ABSTRACT
As technology proliferates, the relationships between people, information and technology become increasingly significant. However, software development methodologies too often ignore the human factors by concentrating on advanced technologies and communications protocol. Consequently, numerous courses have been designed to reinforce the user's ability to interact effectively with technological applications, especially through Computer-Based Training (CBT), Embedded CBT (ECBT) and Intelligent Computer-Assisted Learning (ICAL). A model is proposed integrating the advanced components of Intelligent CBT (interaction, individualisation, availability) with systems design methodologies and human factors issues to provide an error-free operational environment. Research implications in communication and the information sciences are discussed.

INTRODUCTION
Almost daily, the capability of computer technology to perform more quickly, access more information and outclass existing software applications is reported. However, users of the technology have changed very little, except perhaps in expressing increased frustration at the inability of applications to meet individual skill levels and the ever-present expectation from management for information to support decisions. Based on this condition, this paper examines the options for both mainframe-based Management Information Systems (MIS) and micro-computer applications to respond to the increasing emphasis on human factors considerations. Specifically, emphasis is placed on the human-machine interface of installed systems rather than the human factors inherent in the systems development cycle such as organisational and data analysis, documentation and programming standards. The analysis from human rather than technological perspectives provides a rationale and context for an integrated systems design model, based on the assumption that "ease-of-use" is of paramount importance for effective and efficient use of information systems and application packages.

Second, the importance of training for users is considered in terms of the ever-increasing growth of personal computers and the almost mandatory requirement for employees to use software applications in their day-to-day work. The use of Computer-Based Training (CBT) as a solution to this environment is evaluated with reference to developments in educational technology research into training systems based on Embedded CBT and Intelligent Computer-Assisted Learning. For this analysis, CBT is assumed to apply to computers as the instrument of training; that is, technology is used (in some way) as a surrogate instructor. This is to differentiate the use of CBT to describe Computer Education (where computers are the object of instruction) and training which uses computers as equipment for teaching specific skills such as CAD/CAM or Desktop Publishing.

Based on the converging paths of instructional technology, systems design methodologies and human factors, the third section proposes a model which integrates these elements into an adaptive computer-based system which caters for the varying expertise and learning styles of individual users. The model also describes a system oriented towards people rather than machines, adapting to individual user requirements and providing a satisfying and productive operational environment. Developmental considerations for the model including cost-effectiveness, user demand and organisational commitment provide a starting point for further investigation.

MIS AND HUMAN FACTORS
Computer-based information systems increasingly involve online interaction between the human user and the machine: a critical element of the design of these systems is the user interface. The interface consists of screens, keyboards, devices, languages and other means by which the human user and the computer system exchange inputs and outputs.
The User Interface

This acknowledgment of the criticality of human factors tends to emphasise means (devices and languages) rather than ends (system functionality from the user perspective). Of course, in the developmental environment, the former aspects are vital for the production of efficient processing systems. Of equal importance however, is the ability of the user to operate computer-based systems without having to resort to external resources for help.

This brings in the notion of a user-interface flexible enough to meet novice and expert needs, consistent in its presentation and adaptable to individual skill levels [1,2,8]. In considering this link or interface between software applications and people, it is also evident that a significant proportion of the actual development methodology concentrates on system attributes such as programming and documentation standards, hardware and communication, personnel planning and project management. While recognising this as essential for systems implementation, the integration of human factors to the development cycle is not yet established [3,4,9]. Additional evidence of the importance of human factors in systems design and development is provided by an analysis of the cost/benefits of introducing human factors into design methodologies. While concentrating on the increased effort involved in design stages such as prototyping, they also suggest the inclusion of human factors will reduce user training [10]. However, the discussion does not include training as an integral component of the systems development cycle.

Developments in the integration of human factors to software applications include adaptive models which cater for individual skill levels and cognitive psychology issues relating to the ways people process information [1,9]. This interest in adaptive systems and cognitive science is not only applicable to human factors, but also in the area of instruction and intelligent job aids [11,12]. The result of this is a convergence of factors indicating a movement towards an holistic approach to systems development, with the user as the central factor.

Despite a broader, people-oriented approach to system development, there remain a number of cases where technology fails to provide adequate communication with the user. The design criteria for these systems (whether operating or application) appearing to stop at the screen display, not 30–46cm further out to include the users ability to interact with the system. In other words, there are human factors concerning user knowledge and skills which are not included in the design methodologies.

For example, many computer applications, particularly micro-computer systems, assume that the user understands how to achieve their particular goal, despite being presented with blank screens, a prompt such as "C:" or "OK" and a flashing cursor. To the initiated, of course, the required response is virtually automatic, but for the new user there may be many unanswered questions. The implications concern the quality of communication taking place in the human-machine interface. As the demands of business require the user to access technology options available for successful operation, it is essential that an effective interface in terms of user-skill and screen organisation be maintained [1,8].

However, very few computer systems actually communicate with the user apart from responding to specific input, which even then may be cryptic: "Abort(A), Retry(R), Ignore(I)" or "Bad Command or Filename". Such responses assume considerable knowledge on the part of the user, including functions of the computer operating system and skills in providing meaningful commands to be interpreted by the software. The level of user-skill is therefore paramount in the effective operation of software applications.

The Need for Skilled Personnel

User knowledge and skills are the concern of training, and therefore provide an opportunity to integrate training into the development of software systems. For example, consider the person who has no computer experience other than recognising a particular need for a particular software application. If that person were to purchase an IBM-PC with an MS-DOS operating system and open the User's Guide, the first words are "Welcome to the MS-DOS operating system, version 3.2" [13]. The guide assumes the user knows about operating systems, but from an instructional perspective there is no guarantee the user has this knowledge.

On the other hand, if the new user was able to be guided through the application or manuals using the technology, the importance of the user could be reinforced through specific in-built training functions. Significantly, these assumptions extend across the range of software packages and highlight the opportunity to develop skills in user-interface or communication.

However, the technology itself is rarely used to support this training, other than providing the means to operate software in classroom settings [7]. More recently, Computer-Based Training (CBT) has been implemented in an attempt to facilitate the availability of training, with considerable success. In fact, the support of application software with CBT would appear to be an attractive alternative for both vendor and user [14]. With this in mind, the following section considers the developments and opportunities for CBT as a solution for the integration of human factors to application software.

COMPUTER-BASED TRAINING

An Introduction

In parallel to the growth of information technology, Computer-Based Training (CBT) has developed to be recognised as an effective solution for skills training. However, despite the accepted
Instructional benefits such as individual learning and self-pacing, CBT is only now beginning to make a significant impact on training practices [15]. One reason for this delay is the development costs associated with instructional software and the lack of identified training needs to justify the development effort. In fact, instructional technology often appears to be used for its novelty rather than instructional worth, emphasised by the suggestion that only tomorrow's technology will provide effective and efficient CBT [6].

Nevertheless, in the Australian business environment, there is a growing demand for trained employees, based on the assumption that skilled operation will result in a minimisation of errors and increased productivity. A basic form of CBT which has been adopted in software applications to assist the novice user are the on-line or 'hot-key' HELP facilities [16]. However, these only provide information to the user rather than training in skill deficiency; additionally, the information displayed often assumes a certain level of user knowledge to allow interpretation of the HELP message.

For example, Figure 1 illustrates a display from the on-line HELP facility for MicroPro International Wordstar(R). In this instance, the HELP simply details the function of particular commands, requiring prior-experience and/or a manual for successful comprehension.

An application for this extended-HELP facility is illustrated in Figure 2, using Software Publishing's First Publisher.

There are of course many areas where CBT has been used to assist with developing skills in using application software [14]. However, CBT has typically been used in parallel with the software application for which it has been applied. One of the more recent developments, and incorporating the "extended-HELP" identified above, is the concept of Embedded Computer-Based Training (ECBT) which does not distinguish between the software application and the training system, but rather integrates the two in the system development cycle.

Embedded Computer-Based Training

Embedded CBT (ECBT) has the benefit of being delivered to the user/trainee on a "live" rather than an off-line training system:

ECBT is quite definitely best employed to train people to make effective use of the technology they use in their daily work. It does this by utilising the same technology used in the workplace as the main training tool [15:19].

The major elements of ECBT include a training system, an operational system, an interactive help system, a training database and an activity monitoring facility [17]. With these components the user has the option to access training which is supervised by the software application to control the communication between the user/trainee, the training database and the training system.

The interactive help system is always available for on-screen assistance to the user. The training database is maintained to reflect the actual operational system and the monitoring facility provides feedback information relating to the effectiveness and use of the training system. The latter component is essential for effective

M (merge print a file) prints multiple form letters from a master document you create. Using dot commands, you can merge into your form letter variable information from data files on your disk.

See the Reference Guide in your manual for more information on merge printing and dot commands.

Press Esc to continue.

Figure 1: WORDSTAR(R) Help Display

A future development for this concept will be the HELP "window" which describes more accurately the item requested, as well as providing the opportunity for further Computer-Based Training. The only drawback to this approach will be the development effort required for implementation (in addition to those for human factors and possible hardware and software constraints, such as memory, storage or response times [10]. As new operating systems emerge (e.g. O5/2) and the user base expands, there will not only be the flexibility but also the demand for systems which adapt to the user population.
training as it provides data for the modification of the training to meet changes in learner needs or system characteristics.

ECBT requires a novel approach in that trainers, systems analysts and users must all form part of the design team; these individuals define the project group whose task is to analyse the software applications well as the training requirements. The significant advance in training technology is that training based on ECBT is designed at the same time as the application software, and the training itself forms part of the operational system. The importance of this approach is that rather than training users about the technology, ECBT involves learner-centred training using an integrated training system. In essence, ECBT modifies the application of human factors to systems design to include training and learner factors [10].

While ECBT provides immediate access to training based on the active system, there is no guarantee that it will cater for the individual skill levels of users or their particular learning requirements. This leads to a consideration of advancements in software technology which are bringing training systems, information systems and human factors closer and closer together by the integration of artificial intelligence and expert systems to produce intelligent computer-assisted learning resources [6,18].

Intelligent Computer-Assisted Learning (ICAL)

The development of Intelligent Computer-Assisted Learning (ICAL) provided a new era for computer-based learning systems, with the potential for instruction based on an individual's particular needs. The significant difference between ICAL and traditional CBT is the move from teacher-based systems towards learner-based systems, such that the instructional resource can draw on a knowledge-base (content) by a tutorial model or inference engine which is activated by trainee responses. The content presented to students is dependent upon their particular circumstances rather than those defined by an instructor.

One of the significant research areas in artificial intelligence is the natural language interface whereby users can communicate with the system using everyday language. The benefits of this in terms of human factors and software applications is the option for users to make inquiries about a particular operation using their own terms and whatever their particular skill. Currently however, the availability of such systems is restricted by the capability of the technology to communicate with the user on an "intelligent" basis:

We did have to train natural language subjects in a variety of material which was not anticipated. First, we had to describe the philosophy and coverage of the language. This was done by emphasising restrictions. Then, we had to show subjects substitute procedures for getting round language "gaps". Third, we had to describe the meaning of messages and other feedback from the system. Much of this material is likely to be difficult to retain. There are no easy rules to guide users as there are with some artificial languages. It had not been obvious to us at the beginning of the project what topics would be important for natural language users to learn, or how to go about teaching them [19:185].
The desirability of natural language as a communication medium must be considered in terms of software engineering, which is currently far from providing this facility. However, in discussing options for "intelligent" training, the possibilities shown in Figure 2 can be extended to those in Figure 3.

In this example, Software Publishing's First Publisher would be modified to include a record of tasks which cause problems for individual users. Once a user attempts a pre-defined number of incorrect or unavailable options, the system will respond with guidance and a request for clarification. The user may then pose a question (shown in bold type) which the application, through natural language processing, will interpret and make a response. While this example is purely hypothetical, it illustrates the direction of user training and the convergence of human factors and systems development issues.

Like traditional and embedded CBT applications, Intelligent Computer-Assisted Learning (ICAL) has often been used in isolation from the system for which it is providing the training. However, the benefit of ICAL is its ability to adjust and modify training to cater for individual levels of knowledge and understanding. In doing so, it draws upon a particular knowledge base, a model of the current student and a tutorial model which can determine the next step in the instructional process. More recently, this has been enhanced by evaluating the human factors issues involved in learning such as screen presentation and the cognitive abilities of the learner [11,20].

Having described how human factors and training technology are significant components of software applications, the final step is to integrate the systems development methodologies with those for human factors and intelligent CAL to produce an integrated system catering for the individual skills and requirements of the user [21,22].

A MODEL FOR PEOPLE-ORIENTED SYSTEMS

Converging Systems

The basic thesis of this paper is that software applications have only recently included human factors in the development process, and that requirements for training have been determined as a separate rather than integral component of the application. The logical step is therefore towards a systems development methodology which includes not only the specifications for optimum processing but also the capability to address individual user requirements based on skill level or learning needs.

A starting point for this approach is an "adaptive" model (Figure 4) for user understanding which includes both human factors and training [1].

This adaptive model is designed to pose questions to ascertain user knowledge levels and generate alternate processing paths to optimise performance and implies a system adapting to user needs rather than a user adapting the system to satisfy those needs [1,23]. Although this approach emphasises the user-interface, it also includes the elementary components for an intelligent training system. However, the important considerations for adaptive systems are whether the additional design effort and the apparent speed of novice learning justify the development of a user-oriented system [10].

The conclusion that "design of business computer systems must gravitate towards the user", supports
the role of technology as a tool, although the development effort required for such "people-oriented" systems cannot be ignored [1:29].

Moreover, it should be noted that the adaptive model proposed by does not include training as a component of the software application [1]. Instead, training is portrayed in its traditional environment - separate from the technology for which the training is designed. Given the developments in embedded CBT and intelligent CAL, the stage is set for software development based on the converging technologies of systems development, human factors and training.

An Integrated Model

A modification to the adaptive model is shown in Figure 5, with an iterative development process generating a software application which caters for data processing, user requirements and training to provide operational efficiency for the user. The implications of this model are that user understanding will be the major outcome, producing minimal errors in operation, increasing opportunity costs by minimising off-the-job training and achieving organisational objectives. This understanding will be achieved only if the complete system is continually evaluated and maintained.

![Figure 4: A Model for User Understanding](image)

![Figure 5: Integrated Developmental Model](image)
An example of the user-interface provided by this model is shown in Figure 3, which illustrates the use of a context-sensitive ICAL which, when the user has made successive (abortive) attempts at a particular function, generates on-screen remedial information, provides the opportunity for natural language conversation, and produces feedback when required. However, it is the actual attributes of the model which demonstrate the potential for this form of software application.

The first component of the model is based on an iterative design and development process. The systems specifications relate to the computer application and include accepted systems analysis practices with additional input of training and human factors elements. Analysis of human factors includes the type of system being developed (e.g. mainframe/micro-computer; MIS/application) in terms of the user-tasks required and the relevant training required for those tasks, dependent on the anticipated skill level of novice and expert users. Similarly, the training and documentation must define the user-tasks in terms of learning objectives to allow the sequence of instruction to be delivered according to a stable instructional strategy.

By applying systems analysis, human factors and instructional design methodologies, the resultant design specifications will include not only a model for the software application, but also a model of the user(s) who will be working with the application which addresses both the person-machine interface environment and the relevant training requirements.

On acceptance of the first stage design, the development of the computer-based system may commence. With the inputs from the initial stage, the software application becomes a model of the user and the computer. While current technology will allow this development to centre on the ability of the system to adapt to individual requirements (and the necessary training factors to ensure that operational deficiencies are rectified) the opportunity exists to consider the integration of expert systems and artificial intelligence [1,5,12,21,32].

However, while the advantages of an integrated "people-oriented" system will be seen through an application which provides an interface designed for user operation and education, considerable challenges remain:

**Serious obstacles to overcome include the difficulties involved in extending the leading edge of research in artificial intelligence, and uncertainty about how much resources and time will be needed to develop products. Limits on the sophistication of user interfaces, on the scope of subject domains, and on current understanding are seriously constraining the effectiveness of current intelligent tutors, coaches, and empowering environments [21,181].**

While these factors must be taken into account, the positive aspects of this integrated approach to systems and application software development will be operational efficiency achieved through users being prompted for incorrect actions, presented with option selections dependent upon their particular skill level and provided with remedial or orientation training when required.

This "people-oriented" model illustrates the advantages available to developers of computer systems through interacting with instructional and human factors engineers. While the requirements for storing user-data and monitoring user-performance may currently be reserved for mainframe environments, the continued developments in storage and retrieval technology will ensure the availability of this mode of processing. As the emphasis on human factors and training grows, so will organisations insist more and more on applications which support "error-free" processing.

**Error-Free Processing**

The integration of software, training and human factors will generate systems in an environment which is totally self-supportive and which will not, in any way, confront the user with "errors". When the user interacts with the system, the application will not differentiate between right and wrong, but have the facility to query the user as to their desired action. Some application software has already implemented this form of processing; in 1984 Cincem's TIS database product assumed the user's intent if it could match a misspelt field-name or provide a list of available field names for the user to make a selection.

The importance of an "error-free" processing environment is that operations are based on the user rather than the software, and any mistakes on the user's part are not met with complex error-messages. Instead, the user's (erroneous) inputs may be automatically adjusted, queried for clarification or highlighted by initiating an interactive training session. To achieve this, systems development will have to include constant evaluation and ensure that such quantitative data is used to modify the application and maintain the system as a dynamic entity.

**RESEARCH IMPLICATIONS**

The development of integrated models will provide significant areas for research in communication and information science. The increasing impact of technology in everyday activities will require people to have generic person-machine communication skills. For example, modeling the particular "map" which a person has of a software application and correlating this with the actual structure of the software, which forms the basis of the User Model/Computer Model illustrated in Figure 5. Developing an understanding of the dynamics and relationships in user-computer communication will lead to more effective applications, through the use of technology as an efficient tool.

For information science, the challenge is to assess the impact technology will have on information use, dissemination and retrieval activities. In
comparison to traditional oral transmission of information (facts, tradition, data), technology is quickly becoming the repository for information. Research and development in knowledge-based information retrieval systems, linked with communication, will assess the viability of such systems to interrogate the user as to their requirements and to provide information for use based on probability factors and rules [18]. An evaluation of changes in cognitive structure will also determine the extent to which technology is taking the place of human memory and opening alternate avenues for our intuition.

With these research avenues, curriculum for graduates in higher education will require extensive analysis of technology not as a data processor and information provider, but as a surrogate for human memory. More importantly, the courses of study will become more dependent on human-oriented disciplines such as psychology and sociology. An integration of the traditional information science units with computer technology, cognitive science, artificial intelligence, education and psychology will provide a unique environment for the acquisition of skills suitable for a technology-based future.

CONCLUSION

It is apparent that information systems and instructional systems are converging on a track which will generate "people oriented" systems catering for varying user skills, the range of interactions required between user and system and the knowledge level of the user.

While the problems in developing such systems will be based on current technological capabilities and the ever-increasing literature on human factors, cognitive psychology and the user-interface emphasises the importance of this development. Despite the additional design and development effort required for the integration of human factors and training, the investment in "people-oriented" systems will be repaid by having skilled staff able to use the system effectively, minimising training requirements and maximising productivity.

Another advantage of this approach to systems development is that the principles of integrated, user-centred development can be transferred to other areas where user or worker skills are vital for efficiency. For example, in developing computer-based training for content areas diverse as insurance sales, product knowledge or management skills, organisations can ensure that training is not only interactive but also based on the learner as the focus of the resource.

In fact, as technology becomes more and more a part of everyday work, employers will be demanding environments in which their employees can work effectively. The concept of 'people oriented' systems and the integration of both human factors and training is an important step in this direction.

REFERENCES