# COMDESIGN PRINCIPLES FOR ‘NEXT WAVE’ EDUCATIONAL TOOLS: THE DEVELOPMENT OF THE WEBMCQ ASSESSMENT SYSTEM

[1]Scott Gazzard and [2]James R. Dalziel

[1]eventHorizon Software, Australia.
email: scottg@eventhorizon.zip.com.au
http://www.eventhorizon.zip.com.au/

[2]Department of Psychology, University of Sydney, Australia.
email: jamesd@psych.usyd.edu.au
http://www.psych.usyd.edu.au/staff/jamesd/

## *ABSTRACT*

*A recent trend towards increasing use of computers and the World Wide Web (“the Web”) in higher education is currently merging with an earlier trend towards using multiple choice questions in assessment. The use of computers in conjunction with such assessments greatly enhances their potential effectiveness. The Web further enhances the ability to distribute assessment material widely, allows inclusion of multi-media content, and enables a level of interactivity and data collection not possible with paper-and-pencil assessments. With these possibilities in mind, the WebMCQ system was conceived as a means of presenting multiple choice questions to students, and potentially providing immediate feedback on their responses so that WebMCQ could be used as a learning tool in the first instance, and also as a summative assessment tool.*

*The development of software such as WebMCQ is part of a much broader shift made possible by extensively-networked computers and the emergence of the internet. It will be argued in this paper that new models of distribution and utilisation of software are now poised to bring about a major shift in the way software is perceived. Rather than a traditional view of software as a product that is installed on individual computers, the features of WebMCQ led developers to adopt a new distribution model, whereby software resides on a central server and is accessed by subscribers to a service. It is this shift from perceiving software as a product to perceiving software as a service that has potential to bring about enormous changes to software design and utilisation and which, it is argued, will result in the arrival of ‘next wave’ software. Among a host of benefits to consumers will be vastly superior ease of use and convenience, leading to adoption of new software on a massive scale.*

*Next wave software will force software developers to adopt new ways of designing, distributing and supporting software. To illustrate some of the new design principles that future developers will need to consider, the paper includes a discussion of five core design principles used in the development of WebMCQ: ease of use, flexibility, consistency, accessibility, and integrity. Each of these issues are discussed in detail, and several predictions are made about how each may impact upon future software developments. It is concluded that the arrival of next wave software will herald a revolutionary shift in the way educational professionals perceive and use Information Technology.*

## KEY WORDS

Software, computer, interface design, WebMCQ, assessment, internet.

## 1. INTRODUCTION

Historically, the development of information technology (IT) has comprised elements of both evolution and revolution. While acknowledging the fact that evolution in this field is often over-hyped to the status of revolution by developers and marketers of new technologies, this paper will argue that IT is on the cusp of a genuine revolution in the way software and computers are perceived and used. The revolution will not take place in hardware technology or software design per se, but rather in what software is perceived to be. It is a revolution (like all good revolutions, one might say,) that originates from and belongs to the people – consumers. While change will not be limited to educational settings, the revolution certainly promises special benefits to educators, and thus will have flow-on benefits to generations of future students.

This paper will outline the nature of the software revolution, which will be represented here by the term ‘next wave software’. Major features of the forthcoming revolution will be explained, followed by attempts to predict some of its consequences. One of the major consequences, it will be argued, is a forced change among developers and distributors of software in the way software is designed. To illustrate some of the principles important in the design of next wave software, the development of WebMCQ will be presented along with a discussion of the design principles underlying that software/service and exciting new models of distribution and utilisation.

## 2. WHAT IS NEXT WAVE SOFTWARE?

The central claim of this paper is that the ‘next wave’ will be nothing short of a change in our collective perception of what software is. Current models of software distribution and utilisation are closely related to models that had previously been established for books. However, the rise of the internet and networked computing is making possible new forms of distribution and new models of utilisation. These new models have great advantages in terms of flexibility, convenience, cost, global resource-sharing and ease of use. These advantages (in conjunction with a number of other factors currently gaining momentum) will result in much greater adoption of new software in future. Software will cease to be considered as a ‘product’, which like a book, is purchased in physical form at a store and then transported to its place of use. Instead, software will begin to be viewed as a service to which one subscribes, in much the same way as television content is broadcast and accessed. However, unlike television, next wave software services will be controlled entirely by the consumer (an irony of the television metaphor is that in future our perceptions of television itself will shift towards ‘product’, and will be delivered as software in an identical manner to applications).

At the same time, software will continue to become easier to use. Improvements in artificially intelligent interfaces will unobtrusively offer advice and help in context, as it is needed. And help systems will become much more human, advocating learning-by-doing, with stimulating use of multimedia. Software users will simply ask questions of their software using natural language, and relevant answers will be received immediately. Software design will also continue towards greater consistency across software packages, as the ‘visual metaphorical language’ of software interfaces matures and consolidates. Greater flexibility will be built into software, catering for different styles of working and allowing more than one way of achieving a desired goal. The software will recognise users as individuals and take it upon itself to adapt itself to personal styles, rather than forcing all users to learn how the software wants things done.

The essence of ‘next wave software’ is captured in the phrase software that comes to the consumer. Current consumers are required to ‘go to’ the software, both in the physical sense when acquiring the product, and in the (often more time-consuming and intellectually draining) psychological sense when labouring to learn how to use new software. When current barriers to adoption of new software are removed, its adoption will increase explosively. Simultaneously, falling hardware prices and advances in technology will continue to encourage educational institutions to broaden IT investment. But here’s the twist: the perceptual shift brought about by next wave software means that educators charged with using new software will be glad to adopt it. Rather than seeing new software primarily as a difficult challenge to master, it will be seen for what it should be: a new opportunity with minimal interference to productivity.

## 3. AN EXAMPLE OF NEXT WAVE SOFTWARE: “WEBMCQ”

WebMCQ exists today as a professional service allowing educators and researchers around the world to present multiple choice quizzes and questionnaires via the Web. Its beginnings lay in a simple idea to produce a one-off program for on-line delivery of a practice multiple choice quiz to students in the 1997 first-year psychology course at The University of Sydney. It was envisaged that students would access the quiz from tutorial rooms at the university, and/or from their homes. As outlined in Figure 1, WebMCQ consisted of a bank of around 80 questions organised into 8 sections, with each section corresponding to a topic covered in weekly tutorials across one semester. After attempting each question students received a basic level of feedback, including whether they were correct or incorrect, giving brief information why, and displaying a button marked ‘More Info’. Students who wished further explanation of the answer to the question they had just attempted were able to press the More Info button to receive more detailed feedback. The nature of more detailed feedback varied from question to question, but some examples included illustrations, extracts from handbooks, references to textbooks, and links to other Web resources. At the end of the quiz students received a summary of their performance.

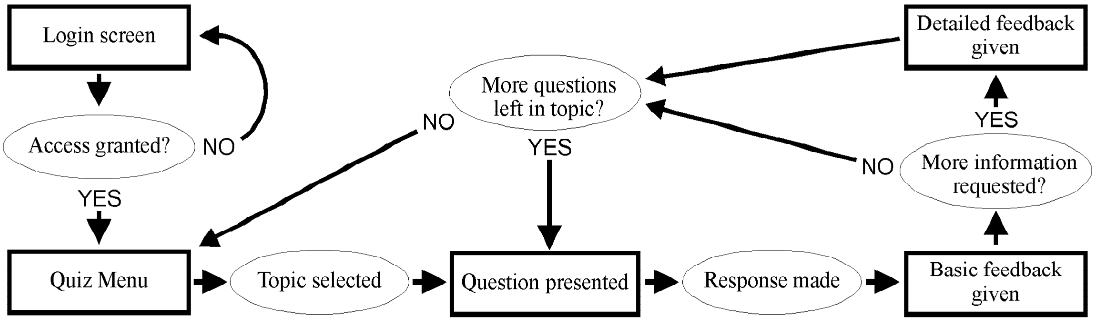

Figure 1: Schematic representation of the structure of a WebMC quiz.

Student evaluations of the quiz as an educational resource were collected via an anonymous HTML form linked to the Web quiz. Overall response to the program was so positive (see Figure 2), and usage rates so much higher than anticipated, that it was realised the product had far greater potential than simply the first-year course in the Department of Psychology, and further development was undertaken with the cooperation of the University of Sydney.

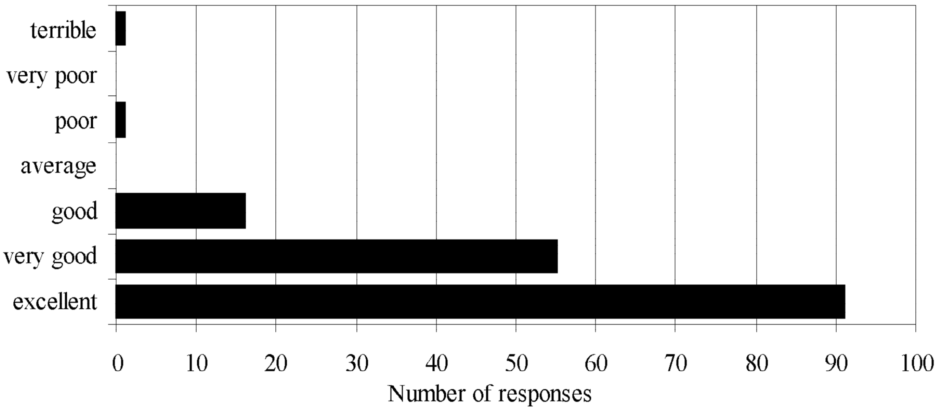

Figure 2: Psychology student responses when asked "How would you rate the quality of this quiz as a learning resource?" about a WebMC revision quiz in 1997 at The University of Sydney (N=164).

Initially, conventional notions of the software as a product were assumed, but during the early stages of the development process it became increasingly evident that an entirely new conceptualisation of WebMCQ was required. This was reinforced repeatedly as the software was refined through multiple stages of development involving a number of test – modify – retest stages. And, originating as it did from academics rather than computer programmers, with the benefit of detailed feedback from hundreds of users students, the resulting software was developed with a constant focus on the needs of educators and the wishes of students.

## 4. WEBMCQ DESIGN PRINCIPLES

At the outset, five core design principles were identified as being essential to the success of WebMCQ and these principles were used to guide responses to user-feedback and to experience gained by testing the software in different contexts. Those principles are ease of use, flexibility, consistency, accessibility, and integrity (a sixth principle of security is less relevant to this discussion). These principles were identified with consciousness of the potential for developing a new model of distribution and utilisation, and in turn helped to formalise the operational parameters required. However, the distribution and utilisation model itself is not a design principle as such, and so will be discussed in a subsequent section, after an examination of each design principles in detail.

### 4.1 EASE OF USE

All software touts ‘ease of use’ as a feature, but next wave software will much more effectively focus on bringing the software to the consumer by designing software according to how people will most readily use it. In the case of WebMCQ, the meaning of the term ‘ease of use’ is made interesting by the need to cover two very different ‘uses’. First, the software must be easy for students to use as they complete a quiz, especially as it is designed to be capable of summative assessment as well as a learning tool, and as it will often be used by lone students without the presence of a guide. If students don’t find the software easy and intuitive, they will look to alternative forms of study, or they will use WebMCQ ineffectively.

In the case of software such as WebMCQ however, it is even more important that the software be easy for educators to use, for it is they who create the content to be presented to students. If educators find it difficult to use, or daunting to learn to use WebMCQ for content-creation, then there will be no quizzes, and the issue of ease of use for students becomes irrelevant. The issue of ease of use for educators was fundamental to the development of specialised software tools for creating WebMCQ quizzes (‘WebMCQ Creator’), and for accessing results and usage patterns (‘WebMCQ Analyser’). To make these tools as easy and intuitive as possible, several guidelines were developed. First, the software was to be made as simple as it could possibly be, without compromising desired flexibility. Second, the ability to access advanced features was never to be granted at the cost of making it more difficult to access basic features.

The third guideline, and the most important in terms of supporting the next wave approach, was that where possible, the software was to do things in ways that were already familiar to educators. This meant utilising familiar tools such as Web browsers, and using metaphors such as word-processors. By designing the WebMCQ interface to take advantage of skills already developed in familiar environments, it is almost possible to claim that a user can know how to use WebMCQ even before he or she knows what it does. This feature will be of central importance in next wave software, much of which you already know how to use, even though it hasn’t been conceived yet. In the same way that we have the potential to know the meaning of sentences before they are constructed, future software will all use a common ‘visual metaphorical language’, and after learning that language, new software interfaces will be comprehensible at first sight. The task of familiarisation with new software then becomes one of learning what the software can do, rather than how it is done, which leads to the fourth ease of use design guideline: where features of the program are unfamiliar, the interface should pro-actively guide users step-by-step. The fifth and final guideline was that a ‘Help’ button was to be visible at all times to offer immediate, context-sensitive assistance.

## 4.2 FLEXIBILITY

Flexibility, crucial to widespread adoption of next wave software, is the most challenging design principle to implement. The reason for this is that increasing the level of flexibility in software almost always forces compromises in some or all of the other principles. For example, more flexible interfaces mean greater complexity, and therefore a reduction in ease of use. Increased flexibility also means less consistency as the software is used in a greater variety of ways. Flexibility often takes advantage of recent innovations in hardware and/or software, and so can reduce accessibility to those whose IT configuration does not include the latest advances. And flexibility means that the software must account for a wider range of possible uses and work with a greater number of variables, often reducing efficiency.

In the case of WebMCQ, the task of balancing flexibility against other principles was made somewhat easier and more effective by the multiple stages of development through which the software passed. When the software was first created, it was relatively inflexible. After receiving a large number of ideas about possible new features, and after identifying the market potential of WebMCQ, a great deal more flexibility was added. After its second major trial (incorporating formative and summative assessment), flexibility was again tweaked to incorporate features that were still widely desired, and to rationalise others that were not found to be necessary. In the third stage, feedback from users of Creator and Analyser was sought, and further modifications were made. Of course, the development process is continual, with incremental improvements being made still.

## 4.3 CONSISTENCY

Consistency for students and educators refers to consistency with previously encountered software interfaces (e.g. web browsers), and an internal consistency such that the software’s behaviour is similar and predictable in different situations. Consistency is essential to reduce anxiety when using the software, and to minimise time required to become familiar with its use. At the same time, consistency should not unnecessarily limit possibilities for the software’s application. In WebMCQ, consistency was maintained by imposing a standard layout with common visual elements (buttons, navigation bars etc.) to all quizzes. WebMCQ is still flexible enough that content can incorporate different colour schemes, headers, images, different font styles, etc., but the underlying layout imposes the required level of standardisation to quizzes so that, for example, the multiple choice questions always look and feel like multiple choice questions, making it clear to users what they are expected to do at any point.

## 4.4 ACCESSIBILITY

One of the great advantages of the Web is its potential for platform independence. For software like WebMCQ, this greatly aids accessibility through consistency and ease of use, because a quiz can be created once, and accessed (in theory, at least) by any person using any type of computer. However, the rapid and unsupervised early development of HTML, coupled with equally rapid developments in hardware, operating systems software, and browser software, has resulted in a virtual world where programmers are faced with a fearsome array of different browsers on different platforms with differing (often faulty and idiosyncratic) implementations of ‘standards’. Problems of platform dependence have been replaced by browser dependence.

As HTML and related specifications expanded, so too did incompatibilities among browsers. For this reason, it was decided that WebMCQ would take a ‘lowest common denominator’ approach to hardware and software requirements. New technologies such as Java, Javascript and ActiveX were all rejected because, although they make it easier to implement the software’s features, every new technology prevents groups of users from accessing WebMCQ. As a result of this approach, minimal demands are placed upon the client web browser, and students can access WebMCQ quizzes with browsers as old as Netscape version 1.2. This means that the WebMCQ server software must do much more work and be far more intelligent when it comes to preparing content suitable for a range of browsers - the server must prepare layouts, respond to student actions, handle security issues, calculate results, etc. – but the benefit is that no students (within reason) are disadvantaged by using older versions of software. Another benefit is that WebMCQ can be expected to be far more robust, as it uses only the older, well established and better implemented features of browsers. Furthermore, the time taken for a browser to prepare and present a page of information is far less, because no complex translation or preparation of content is required.

The philosophy of accessibility also carries through to the content creation and analysis components of the software, which are accessed through Web interfaces. Though Creator and Analyser require more advanced browsers, they allow content to be created, edited, and reviewed from any point of access to the Web. No software needs to be acquired and/or installed on the user’s machine. Instead, the interface is accessed remotely, and WebMCQ software does all the required work behind the scenes. Changes to a quiz made using Creator are instantly reflected in versions of that quiz accessed anywhere in the world. It’s another example of software coming to the consumer.

### 4.5 INTEGRITY

The final point discussed here relates to the reliability, speed, and efficiency of the software, which were considered to be of paramount importance to the successful adoption of WebMCQ, and its ability to perform well as a mission-critical assessment application. Issues of integrity ought not be limited to next wave software, but in recent times, there has been a tendency toward releasing complex software packages that contain significant ‘bugs’ (cf. recent release histories of Netscape Navigator and Internet Explorer products). One of the reasons for buggy software releases is the conventional distribution model, which requires periodic releases of product upgrades. The problem is that once bugs become known, consumers must wait until the next release date of the product for bug fixes to be available. In the meantime, new features will be incorporated into the software to make the new release sufficiently attractive to consumers, and these new features bring a whole set of new bugs. Consumers must then wait until the next release of the product for new bug fixes to be available. In the meantime … etc.

One of the major advantages of next wave software is that if it resides on central servers rather than on individual consumers’ machines, the vicious upgrade cycle is eliminated. Bugs can be fixed automatically, and updates to software can be performed as soon as they are ready without any need for action by consumers. This will also force software developers to be much more careful about releasing new features, because they will not risk introducing problems into software used by consumers where the consumer did not make an informed choice to accept the risk inherent in new software. Control over the maintenance of software will entail a serious responsibility to ensure software integrity that no provider can afford to take risks with. So, upgrades will occur more often, they will not require any effort on the part of the consumer, and they will be far less prone to bugs than current releases. If problems do appear, they can be fixed automatically by the software provider. Even software upgrades will come to consumers!

## 5. THE NEXT WAVE ‘SERVICE’ MODEL

As mentioned above, WebMCQ has been developed to be offered as part of a service, rather than as a product. This decision was taken for a number of reasons, and brings a long list of benefits to consumers, including convenient acquisition and setup, improved ease of use, lower costs, more flexibility, better after ‘sales’ support, more involvement in future software developments, and automatic continual service improvements. Each of these benefits of the next wave service model of software distribution will be discussed briefly.

### 5.1 CONVENIENT SETUP

Next wave software removes barriers of acquiring and installing new software. New customers of WebMCQ simply subscribe over the telephone or via email, have a new account created for them, and usually within minutes are able to begin using the software. Note that this distribution model is very different from acquiring software by downloading it to one’s computer, then installing a copy on individual machines. Software distributed in this way is only installed once on the machines of service providers, part of whose service is to take care of the complexities of installation. As soon as a new customer gains access to the service, they can access it from anywhere there is a connection to the Web.

### 5.2 EASE OF USE

Again, ease of use is an important motivator to adoption and widespread use of next wave software. The ease of setup inherent in this model carries through to ease of maintenance for users, due to the fact that there is no need for maintenance! The responsibilities of maintaining and upgrading software fall to the service provider, rather than consumers who should not be expected to achieve the level of technical expertise required for effective maintenance.

### 5.3 LOWER COSTS

Subscription to a service replaces purchase of a product. In the case of WebMCQ, consumers pay a license fee to access the software, which is renewed on an annual basis. This model results in lower overall costs, since consumers only pay for the software as long as they wish to use it, and the costs associated with packaging and distributing a physical product are eliminated. It may appear to be more economical for consumers to follow the more common practice of one-off payment for a product, on the assumption that the higher initial fee 'buys' the software for eternity. But this doesn't account for the fact that consumers are then expected to pay for future upgrades, and that the software is generally obsolete within several years of purchase, so that any 'one-off' purchase is really only valid until the next 'one-off' purchase is required. The next wave distribution model provides cost benefits through greater economies of scale when it comes to software development, installation and maintenance. In addition, consumers are free from the costs associated with performing ancillary maintenance tasks, and receive the benefit of continual software improvements with zero effort on their part.

### 5.4 MORE FLEXIBILITY

Part of the next wave service will be an ability to tailor the service to consumers' specific needs. Rather than buying a mass-produced product in a box, services allow greater input from consumers and actually empower consumers because they can incorporate an interactive relationship with the service provider. Software will be designed and offered in modules, and offered as resources. Consumers will then be given choice about which parts of a service they wish to subscribe to, and which they don't want to pay for. Choice will be extended to other resources such as manuals, help references, and informative articles will be available as part of the service. These will be offered in a variety of forms, for example, in HTML, as word processor documents, or as printable files. The consumer will have flexibility in terms of what resources they wish to use, and how.

### 5.5 BETTER AFTER-SALES SUPPORT

Offering a service with regular re-subscription will motivate service providers to keep their subscribers happy. When selling a product, there is a temptation among some suppliers to adopt the attitude, "After we've got the money, our job's done." After-sales support is then difficult to attain, of poor quality, and/or incurs further expense. Discounted future product upgrades are offered as an effective control mechanism to existing customers, so that even disgruntled customers will choose to buy an upgrade from their current supplier simply because it is cheaper than switching to a new product. Furthermore, the software supplier takes full advantage of consumer resistance to adopting and learning new software. They know that with the current barriers to adoption in place, consumers are unwilling to change from a familiar product to an unfamiliar one. With these barriers dismantled by next wave software, suppliers will not be so complacent, and providers of products and services will all be forced to re-evaluate their commitment to consumers.

### 5.6 INVOLVEMENT IN FUTURE DEVELOPMENT AND CONTINUAL IMPROVEMENT

Following from the previous point, the existence of an interactive provider-subscriber relationship in which subscribers are empowered will make software upgrade developments more responsive to consumer needs. And as already explained, these upgrades will be delivered automatically and regularly in the next wave distribution model.

## 6. CONCLUSION

The recent rise of networked computers and advancement in computer software design brings us to an exciting point in IT development. It has been argued here that the internet will make possible a new model for software distribution, utilisation, and design. Software will move from current perceptions as a product, towards being perceived as a service. The burden of difficult technical tasks currently thrust onto consumers will become the responsibility of service providers. A host of other benefits will flow to consumers, leading in turn to an explosive increase in the adoption of the coming technologies. In the field of education, consumers are generally professionals who are typically time-pressured and have moderate to little experience with technical computer issues, and for this reason education is just one field that can benefit enormously from the ‘next wave’ in IT.

## 7. ACKNOWLEDGEMENTS

The authors wish to acknowledge the assistance of the following in the development of WebMCQ: the Department of Psychology and the Faculty of Science at the University of Sydney, and eventHorizon software.

More information about the WebMCQ service can be obtained by sending email to WebMCQ@psych.usyd.edu.au, or by visiting the WebMCQ homepage at http://www.WebMCQ.com.au/

## 8. REFERENCES

Alexander, Shirley. (1998) Internet-based teaching and learning: the past and the future. Proceedings of University Science Teaching and the Web Workshop. Uniserv Science, The University of Sydney, 5-6.

Brown, C. Marlin (1998) Human-computer interface design guidelines. Ablex Publishing Corporation.

Carroll, John M. (1997) Human-computer interaction: psychology as a science of design. Annual Review of Psychology, 48, 61-83.

Dalziel, J. R. and Gazzard, S. (1998) WebMCQ: A Web-based multiple choice assessment system. Proceedings of University Science Teaching and the Web Workshop. Uniserv Science, The University of Sydney, 25-27.

Gazzard, S. and Dalziel, J. R. (1997). How it looks from their side of the screen: Student evaluations of a World Wide Web tutorial in the Department of Psychology at the University of Sydney. Proceedings of the Fourteenth Annual Conference of the Australian Society for Computers in Learning in Tertiary Education. Curtin University of Technology, 213-218.

Lowther, Deborah L., Bassoppo-Moyo, Temba. and Morrison, Gary R. (1997) Moving from computer literate to technologically competent: The next educational reform. Computers in Human Behavior 14: 1, 93-109.