen much advertisement for Java programmers.

Table 3: Suitability for complex jobs

	C++	Java	Visual Basic	Pascal
Not suitable	5	0	0	10
Manageable	10	10	5	15
Suitable	35	20	30	20
Good	15	25	25	20
Very suitable	40	55	50	5
No response	5	0	0	35

4.4 Overall Rating for the Languages

From the data in Table 4, C++ and Visual Basic are considered to be equally best by 41% of the respondents, possibly due to its use in most schools today. Java comes next at 27%, probably due to the fact that it is newly being introduced in most schools.

Table 4: Overall rating for the programming languages

	C++	Java	Visual Basic	Pascal
Best	45	30	45	10
Better	15	25	10	0
Good	20	15	25	40
Bad	0	0	5	10
Worst	10	25	15	10
No response	20	15	10	40

4.5 Response Data Clustering

To obtain a clearer picture of the responses, it is necessary to group the scaling system. The information is then presented in the form of bar charts as given below:

4.5.1 Ease of Learning

From Figure 1 below, we can see that a total of 95 respondents indicate VB to be either very easy, relatively easy or okay to learn as compared to 75 responses for each of C++ and Java. Again this is probably true because VB has the "drag and drop program generating" facility in which you simply drag an object such as a Combo box unto the desktop, and then VB translate to equivalent program lines.

Pascal however, tops in the "Don't know + No response" bar. This could be due to the fact that it is outdated by now.

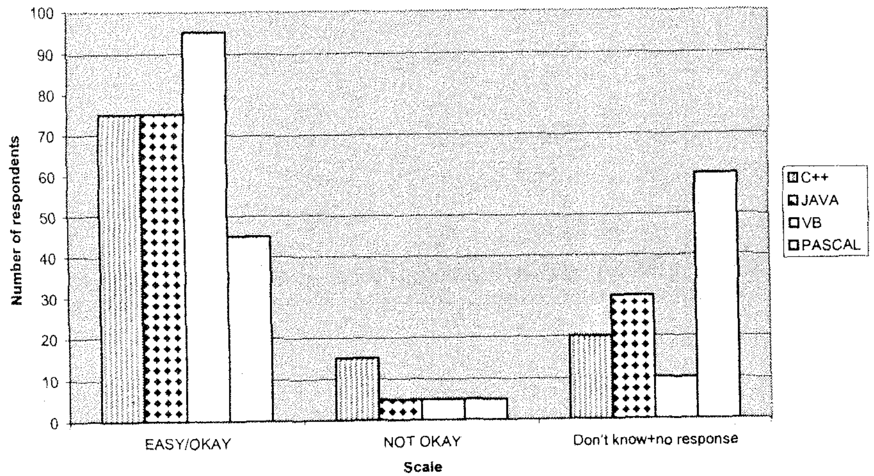


Figure 1: Ease of Learning

4.5.2 Ease of Use under Pressure

From Figure 2 below, more respondents (90) indicate VB to be easier to use under pressure, followed by Java and then C++. This is due to the "drag and drop facility" in VB as well as the windows development environment for both VB and Java which, in addition, also has Microsoft Disk Operating Systems (MSDOS) executing modes.

Pascal however runs mostly from the MSDOS prompt. To use it effectively, one has to be familiar with MSDOS commands and MSDOS development environment tools.

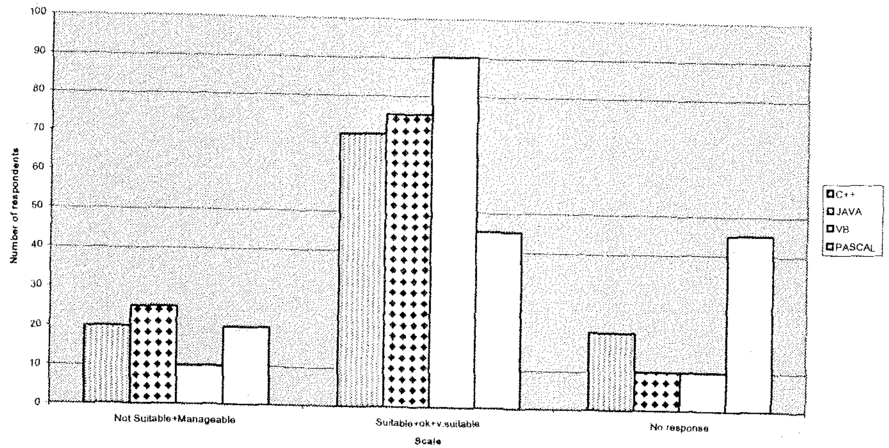


Figure 2: The ease of use under pressure

4.5.3 Suitability for Complex Jobs

From Figure 3 below, 90 respondents indicate VB to be best suitable for complex jobs, followed by Java and then C++. This is due to the fact that VB and Java can connect to other software tools like internet web pages and databases. VB does that via VB script and Java does that via javascript, servlets, and other middlewares.

Again, Pascal is less suitable for complex jobs, as those new tools were not yet available when Pascal came to being.

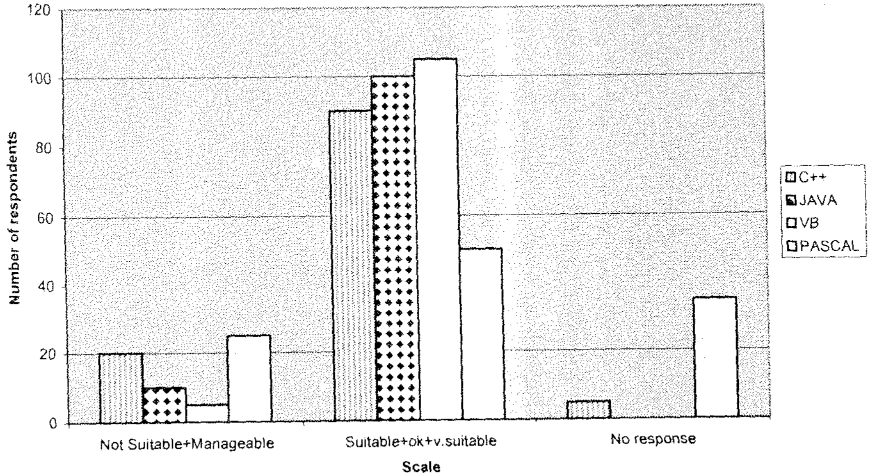


Figure 3: Suitability for complex jobs

4.5.4 Overall Rating for the Languages

From Figure 4 below, C++ and Visual Basic are considered to be equally best by 80 respondents. This is due to the fact that these two languages are the ones being currently used in most schools today. Java comes next with 70 responses, probably due to the fact that it is newly being introduced in most schools and in the industry.

With respect to the overall rating above, here again Pascal is rated the lowest as a programming language.

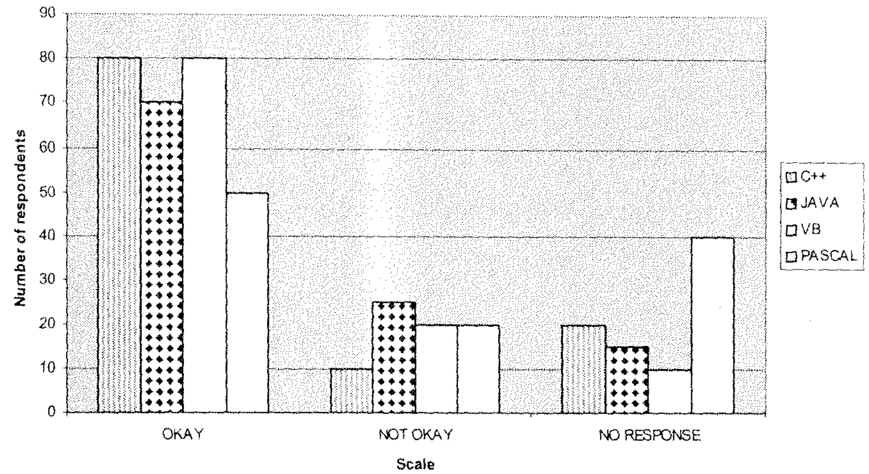


Figure 4: The overall rating for the programming languages

4.6 Line of Code (LOC) Estimation

From the completed questionnaires, the line of code was estimated for each programming language's solution to the given problem. The minimum line of code for each solution is given below:

Java: LOC = 10

```
import java.lancs.*;
public class CalculateArea
{
public static void main (String[]args)throws Exception
{
int length, breath;
Basiclo.prompt("PLEASE ENTER THE LENGTH OF THE AREA");
length=Basiclo.readInteger();
Basiclo.prompt("PLEASE ENTER THE BREATH OF THE AREA");
breath=Basiclo.readInteger();
System.out.println("THE AREA OF RECTANGLE =" + length * breath + "|");
}//end of method main
}//end of class CalculateArea
```

VB: LOC = 8

```
Private Sub cmdCalculate_Click()
Dim intLength As Integer, intbreadth As Integer
Dim IngArea As Long
intLength = Val(txtLength.Text)
intbreadth = Val(txtBreadth.Text)
IngArea = intLength * intbreadth
txtarea.Text = IngArea
End Sub
```

This however is in addition to the "drag and drop facility" being used to design the input and output form as illustrated in the figure 5 below:

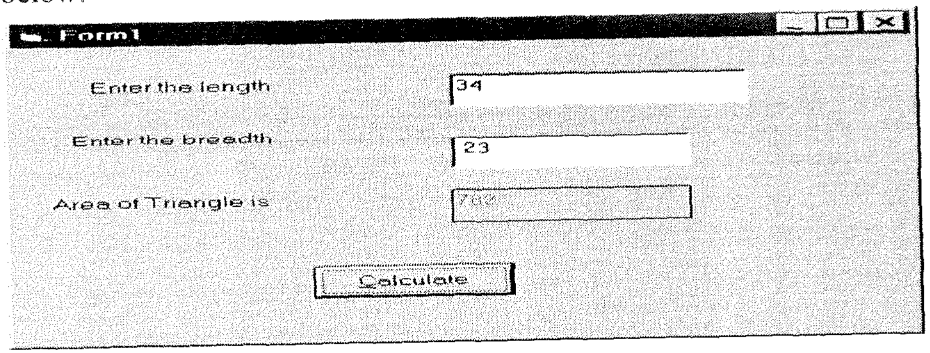


Figure 5: The effect of the drag and drop facility

C++: LOC = 9

```
#include <iostream.h>
main ()
{
        int length, breadth;
        cout << "Enter length of triangle";
        cin >> length;
        cout << "Enter breadth of triangle";
        cin >> breadth;
```



```
        cout << "The area of the triangle is "<< length * breadth<<endl;
        return 0;
}
```

Pascal: LOC = 10

```
Program calcArea (input, output)
var
  length, breadth, area  : integer;
begin
  writeln ( 'Enter length of triangle')
  readln ( length );
 writeln ( 'Enter breadth of triangle')
 readln ( breadth );
 write ( 'The area of the triangle is  ',  length * breadth);
end
```

VB has the smallest LOC. In addition to the "drag and drop facility", this makes it easier to learn and use under pressure. C++ and Pascal also have reasonable LOC (8) minus the "begin" and "end" statement, making them easy to learn. Java however, has the longest LOC, making it more difficult for beginners to learn. However, this is compensated for by the flexibility it offers for complex jobs.

5. Conclusion

It was found that Pascal is simple to write for beginners, but not suitable for complex tasks.

VB is easy to use under work pressure possibly due to the "drag and drop program generating" facility. Additionally, VB has the smallest LOC. In addition to the "drag and drop facility", this makes it easier to learn and use under pressure. C++ and Pascal also have reasonable LOC (8) minus the "begin" and "end" statements, making them easy to learn. Java however has the longest LOC, making it more difficult for beginners to learn. Like VB, Java is suitable for complex jobs and is considered very flexible as it interfaces with other web technologies like Java Servlets, and other middleware platforms.

The above findings therefore show that no single language can adequately satisfy all the requirements. But a careful combination of the languages can achieve the desired result. The study therefore concluded that the low pass rate for C++ is not due to C++ being an exceptionally difficult language, as there is no significant difference in the factors studied for C++, Visual Basic and Java.

6. Recommendations

From the above findings and conclusion, C++, Java and VB are all recommendable. VB is recommended for its ease of use under pressure due to the "drag and drop program generating" facility. C++ is recommended because of the fact that for simple programs, C++ has few lines of code (LOC), which would make it easer to learn. Java is recommended for its flexibility and suitability for complex jobs.

In my own opinion and experience, since C++ and Java have similar constructs, it will be alright for students to start with C++ in the first year and as soon as it starts becoming difficult at the second year level, they should move to Java which can do all C++ can do and more with a "gentle" language constructs. Visual Basic can be taught parallel to either C++ or Java in either first or second year respectively. Thus, a systematic combination of the programming languages could achieve the desired result of enhanced comprehension and potential capacity for future complex challenges. Further studies and analyses are therefore needed to pinpoint the cause of the low pass rate in C++ in previous years.

Further studies and analyses were initiated to pinpoint the cause of the low pass rate in C++ in previous years. This need is communicated to the Instructional Offering Committee (IOC) for programming subjects, which deliberated extensively and came up with the followings:

Program Design should be immediately translated into programming language in the same subject and not in a separate subject. This

will therefore mean that C++ programming in the first year will be divided into two components namely:

- Program design plus the lexical structure, and control structures.
- Functions, Arrays and Pointers.

By so doing, program design will serve as a background to the major programming concepts. Yet, the implementation of design into programming should not be delayed, thereby avoiding situations where students think program design is separate from programming. This, we hope will improve the comprehension of the programming concepts, thus leading to improved pass rate.

The same principles were found to be applicable to second year subjects. Object-oriented Analysis and Design should precede object–oriented programming as one need to analyze a problem before solving the problem. Therefore, Object-oriented analysis and Design should be incorporated into the beginning part of both "Java object-oriented programming" and "C++ object-oriented programming". The program design should be immediately translated into the applicable programming language in the same subject and not in a separate subject.

Similar principles are applicable to Visual Basic instructional offering. By the end of the second year, the students are thus expected to be versatile in the C++, Visual Basic and Java. This is in line with the current trends in the market where web-based development involves combining database with VBScript and JavaScript in server-based programming environment.

7. References

1. Bergin, J 1996. *Object Technology in the Classroom - Java as a Better C++* ACM SIGPLAN Curricular Patterns, pp. 21-27. USA : Wiley.
2. Corbetta, P 2003. *Social Research: Theory, Methods and Techniques.* U.K: Sage Publications.

3. Dehinbo, J 2004. The Impact of Web-Based Middlewares on Training and Assessment through In-House Developed On-Line Testing System. *Proceedings of the Informing Science + Information Technology Education (I[n]SITE) Conference*, Australia. http://proceedings.informingscience.org/InSITE2004/027dehin.pdf .
4. Deitel, HM, PJ Deitel, & TR Neito 2001. *e-Business & e-Commerce: How to Program*. USA: Prentice Hall.
5. Hamilton, MA 1996. *Java and the Shift to Net-Centric Computing Practices* IEEE, pp. 31-39. USA.
6. Lerdorf, R. & K Tatroe 2002. *Programming PHP: Creating Dynamic Web Pages*. USA: O'Reilly & Associates Inc.
7. Sebesta, RW 1996. *Concepts of Programming Languages* 3rd Edition. USA: Addison-Wesley Publishing Company.
8. Zak, D 2001. *An Introduction to Programming with C++*. 2nd edition. USA: Course Technology – Thompson Learning.

Author's Contact Details

JO Dehinbo (DehinboOJ@tut.ac.za JDehinbo@yahoo.com)
Computer Studies Department
Tshwane University of Technology, Pretoria, South Africa

Introducing B.Com Information Systems Students to Basic Programming: Overcoming Some of the Difficulties

Lyrice Cohen

The objective of equipping B.Com Information Systems students with basic programming skills is not without its hurdles. Difficulties include large class sizes, varying levels of prior exposure and low motivation for technical learning. The problem is compounded by the standard (and not minor) challenges of teaching programming such as the challenges of teaching cognitive thinking and the need for individual interaction with the computer. Lecturers who are both technically adept at the current programming tools and genuinely concerned with enabling students to "see the light" are not so easy to come by.

This paper discusses and demonstrates an approach to dealing with some of these problems. This approach has been chosen because of its applicability to our specific context, where we have limited laboratory resources and are dealing with students from many diverse educational and technological backgrounds. Elements of the approach would, however, be relevant in all contexts.

The approach includes a programming concepts course that is run during lecture time, complemented by a series of detailed on-line tutorial sessions. A Visual Programming Tool is used for these tutorial sessions, exposing students to ideas of event-driven programming and the mechanics of using the tool. Through the visual environment, students also have practice with the programming logic they have learned

*Alter*nation 12.1b (2005) 615-625 ISSN 1023-1757

in the programming concepts course. The tutorials are consistent with the methods used by John Barrow et al [1]. Techniques are employed to encourage and motivate the students as they work through the tutorials, giving them a sense of empowerment with the technology. The tutorials have been prepared in detail, the longer term goal being for them to be re-used in future years, regardless of which lecturer is actually giving the course. A survey of student reaction to these tutorials indicates a positive response.

This paper consists of four sections. These discuss the scope of the course and the choice of the tool to be used, low student motivation for learning basic programming, student evaluation and the future.

Scope of the Course and the Choice of Tool to Be Used

It is important to point out at the outset that this approach has been developed specifically for First Year B.Com Information Systems students. The emphasis of the Information Systems course lies not on the understanding of intricate advanced programming concepts, but rather on the deployment of Information Technology in a way that will further the objectives of the business. The Information Systems student, therefore, need only attain an understanding of the fundamental principles of programming and a working knowledge of a visual tool that is sufficient for him / her to develop the Second Year project, which is a simple Information Systems application. This understanding and working knowledge should provide the student with a level of confidence about programming that will stand him / her in good stead in whatever commercial environment he /she chooses to go into. It is not considered to be necessary for the First Year Information Systems student to acquire in-depth advanced programming skills.

Bearing this in mind, the choice of the particular Visual Tool to be used for the tutorials was not considered to be of prime importance. With a view to using a product that is extensively used in the industry and therefore constitutes a marketable skill, a Java option was considered and then discarded due to problems experienced[1]. Visual Basic was then turned to as another marketable option.

The final decision was to use Visual Basic for Applications in the Microsoft Access environment (see, for example [2]). This choice proved to be advantageous for a number of reasons. Firstly, the programming tutorials are done just as the students have finished learning Microsoft Access. They are now familiar with the Microsoft Access environment and it is a natural progression for them to learn Visual Basic in that environment. Secondly, Visual Basic for Applications is a natural precursor to the Visual Basic.Net course they will be doing in the following year of study. Thirdly, Microsoft Access is a stable product that has been installed in our computer laboratories for a number of years. It does not have a high memory requirement and gives no installation problems. This makes laboratory sessions using limited facilities for large classes much easier to manage. A further advantage in this regard is the fact that most students who have computers at home also have Microsoft Access installed on those computers. They can therefore work through the tutorials at home – freeing up the computer laboratories for those who do not have computers at home, a common problem [3].

Once the tool was decided upon attention could be turned to the mode of presentation of the actual tutorials. After careful consideration it was realized that the mode of presentation, if carefully constructed, could make a major contribution to addressing one of the difficulties discussed earlier, i.e. low motivation of B.Com students for technical learning.

[1] The intention was to use a Java tool called "Sun-One" that was downloaded from the Internet. This was, however, found to be an advanced IDE rather than a fully developed Visual Development environment. In addition serious problems with product support were encountered, causing it to not be a viable option.

Low Student Motivation for Learning Basic Programming

As mentioned previously, one of the difficulties that have been encountered in past years of teaching programming to B.Com students is the low motivation levels of these students when it comes to learning how to program. The fact that programming is not considered to be of prime importance in the Information Systems curriculum means that insufficient weight is given to it to make the acquisition of the programming skill an essential part of attaining the credit. This enables many students to "prioritise it out of the picture" – to pay minimal attention to it and get through their degrees without passing the programming part of the course. For this reason it was considered important to establish the possible causes for these low motivation levels and to work on ways of combating these.

Four main areas were identified here: manipulating student priorities, fear / phobia / resistance to programming, maintaining student interest and dealing with complexity.

Manipulating Student Priorities

Perhaps the most obvious cause of students paying insufficient attention to the programming course is the fact that it requires a lot of time and effort to learn how to program. It is not a skill that can be attained quickly and easily or by doing some light reading. Students have many conflicting priorities to achieve in a limited time period, and they will tend to select those things for which their "investment" in time and effort will have the greatest perceived "return". The major currency of this "return" in our institution is marks, and we need to use this as a way of manipulating student priorities, i.e. of getting them to do the programming tutorials.

Last year, a total of 6 of these tutorial sessions were run. At the end of each tutorial session there was an assignment to be handed in. Only 2 of these assignments were marked, but the students did not know which ones they would be, so they had to do them all. In addition, students lost 1/6 of the achieved mark for each assignment that they neglected to hand in. This technique has been used previously for several years for another course, and has shown positive results.

Fear / Phobia / Resistance to Programming

A phenomenon that appears to be prevalent amongst B.Com students that also contributes to low levels of motivation for technical learning is an almost tangible fear / phobia / resistance to programming. Evidence for this can be heard in some common comments coming from Third Year Information Systems students, like "Programming isn't my thing – I cope OK with the other stuff." or "I'm not good at the programming part – I leave that up to the other members of the project team." Last year, introducing this practical programming course to the first year students and explaining that it would involve a large amount of individual interaction with the computer, I asked the class how they felt about embarking on this experience. One of the ladies in the front row replied, wide-eyed, with one word – "terrified!". It is my opinion that her reply echoed the sentiments of many of the other students in the lecture hall at that time.

It would appear that the source of this fear / phobia / resistance to programming is the fear of the individual that they will not be able to sufficiently master the programming skill. This perceived inability leads them to have a very low self-image / self-esteem as a programmer. As always with low self-esteem – we don't want to do what we are not good at – the result is a low motivation for learning how to program. Expecting the students to do programming exercises that are out of their reach or that they have not been adequately prepared for serves to further exacerbate the problem.

Positive self-esteem / self-image as a programmer can only be encouraged with positive achievements, however small. A meaningful objective, therefore, would be to give the students ample opportunities to perform small, achievable tasks with visible results. These tasks would, of course, have to have at least some substance – there would have to be something, however small, in each one that would attempt to "stretch" the student, and make them work out something they had not previously known or apply something they have just learned to a new situation. It is contended here that each of these tasks would provide for a small triumph that would give the student a sense of achievement, perhaps even of excitement, and that this in turn would lead to an improved self-image / self-esteem as a programmer, which would in turn motivate and encourage the student in their learning effort.

Maintaining Student Interest

Many a student is put off from the idea of programming by the perception that it is a boring activity. The common stereotype of a studious "nerd" with glasses who sits in front of a computer all day prevails. We need to break this stereotype and show that programming can be fun,. This can be done in part by careful choice of exercises – games, for example – and by introducing humour. The tone of the text of the tutorials should be informal, friendly and encouraging and it is preferable for the text to be presented in an informal font that is pleasant for the students to work with. Starting each tutorial with a short synopsis that lists the tutorial's objectives also helps to maintain interest.

Care should be taken not to lose the students' attention during the course of the tutorials by indulging in too much detail. It is not necessary, as we uncover each successive cognitive or programming language concept, to be exhaustive and go through every option available. Students do not need to be immediately aware of every option available, they need to be able to understand and apply the concept. An exhaustive discussion of every variable type, for example, can be quite boring and quite meaningless too. It is far more meaningful to discuss a different variable type in each different tutorial and experiment with its application and explore other related aspects of programming before progressing onto the next variable type.

Dealing with Complexity

Programming is a complex activity, and teaching programming involves developing within the student the capability to master this complexity. We are effectively teaching cognitive thinking and we need to look at ways of doing this most effectively.

The human mind is capable of learning only one new thing at a time. It is essential, therefore, that we break up the complexity into manageable chunks and approach the material in a step by step manner. Each "chunk" or step must be fully understood before moving onto the next step, and the only way to do this is to expect the student to apply what they have just learned and to require them to be "stretched" a little. Too many texts that portend to teach computer-based tools fall into the trap of merely instructing the students, step by step, mouse-click by mouse-click , how to achieve the task, never discussing the concepts involved and never expecting an independent application of what has been learned, and leaving the students not too much the wiser than when they started out. It is absolutely vital, in whatever way possible, to encourage the students to think, internalize what they have learned and to be curious about the workings of the program and the programming environment.

Concepts or "steps" previously learned also need to be reinforced with repetition. Using previously learned programming constructs in later exercises is a natural way of doing this.

At all times we should be attempting to move from what is "known" to the student to what is "unknown" to them, so that they start off with a secure basis for understanding. One technique of doing this is to first demonstrate the effect of a program by executing it, and then afterwards to examine the code and explain how it works. Then, of course, the student must be asked to apply what he has just learned. Barrow et al [1] use this way of teaching extensively, referring to it as the "experiential" model.

Unnecessary complexity should be removed wherever possible, so that students are able to focus on one particular concept at a time. In a practical programming environment it is often necessary to complete a whole lot of prerequisites before getting a programming construct, say an "If" statement, to work. These prerequisites require the time and attention of the student and detract from the main issue, i.e. understanding the programming construct.

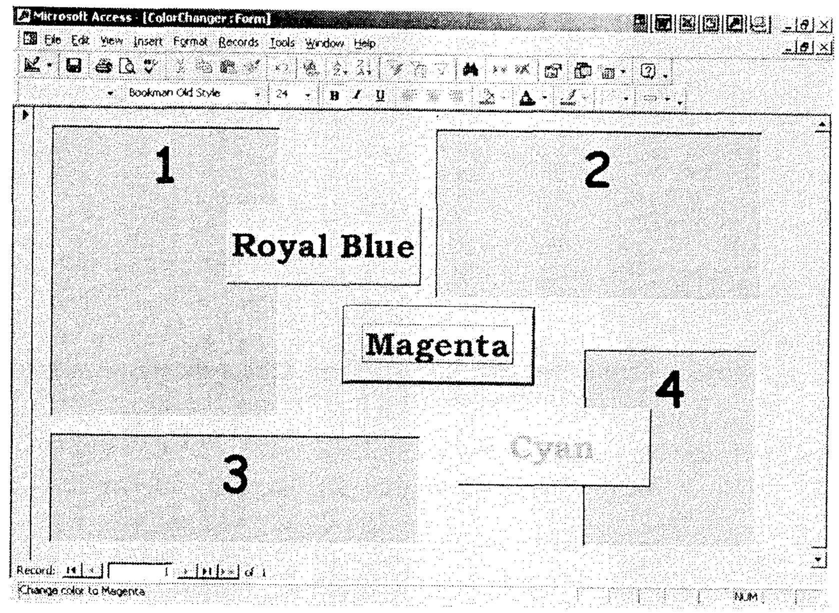
An attempt has been made in these tutorials to find ways of doing these prerequisites for the student wherever possible, so that they can concentrate on understanding the next important concept, say, the "If" statement. For example, the students are provided with ready-formatted forms to work from. They are never asked to develop their own forms. Formatting forms is a time-consuming activity and not a particularly difficult one, and one that they should already have mastered in their Microsoft Access course. Precious computer laboratory time is better spent on the programming itself. Certain subprograms or small pieces of code that enable them to achieve things that they are not yet able to do on their own have also been provided to them. Several constants have also been predefined for them to make the programming easier to understand. For example, constants have been used to associate the names of colours with the lengthy numbers that Microsoft Access uses to represent those colours.

To further remove distraction and allow them to focus on absorbing the concepts students are reassured that they are not expected to understand everything they see on the screen initially and the things they can safely ignore for now are continually pointed out to them. Particular care has been taken to avoid difficult vocabulary and the tutorials are printed in a large font with wide spacing, all of which are believed to facilitate understanding.

Perhaps the most useful part of this approach in terms of conquering complexity is the fact that a Visual Tool is being used as the medium of teaching. Working in the Visual environment does increase the complexity and the amount that has to be learned, as now the stu-

dent has to master both the programming concepts and the Visual tool. The visual aspect and the immediacy of the environment, however, can be used to enhance understanding.

As an example, let us consider the first of the tutorials. The students are asked to display the following form:



As you can see, this is a simple form with four text boxes (numbered 1 to 4) and 3 command buttons. The students are asked to click in different places all over the form and observe what happens. They soon discover that nothing happens when they click anywhere other than on the command buttons, and that when they click on a particular command button the colour of all the text boxes changes to that button's colour. It is pointed out to them that this happens not before they click, not after they click, but as they click. That click is the *event* that made the colour change happen. Actually somewhere there is a little program called an *event handler* that made the colour change.

It can be seen here that the students have just used the visual environment to demonstrate very quickly to themselves the concept of event driven programming. This kind of live "self-demonstration" can be used in many ways to enhance student understanding.

The extent of the visual impact of what is happening on the form is also thought to contribute to the understanding process. This is why the form displayed was designed to take up the whole screen, and the text boxes that change colour are so large.

Student Evaluation

An evaluation questionnaire was handed out to students to determine their reaction to the course. The response was quite positive, with 80% of students indicating that they now feel more enthusiastic about programming, 72% are feeling less afraid of programming, 65% feel they learned a lot and 60% said they actually enjoyed the course. Surprisingly, though they are feeling more enthusiastic and less afraid, only 42% of them are feeling more confident about learning how to program. This could possibly be due to the fact that this was the first time the course was run, and since there were only six sessions, insufficient ground was covered in order for them to really gain confidence.

The Future

Though the approach adopted appears to have been successful, at least to some degree, it is still considered to be a work in progress. It will be continued with during this year and every effort will be made to improve upon it as much as possible, taking student evaluations and lessons learned from experience into account. This year, the course will be starting earlier, and tutorial content will be adjusted to be more appropriate for the time available. An additional 3 computer lab sessions have also been made available, bringing the total to 9. Tutorial

content will also be manipulated and worked upon to even better implement the principles and ideas discussed in this paper.

In closing, this approach appears to have been a step in a positive direction towards resolving many of the difficulties that we have experienced in teaching programming to B.Com Information Systems students. With further effort and development, we will hopefully finally realize our objective of enabling First Year Information Systems students to be motivated to learn programming and empowered with a level of confidence about programming in a visual environment.

References

Barrow, J, H Gelderblom & L Miller 2004. *Introducing Delphi: Theory Through Practice.* Oxford: Oxford University Press.

Zak, D 2001. *Microsoft Visual Basic for Applications.* Thomson Learning.

Winchiers, H 1999. Reflections on Teaching Computer Science to Previously Disadvantaged Students. *SACLA '99*

Author's Contact Details

Lyrice Cohen (lyricec@isys.wits.ac.za)
School of Economic and Business Sciences
University of the Witwatersrand, Johannesburg, South Africa

The Changing Face of E-Commerce in South Africa: 2001 - 2004

Anesh Maniraj Singh

Abstract

E-Commerce has revolutionized business the world over. South African business has been revolutionized as well. The Internet has created new income generation streams, made communication cheaper, easier and more efficient and the Internet has also provided a new medium for competition and collaboration alike. However, online retailing has seen a slowdown in South Africa since 2001. The aim of this study was to determine whether there have been changes in E-Commerce from 2001 to 2004, and what these changes were. A questionnaire was sent to 122 companies with an Internet presence. This study has revealed that there have been improvements in E-Commerce in South Africa with respect to income generation, customer service and cost reduction. However, it is evident that E-Commerce has not reached its full potential and needs to be given greater strategic importance within corporate plans. Other recommendations included: designing effective web sites, improving online security, expanding infrastructure to a wider audience, improving education and developing an e-culture in the country. This study has shown that there is immense growth potential for E-Commerce in South Africa. However, companies need to be more creative in how they harness the power of the Internet. This challenge is not one for private companies alone but requires public, private partnerships to ensure long term improvements.

Keywords

E-Commerce in South Africa, B2B E-Commerce, B2C E-Commerce, Generating Online Sales, Impact of the Internet on Business

*Alter*nation 12.1b (2005) 627-649 *ISSN 1023-1757*

1 Introduction

The Internet is often referred to as a revolution rather than an evolution due to the manner in which it has taken the world and more so the business world by storm. The Internet has radically changed the way business is conducted from a bricks and mortar to a clicks and mortar format. Trade is no longer limited by the boundaries of space and time. Virtual stores make money even when physical stores are closed and customers enjoy the convenience of shopping and transacting from the safety and comfort of their homes. Although largely prevalent in Western first world countries, the Internet and its main application E-Commerce are growing in third world countries. African and more especially South African organisations are rapidly adopting E-Commerce to replace, supplement or complement their physical outlets. Business-to-business (B2B) transactions have proven to be very profitable whereas business-to-consumer (B2C) transactions has not taken off, like it has elsewhere in the world, due to numerous challenges including access to the Internet and literacy. This paper examines the changes that have taken place in E-Commerce in South Africa between 2001 and 2004, from a management perspective.

2 Background

2.1 What is E-Commerce?

There are a number of definitions that are used to describe E-Commerce which includes; the conducting of business communication and transactions over networks and through computers, a term for all kinds of business that are established electronically especially over the Internet, it refers to shopping on the World Wide Web (Schneider 2004). Electronic Commerce suggests the buying and selling of goods and services using all electronic media including cellphones, kiosks, fax machines and the Internet. Internet commerce, however, is the buying and selling of goods and services only on the Internet. Although there is a difference between E-Commerce and I-Commerce, E-Commerce is generally used to describe transacting on the Internet.

2.2 Advantages of E-Commerce

The Internet is a new means of making profit. However, it has helped organisations improve customer satisfaction by offering them twenty four hour convenience and customer care. Other advantages include:

Lower overheads: E-businesses no longer need to carry large inventories and as such require less space which saves on rent. Brochure printing costs are obliterated using online brochure-ware. Due to businesses becoming smaller, fewer staff is required.

Focused markets: According to Barker & Gronne (1996), the Internet has the greatest focus of all advertising media and is targeted specific markets. This ensures that the advertising message is delivered to the intended audience rather than wasting efforts on mass audiences.

Reducing transaction costs: According to Gordon et al (Turban, 2004), using technology to conduct business reduces the need for paper, records can be stored on databases and backed-up onto media such as CD-ROM's, orders can be placed by the customer directly into an online ordering system, reducing the need for, and cost of order clerks. Customers place their own orders which reduces the liability costs for staff related errors or omissions.

One of the greatest benefits of E-Commerce is that the customer is in control of the buying experience. He/she can choose when or where they purchase without any interference from other shoppers and intruding sales people.

2.3 Challenges Facing E-Commerce

As much as the Internet has benefited E-Commerce, it also introduces new challenges that managers have to be wary of and take measures to counter them. Some of the common problems include: security of transactions, threats from cyber criminals, privacy of the organisation and its clientele, unfulfilled deliveries. Other issues include trust, culture, language and infrastructure (Schneider 2004). Lesser known issues revolve around deviant behavior such as the abuse of an organisations Internet facilities and addiction to the Internet (Singh 2004a). Turban (2004) classifies some of the challenges as technological and non-technological limitations. One of the technological limitations he identifies is the high cost and inconvenience of accessibility to the Internet. The cost of connectivity in South Africa is extremely high for the home user, as the dial-up service is controlled by a monopoly. One of the non-technological limitations identified by Turban (2004) was the issue of trust. People do not as yet trust paperless, faceless transactions. According to Singh (2001), South African Internet users did not engage in E-Commerce because they did not trust the Internet and their own ability to use the technology properly, language and literacy posed a major problem as the language of the Web is English which is not the mother tongue of the majority of South Africans many of whom are illiterate, and the lack of access to the infrastructure also made E-Commerce inaccessible to the South African consumer. Mc Kinnell (2000) reported that South African shoppers have no motivation to shop online, because it is more difficult, and even after discounts, some sites are more expensive than purchasing at the local mall. Other complaints from users were the lack of delivery from suppliers, and online payment as most of the population did not have credit cards (Madonko 2000).

In order to be successful in E-Commerce in foreign markets, Schneider (2004) suggests that businesses have to "think globally, act locally". Many web sites are taking this advice and changing their approach. The world's leading search engine Google offers searches in four of South Africa's eleven official languages.

2.4 Who engages in E-Commerce?

The three players in the E-Market are businesses, consumers and governments. Governments do not sell their services online; however, they purchase labour, goods and services online. The business, consumer relationship has given rise to four classifications based on the

nature of transactions between the parties as illustrated in Figure 1. It is evident from Figure 1 that all the relationships have always been in existence; they have just been given three letter acronyms which is the norm in the IT industry. Businesses that sell to other businesses are generally manufacturers selling to resellers or wholesalers selling to retailers. B2C is the relationship between the seller and the end user of the products or service.

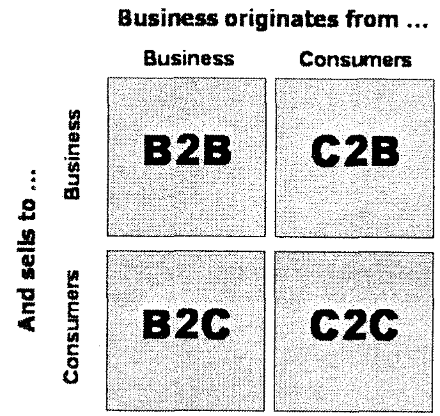


Figure 1: E-Commerce Classifications Adapted from: Rayport & Jaworski, 2002

C2C has also been a popular means of transacting using the classified sections of newspapers and magazines. C2B transactions also take place in the real world especially where people sell their labour to industry. These classifications are important when measuring the success or failure of E-Commerce.

2.5 Current Trends in E-Commerce

E-Commerce initially took off with great expectations from businesses and investors alike. The turn of the century saw a number of failing dot.com businesses. Some of the reasons for the failure of these organisations include: products were not appropriate for the

Internet, they did not deliver customer value, they failed to develop a profitable business model, a number of bad ideas were proposed and there was an over-reliance on advertising as a revenue source (Schneider 2004, Turban 2004). According to Strauss et al (2003), the downturn in E-Commerce was termed the trough of disillusionment. However, they have predicted that 2004 into 2005 would signal a slope of enlightenment seeing a trend towards a positive cash flow and 2006 onwards would see the beginnings of a plateau of profitability where E-Business just becomes business. Turban (2004) calls this turnaround the second wave. One of the characteristics of the second wave is the reduced hype and a focus on the basics of business. A major Internet trend recently has been the growth in global Internet users as illustrated in Figure 2.

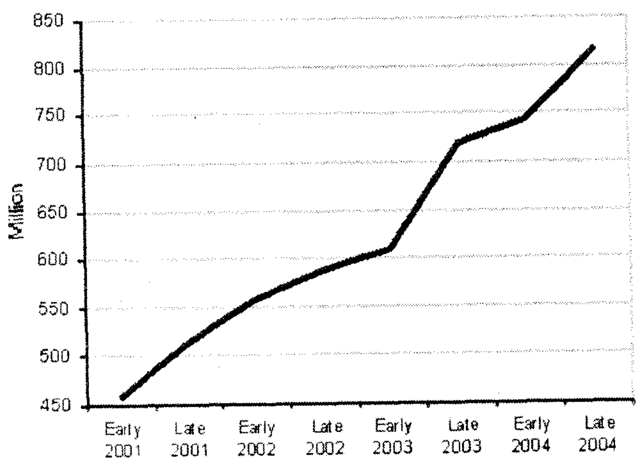


Figure 2: Growth of Global Internet Users Adapted from: Internet Growth Statistics, 2005.

It is evident from Figure 2 that the number of Global Internet users has grown steadily since 2001with just over 500 million users to 719 million users in early 2004. From early 2002 to the same time in 2003, growth tended to flatten out. However, late 2003 saw a rapid increase once again. Late 2004 recorded almost 1 million new users. Total Internet users for 2004 accounted for 12.7% of the world's population (Internet Growth Statistics ... 2005). For statisticians it would seem that 12.7% of the world's population is not a significant market. However, for online sellers, the growth in users signifies a growing potential market. Due to the growth in users, it could be assumed that online sales would grow as well. Table 1 illustrates the growth pattern of global E-Commerce. It is evident that like the user growth, there has been a steady growth in total online revenues.

Year	B2C US$	B2B US$	Total US$
2001	90 bn	240bn	330 bn
2002	180 bn	410bn	590 bn
2003	220 bn	680bn	900 bn
2004	300 bn	1100bn	1400 bn

Table 1 Online Purchase Revenues 2001 – 2004
Adapted from: Statistics for Online Purchases, 2005.

Business to consumer (B2C) revenues has also risen steadily from $90 billion in 2001 to $300 billion in 2004. In South Africa, the number of online retailers has grown from 215 in 2001 to approximately 719 in 2003 and total online retail sales has grown from R162 -million to R341-million over the same period (Goldstuck 2004). It is evident that the growth in online sales in South Africa has been slow, due to issues previously mentioned such as trust, cost of connectivity, and inadequate infrastructure and access to it. However, according to Goldstuck (2004), the online market is dominated by the top eight online retailers who account for some 80% of online sales. Two of these retailers namely Pick 'n Pay and Woolworths happen to be giants in the offline world as well.

It is evident that globally and in local economies, there have been positive changes in E-Commerce, which raises the question "what changes have firms experienced?"

3. Methodology

3.1 Motivation for the Study

Singh (2001) found that although South African organisations were present online, they were not making optimal use of the advantages offered by the Internet. According to Goldstuck (2004), there has been a slowdown in online retail in South Africa. However, he has shown that there has been an upward trend between 2001 and 2003. This study aims to determine what changes, if any, firms have experienced since 2001.

3.2 Research Problem and Objectives of the Study

The aim of this study was to determine what changes firms have experienced since 2001. In answering this question, this study sought to determine:

- The reasons why businesses used the Internet
- What impact the Internet had on income generation, cost saving, and customer service
- Management impressions of online business
- Recommendations to improve online sales

3.3 Research Tool

A questionnaire that was developed by Singh (2001) was used for this study in order to draw comparisons between the two studies. The questionnaire was composed of open ended and closed questions that the researcher had devised to test criteria such as reasons for using the

Internet for business, impact of the Internet on business and managers impressions of online business. The questionnaire was hosted online which was linked to a database. Both these approaches were improvements on the original questionnaire that was sent by post and captured manually.

3.4 Sampling

A convenience sample of 250 companies with an Internet presence was chosen for this study. Only 122 managers responded. The sample was slightly larger than that used in 2001 and included 60 of the original companies that participated in that study. Of the nineteen original companies that did not respond 10 had closed down their online operations. An e-mail together with a link to the questionnaire was sent to managers of the firms involved in the study.

4. Results

4.1 Composition of the Sample

Figure 3 illustrates the industry's in which the firms operated. A significant difference between the two studies shows that more Educational institutions (28.7%) had an online presence, than organisations from other sectors. Similarly, there were more financial institutions in the 2004 study (15.6%) than those in the 2001 study (8.9%). However, there were more manufacturers in the 2001 study (25%) than there were in the 2004 study (15%). Retail and Computers showed minimal changes between the two studies.

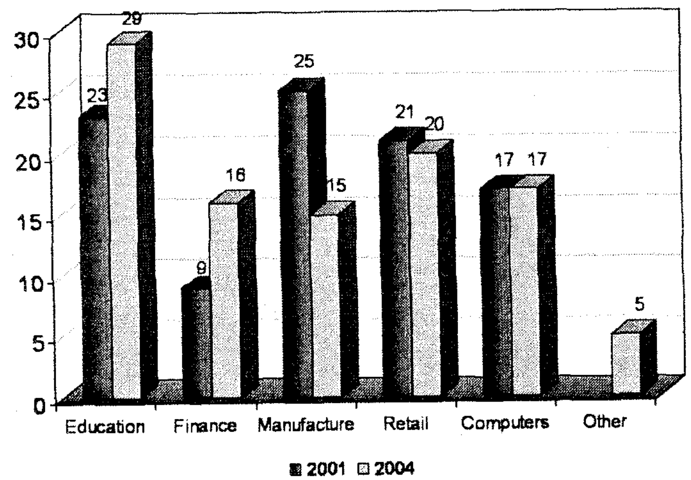


Figure 3: Composition of the Sample

A new category of other represents 5% of the sample; however, these were unspecified making it difficult to determine what sector they came from.

4.2 Reasons for Conducting Business Online

According to Gow (1997), America's top 100 companies went online to achieve cost savings, improve customer care, generate new revenue and conduct marketing among other reasons. Figure 4 illustrates the reasons South African businesses conduct business online. Three categories namely cost saving (23%), customer service (19%) and marketing (18%) have decreased from the 2001 study as reasons for being online. However, income generation (17%) and other (23%) have become important reasons for being online. Some of the other reasons for being online included: extending the reach of the organisation, reducing transaction costs, reducing paper work and paper flow, and cutting back on staff.

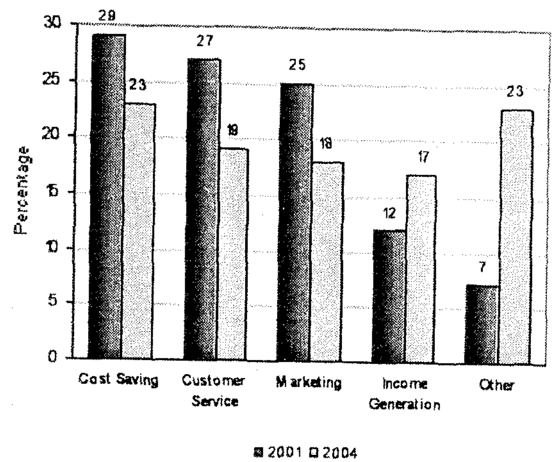


Figure 4 Reasons for Conducting Business Online

It is clear that the categories as proposed by Gow (1997) although still significant are being overtaken by other reasons for businesses to be online. It is a concern, however, that two key success factors; cost saving and customer service have dropped in importance as reasons for being online.

4.3 Impact of the Internet on Income Generation

In 2001, 82% of the respondents reported 0-9% additional revenue generated by being online and the other 18% reported 20-29% additional revenue (Figure 5).

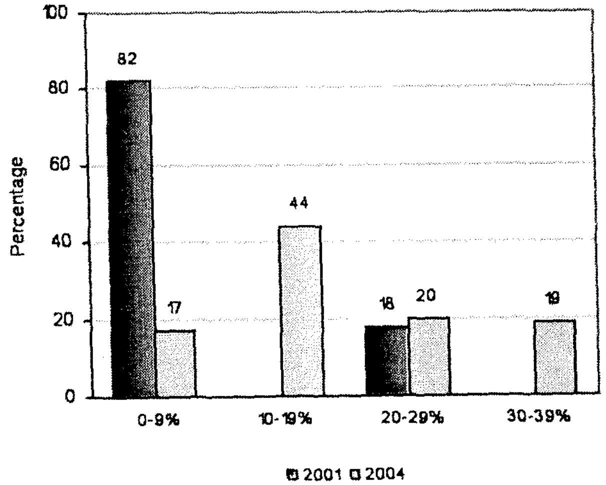


Figure 5 Additional Revenue from Online Sales

In 2001, 82% of the organisations reported generating 0-9% additional revenue from online sales, this category dropped to 17% in 2004. However, overall, additional revenues increased with 44% of organisations reporting 10-19% additional revenue, 20% reporting 20-29% additional revenue and 19% reporting 30-39% additional revenue. The industries that generated additional revenue from online sales were: Education (28%), Retail (20%), Computers & Technology (17%), Financial (16%), Manufacturing (15%), and other accounting for 4%. It is evident that cumulative online sales revenues have been increasing in South Africa even though according to Goldstuck (2004), online retail sales have slowed down in South Africa and globally as well (Mason 2004).

4.4 Impact of the Internet on Cost Saving

It is evident from Figure 4 that managers in this study saw cost saving as an important factor for trading online. Fifty percent of the organisations achieved cost savings of 19% and less. The other 50%

achieved costs savings between 20 and 40 percent. Table 2 lists the factors that contributed to cost savings.

Variable	Min	Max	Mean	Sd.
Advertising	1	5	3.319	1.472
Overheads	1	5	3.131	1.212
Communication	1	5	2.508	1.461
Printing	1	5	2.286	1.016

Table 2 Factors that Contributed to Cost Saving

Advertising costs in traditional media such as television and newspapers are extremely high in South Africa. According to Barker and Gronne (1996), the cost of advertising online is very low. However, its reach is limited. According to Strauss et al (2003), a problem associated with online advertising is that advertisers are unable to pin point who they are reaching demographically, geographically and psychographically.

Overheads such as rent, storage and handling costs, and salaries are much lower as a result of being online. E-mail, frequently asked questions pages on websites and chat rooms reduce the exorbitant costs of telephone and fax based communication. Websites serve as brochure ware that carries more information with text, video, animation and sound for a fraction of the cost of a glossy printed brochure, newspaper or television advert.

4.5 Impact of the Internet on Customer Service

In-store customer service and support takes place during business hours. A customer requiring support after hours has to wait until the next business day. However, due to online support, organisations have seen an improvement in customer service (Figure 6)

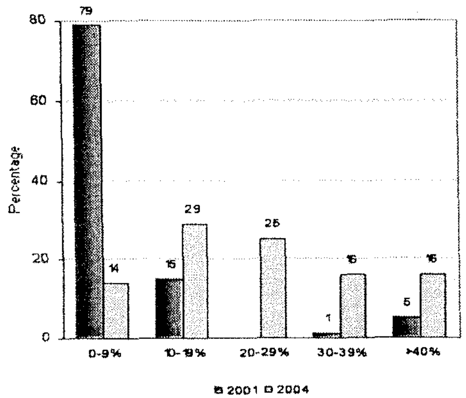


Figure 6 Improvements in Customer Service

In 2001, the majority of managers (75%) reported 0-9% improvements in customer service, 15% reported 10-19% improvements and 5% reported more than 40% improvement in customer service. The 2004 study has shown a major drop in the 0-9% category. However, there is a more even spread with an increase in improved customer service across all categories.

Respondents were asked to rank the factors that contributed to improved customer service. These factors are illustrated in Table 3.

Variable	Min	Max	Mean	Sd.
Cheap communication	1	5	4.336	0.941
Frequently asked questions page	1	5	4.229	0.652
Mass Information	1	5	2.852	1.033
Other (unspecified)	1	5	2.040	0.982
Instant feedback	1	3	1.540	0.562

Table 3 Factors that Contributed to Improved Customer Service

Communication via e-mail is extremely cheap in relation to phone calls and faxes, hence organisations were able to provide customers with detailed information at minimal cost as and when they required it. Frequently asked questions pages (FAQ's) ranked very favourably where the maximum rank was 5 and the minimum rank 3. Customers who required urgent product support could obtain it from a website 24/7/365. Instant feedback with a maximum rank of 3 was not considered to be a major factor. It is important to note that these results are based on the perceptions of managers. According to Schneider (2004), customers rated E-Commerce sites to be average or low in customer service due to slow response times and inadequate integration of call centres with their web sites.

4.6 Management Impressions of E-Commerce

Figure 7 illustrates management satisfaction with the state of E-Commerce in their organisations. These impressions are based on overall performance in terms of: income generation, cost saving, and customer service. It is evident that in 2001, the majority of the managers questioned, (60%), found E-Commerce to be acceptable, with 13% who were extremely dissatisfied.

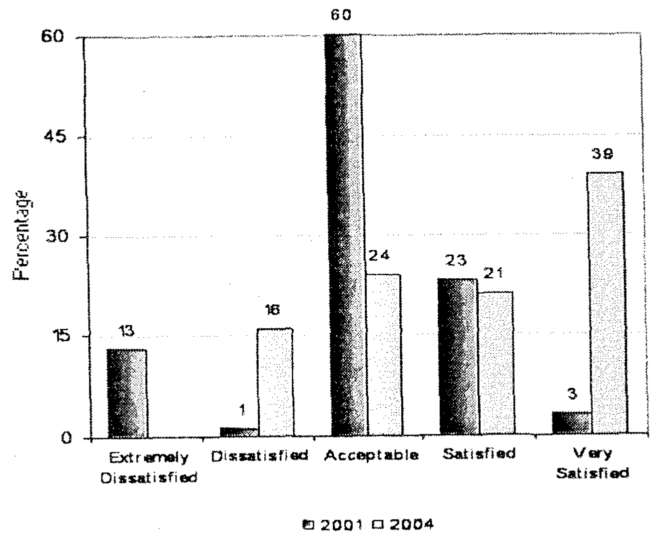


Figure 7 Management Impressions of E-Commerce

The current study has shown that there is a positive skew with 84% of the respondents indicating their impressions as acceptable (24%), satisfied (21%) and extremely satisfied (39%). This could be attributed to the fact that E-Commerce has passed the media hype, through the trough of disillusionment and is moving towards real returns (Strauss et al 2003). Furthermore, users of E-Commerce probably see the usefulness and ease of use of E-Commerce and are therefore more willing to use the technology (Davis 1993).

4.7 What is Needed to Improve B2B E-Commerce?

The majority of the respondents (84%) felt that E-Commerce is South Africa was strengthening and that a turnaround could be expected soon. They felt strongly that in order to improve B2B in South Africa, the following requirements needed to be met: effective websites (30%), improved infrastructure (27%), an e-strategy (20%), im-

proved security (13%), and an e-culture (10%), as illustrated in Figure 8.

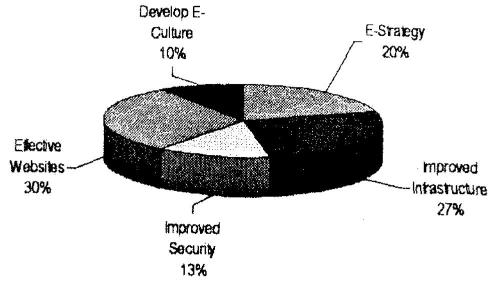


Figure 8: Requirements for Successful B2B

It is evident from the list, that some of the requirements as recommended by the respondents can be implemented by businesses such as website design, improved security and e-strategy. However, improved infrastructure is the sole responsibility of the state. It is important to note that Government should also have an E-Strategy in place that would incorporate plans for developing an E-Culture and thereby enabling e-commerce.

5. Recommendations

It is evident from the results that there have been positive changes in E-Commerce in South Africa from 2001 to 2004. These changes have been evident in all aspects of this study namely income generation, cost saving, and customer service. These positive changes have resulted in managers having a positive impression of E-Commerce. However, Goldstuck (2004) has highlighted the fact that there has been a slowdown in online retail sales. What should organisations be doing to reverse the trend?

5.1 Customer Service

Microsoft suggests that organisations should use the web creatively to deliver customer service (Give Them Online ... 2005). Satisfied customers become loyal customers and would think twice before shopping elsewhere. Some of the suggestions included: FAQ's, information and updates, policies, order tracking and order histories, and contact information. FAQ's are in use by South African companies. New product information and alternate uses could be provided on the company website. The company returns and exchange policy should be available online to prevent any misunderstanding after the purchase. Order tracking allows a customer to determine for themselves when to expect delivery. Order histories such as favourite buys or a previous shopping list speeds up shopping which saves customers time and money. Providing physical contact details is reassuring to the customer that they have someone to talk to or a place to visit if there are problems related to online purchases.

5.2 Income Generation

In order to increase online income, businesses need to attract traffic to their websites, which requires extensive offline advertising (Singh 2002). They also need to develop effective websites and provide adequate online security.

5.2.1 Design Effective Websites

Rayport & Jaworski (2002) suggest that web designers need to use the 7C's model namely; content, context, connection, communication, community, customisation and commerce when designing effective web sites. The 7C's ensures that the customer sees what he/she needs to see and that they can communicate with the site before and after the sale. Web designers need to balance aesthetics with functionality that is, a website need not be populated with fancy backgrounds, pictures and animations; however, the site must work immaterial of whether a person connects via a dial-up link or a leased digital line.

5.2.2 Improve Online Security

Secure transactions will provide customers with the additional reassurance that they require in order to purchase online. Some of the larger sites such as Standard Bank and Kulula.com offer customers 128 bit encryption which claims Scholtz (2005) is safer than using one's credit card at a restaurant where a waiter has access to all the details of a customer that are required to conduct a transaction online. Businesses need to develop a privacy policy which is available for the customer to see on the site. They should also register their site with authentication authorities such as Verisign. Digital certificates are further proof to customers that their transactions are safe and secure.

5.3 Cost Saving

By its very nature, the Internet reduces overhead costs. This should be integrated into an organisations competitive strategy. Standalone e-strategies do not make sense where businesses run an offline and online service. Porter (2001), Botha et al. (2004) and Boddy et al. (2004) advise strongly that businesses integrate their E-Strategy with their IT strategy which must also be aligned to the broader organisational strategy. For example, an organisation pursuing a lowest cost producer strategy would use IT to enable low cost manufacturing and the Internet could be used as a means to market and distribute products at a low cost. In this example, the business strategy guides the IT strategy which guides the E-Commerce strategy. In organisations that only have a web presence, the E-Strategy is the business strategy. However, businesses need to concentrate on the core values and basic rules of business that is to concentrate on the customer and create customer value which will stimulate sales.

5.4 Other Recommendations

Some of the suggestions made by the respondents included: expanding infrastructure to a wider audience, improving education and developing an e-culture in the country.

5.4.1 Expand Infrastructure

Developing ICT infrastructure is one of the Government priorities in South Africa. However, infrastructure alone is not enough to generate online sales. People need to be educated and empowered to use the Internet for purchasing. There are a number of initiatives that the Government in conjunction with private partners are driving to make telephony and electricity available to rural areas of South Africa (Singh 2004b). Once the infrastructure is available, people will have access to the Internet, and its related applications such as e-commerce, e-mail among others.

5.4.2 Develop an E-Culture

According to Singh, (2004b), universities and other educational institutions need to inculcate in learners that they are a part of the E-Society and as such they should lead the masses towards embracing the change.

Online organisations have to reinforce in customers the value of shopping online. They must be constantly reminded that online shopping is easy, it is cheaper and it offers 24 hour convenience.

6. Conclusion

This study has shown that E-Commerce in South Africa is growing. It is only a privileged few who have the access and purchasing power to shop online currently. However, major strides need to be taken to expand the market. This study has been limited in that it has not examined the value of business-to-business transactions, and it has also not been able to quantify the value of foreign sales against domestic sales. These issues however, lend themselves to further study.

South Africa is a rapidly developing country, and with it ICT is developing equally rapidly. Making E-Commerce work requires an investment from both the public and private sectors. Public, private partnerships will aid in driving E-Commerce forward in South Africa. It is evident from the study that organisations need to improve their

online customer service, they have to design more effective web sites and they need to improve their online security. However, there is an obligation on the Government to make infrastructure available to enable e-commerce and to ensure that e-commerce awareness and use is promoted.

E-Commerce and an E-Culture have the potential to change and grow as younger generations embrace the technologies and influence others to embrace it as well.

Acknowledgement

I hereby wish to thank Mr Yuri Baijnath for allowing me to use the data from his 2004 study to compile this paper.

References

Barker, C, & P Gronne 1996. Advertising on the Web. Unpublished Doctoral Thesis. Copenhagen Business School.

Boddy, D, A Boonstra, G Kennedy 2004. England: Prentice Hall.

Botha, J., Bothma, C., Geldenhuys, P. 2004. *Managing E-Commerce.* Cape Town: Juta Academic.

Davis, FD 1993. User Acceptance of Information Technology: System Characteristics, User Perceptions and Behavioral Impacts. *International Journal of Man-Machine Studies* 38: 475 – 487 www.microsoft.com/malasia/smallbusiness/issues/creating/improve/smile.mspx. Accessed February 2005.

Goldstuck, A. 2002. The Goldstuck Report: Online Retail in South Africa, 2002 www.theworx.biz.retail02.htm. Accessed February 2005.

Goldstuck, A. 2004. Online Retail in South Africa 2004. The Need for Speed in Online Retail www.theworx.biz/ retail04.htm. Accessed February 2005.
Gow, K. 1997. Risk v Opportunity. *The Premier 100 Supplement to Computerworld* February 1997.
Internet Growth Statistics 2005 Today's Road to E-Commerce and Global Trade. 2005 www.internetworld stats.com/emarketing.htm. Accessed February 2005.
Madonko, I 2000. Allure of Internet Reaches all Corners of SA. *The Mercury-Technology,* 18 July p. 4
Mason, H. 2001. Price Driving Online Retail Growth. http://www.theretailbuletin.com/index.php?page=5&cat=news&id=4464. Accessed January 2005.
Porter, ME 2001. Strategy and the Internet. *Harvard Business Review* 79(2): 63-78.
Rayport, JF & Jaworski, BJ 2002. *Introduction to E-Commerce*. Boston: McGraw Hill.
Schneider, GP 2004. Electronic Commerce. *The Second Wave* (edition 5). United States: Thomson Course Technology.
Singh, AM 2001. The Effectiveness of Internet Marketing: A Management and Consumer Perspective. Unpublished Doctoral Thesis. University of Durban Westville.
Singh, AM 2004a. Managing Employee Internet Abuse. *South African Journal of Information Management* 6(3) September. http://www.sajim.co.za/default.asp?to=peer6vol6nr3. Accessed July 2005.
Singh, AM 2004b. The Role of South African Universities in Bridging the Digital Divide. *South African Journal of Information Management* 6(2) June. http://www.sajim.co.za/default.asp?to=peer4vol6nr2 . Accessed July 2005.
Statistics for Online Purchases. 2003. www.epaynews.com/statistics/purchases.html . Accessed January 2005.

Strauss, J, A. El Ansary, R. Frost 2003. *E-Marketing* (edition 3), New Jersey: Pearson Education International.
Turban, E, D King, JK Lee, & D Viehland 2004. *Electronic Commerce 2004: A Managerial Perspective* (edition 3), England: Prentice Hall.

Author's Contact Details

Anesh Maniraj Singh (singham@ukzn.ac.za)
School of Information Systems & Technology
University of Kwazulu Natal