As Simple as Possible, as Complex as Necessary: An approach to the design and development of web-based learning environments

Julian Halliwell, University of Hull

Introduction

The Internet and its associated technologies have proved serendipitously to be able to offer the universal network platform upon which to build Illich's longed for "educational web" (Pickering, 1995). Instead of having to be content with the cost and limitations of discrete experimental infrastructures, developers of networked learning have been presented with a ready-made means of translating theory very rapidly into practice and of realizing their goal of new, potentially global teaching and learning environments based on accessibility, flexibility, empowerment, alternative modes of communication and so on.

Amidst the excitement of this unexpected achievement, however, it is easy to forget that the Internet was not designed for the purposes of mass participation to which it is now being put. Ordinary users – people without technical computing expertise of any kind – were very far from the minds of the Internet pioneers who could not have envisaged the outcome of their project. Email and the web have been successful amongst other reasons because: 1) they are genuinely and profoundly useful; 2) they have been developed in an "open source" spirit of universal access and standardization; and 3) by and large they work. On the other hand their success is most certainly not due to their inherent simplicity: they are far from being as straightforward to use as say a toaster, a telephone or even a television set.

Whilst acknowledging and welcoming the success of the Internet, and the web in particular, we should also recognize that this has been in spite of the fact that it is fundamentally complex as a system compared with the mass appliance technologies with which most people are familiar. This recognition should lead us, as both educational and technical developers of internet-based learning, to pay closer attention to counter-balancing the underlying user-unfriendliness of our adopted medium by placing greater value on simplicity and usability. Unfortunately, although ordinary users frequently appreciate these qualities, both pedagogical and technical designers (not to mention managers and decision makers) tend to place greater importance on having sophisticated, comprehensive and "cutting edge" functionality, without considering fully whether the resulting systems are appropriate or will ultimately be effective.

In this paper I will argue for greater emphasis to be placed upon simplicity as a key value in educational and technical design for web-based learning in the interests of achieving systems that are adequate, appropriate and effective. I will suggest that this approach is likely to lead more often to choices about what to *leave out* of the design rather than what to include. Complexity is not excluded, indeed it will often be unavoidable if the system is to be adequate, but it must be properly justified in the context in which the system is being developed and must not unnecessarily compromise effectiveness.

Why complexity should be avoided wherever possible

"Complexity means distracted effort. Simplicity means focused effort." (De Bono, 1998)

Complexity generally gives rise to difficulty. A system may be deemed to be "good" because it can fulfil a wide range of functions and satisfy many potential needs by presenting the user with a variety of features and alternative configurations from which to choose. Continually extending the range of options and functions without consideration of the effects on usability can, however, lead to uncertainty and confusion on the part of the user, who may encounter the need to make choices as a distraction from their primary goal of engaging in learning activity. Ideally the system should be transparent to the user, allowing them to focus on the course rather than on the system.

Conversely, a system may be outwardly straightforward but have an underlying structure which is unsustainably complex. This can lead to poor performance, unreliability or difficulty in maintaining and further developing the system. The more complex a system becomes, the more likely it is to go wrong.

Excessive complexity is not caused exclusively by the unchecked creative egos of technical developers. Much is made of the desirability of pedagogy-led rather than technology-led development, but both approaches can fall into the complexity trap. Clearly where technological imperatives are uppermost, there is a danger not only that usability and system performance will suffer but also that the resulting pedagogical model will be inappropriate. This I would identify as one of the key weaknesses of many "shrink-wrapped" commercial products which are marketed to appeal to the broadest range of educational providers, and offer only limited possibilities for customization. The features tables may be long and impressive, but this in no way indicates appropriateness in each of the learning contexts in which the system is deployed and it is often difficult to make adjustments without the intervention of specialist developers. On the other hand, where educational designers are given *carte blanche* – particularly those unfamiliar with online teaching and learning issues – the technical contortions sometimes performed in order to implement their requirements can result not only in confusing interfaces, but also in overly-complex underlying structures which may threaten to undermine the efficiency and sustainability of the system as a whole.

Seeking simplicity: the case of Merlin

The Merlin web-based learning environment, developed at the University of Hull, distinguishes itself most clearly from commercial "packages" by the fact that it has evolved, and continues to evolve in response to the specific local needs of the courses and initiatives it has been called upon to support (1). Continuous development implies gradually increasing complexity as the system extends to accommodate an ever wider range of disciplines, purposes and modes of delivery. The overriding objective, though, has been to produce a system which is not only *appropriate*, because it responds to genuine local needs, but also one which is and will remain demonstrably *effective*.

Achieving this second goal is, in my view, a matter of balancing the evolutionary drive towards complexity as demands on the system's capabilities increase, with a very deliberate effort to seek the simplest way of implementing genuine requirements adequately. Put differently, to be effective the system should remain *as simple as possible* and only *as complex as necessary*. The optimal balance is achieved on the one hand by ensuring that each new development justifies the complexity it introduces – in other words that it is based on the real pedagogic needs of the given context, and not the idiosyncratic inclinations of the tutor, designer or developer– and on the other by constantly searching for the simplest method of implementation.

Valuing simplicity should not be confused with adopting a simplistic approach. Achieving simplicity is itself not a simple matter and requires expertise, effort and a preparedness to completely rethink existing structures (see De Bono, 1998 for a fuller exposition). To repeat, richness and complexity have their place and will be inevitable to a greater or lesser extent; but optimal effectiveness will result from being able to keep complexity to a minimum, while still managing to produce an adequate solution. Of course, the use of such words as "necessary", "adequate", and "appropriate" implies subjectivity and discrimination, but it is precisely through experience and a broad understanding of the issues surrounding online teaching and learning – technical, educational, administrative – that informed judgements can be made. Achieving simplicity requires expert knowledge.

Example issues from the development of Merlin

Having set out the reasons for preferring a simplicity-oriented approach to system design, let us now illustrate it by considering a few examples of attempts to apply it in the development of Merlin.

Accessibility

Merlin is entirely web-based and does not require users to install any proprietary client software (such as Lotus Notes or FirstClass Client). Either of the two standard web browsers is sufficient to access and interact with most (though not all, as we shall see below) parts of the environment. Sadly, "standard" does not mean that Microsoft Internet Explorer and Netscape Navigator behave identically when rendering HTML pages, nor can their respective interfaces be referred to generically. For the system developer, the convenience offered to users by supporting both browsers is at the cost of , on the one hand extensive adjustments and sacrifices in relation to the underlying code to compensate for the browser disparities and ensure that the user's experience is the same for both, and on the other the need to be sensitive to the different interfaces when referring to browser menus and settings in user guides and support messages. Clearly this is more complex from the developer's point of view than if only one browser were to be supported, but the complexity is justified when set against the valuable simplicity of being able to access the system no matter which browser the user happens to have on their computer.

Interface design

A design is more likely to be perceived as "simple to use" if attention is paid to consistency, both internal and external. Until recently support facilities were provided in two locations in Merlin: a contextually sensitive guide was available in the top right of the screen layout accessed via an "i" (for "information") icon, and general support material was accessible by clicking a "?" button on the bottom left-hand side. Most Windows and Macintosh application interfaces use a "?" to signify "Help" and/or place a Help menu to the right of any other menus at the top of the application window. Observing that awareness of the available online help facilities appeared to be low amongst our users, we decided to unify all of the support information beneath a single "?" icon located in the top right of the interface layout. Not only was all help information simpler to access by being in one place only, it was more intuitively available by virtue of being signalled by a conventional icon and in a location consistent with other applications with which most users would be familiar.

Functionality

A common misapprehension when dealing with online course materials development occurs when tutors perceive that the web-based content templates are unable directly to handle tables, diagrams and the other layout functions they are accustomed to using in word-processing packages. A little lateral thinking brings the realization there is in fact nothing to stop them continuing to use all of the functionality of the their word-processor and then simply attaching the document for their students to download. This is admittedly slightly more complex for the users in that they must work with a second tool (albeit a very familiar one) to achieve their aims, but well justified by the advantage of being able to continue using the standard web browser instead of requiring a special client to be developed which can handle advanced formatting (the technical developer breathes a sigh of relief as the system can remain simple and sustainable!).

Audio integration

Having given three examples where simplicity has been achieved with only a relatively small cost in complexity, there is undoubtedly at least one aspect of Merlin in which a satisfactory balance has yet to be found. Since Merlin was originally developed as a platform for language teaching and learning, support for audio has always been a major feature of the environment. Web browsers and servers are themselves limited in their native ability to support audio (and video), but there are effective and freely available additional software programs which work alongside the browser to deliver acceptably high quality sound in almost real time, thanks to what is known as "streaming technology" (2). In addition to listening to recordings, users can create and share their own by using a second piece of free software together with a microphone and their computer's sound card. In principle the use of these two programs in conjunction with the Merlin web environment allows audio to be supported very effectively. In practice problems getting the audio to work continue to account for the vast majority of all requests for technical help from our users.

This is a difficult area to judge from a simplicity-oriented perspective. On the one hand there appears to be highly distracting complexity for all concerned: users, frustrated at the difficulty of getting the equipment to work; tutors, unable to rely on their students' ability to submit audio-based work; and technical staff bombarded with requests for assistance. On the other hand, audio adds an extremely powerful dimension to an otherwise text/visual-only environment: in terms of personalization, accommodation of learning styles, visual disabilities and dyslexia, and general enrichment of course content and social interaction. Audio will therefore continue to be supported in Merlin despite the regrettably limited scope for making it simpler to use.

Suggestions for a simplicity-oriented approach to design for web-based learning

To conclude I would like to offer a number of general recommendations to both technical and educational developers of online learning environments who wish to improve the effectiveness of their systems by placing greater emphasis on simplicity.

- Make simplicity, appropriateness, adequacy and effectiveness key values in your design approach. They should rank higher than comprehensiveness, being "cutting edge", matching the functionality of a rival system, or fit with an existing pedagogic/communciation model (are electronic whiteboards and real-time chat systems really worth the extra technical and usability burden they incur in every context?)

- Requirements should not be accepted at face value but should be subject to questioning and argument to ensure they are genuine and appropriate. Examine and re-examine each element of the existing or proposed design and try to identify any that are not absolutely essential for the purpose at hand. To discriminate what is important and what is not requires expertise. Educational designers must have a thorough understanding both of the subject matter and of online pedagogic principles to be able, for example, to design a task template which is simple to use but adequate for the context (in other words simple but not simplistic). Technical developers must have in-depth knowledge of their servers and operating systems to enable them to strip them down to the bare minimum components and thereby achieve optimally efficient performance and easier management.

- Opt for a modular design where possible. Where a complex structure seems unavoidable it can often be dealt with by assembling several simpler sub-structures. Merlin features a "core" framework largely consisting of communication tools used by all users, into which separately developed course and resource structures are "plugged in" according to the needs of the particular group. The environment is thus able to accommodate a wide and ever expanding variety of needs without a wholesale restructuring of the system each time.

- Consider who will benefit from introducing simplicity in each case. Given the central premise here that the inherent complexity of computers and networks requires counter-balancing, the beneficiary should normally be the end users of the system. As the example of document attachments versus radical software re-development above illustrated this will not always the case, at least in the short term. Users (online learners and teachers) will of course ultimately benefit from systems that are technically simple and efficient because they are more likely to be robust and reliable.

- Finally, be continually prepared to re-think the system design in the interests of simplicity. Systems will naturally tend to become more complex as they are developed, and so a constant quest for a simpler approach is needed which may from time to time involve re-designing certain elements – indeed perhaps the entire system – from scratch to ensure effectiveness is maintained.

Summary

In this paper I have drawn attention to the fundamental complexity of the platform upon which networked learning is largely being developed, and to the need to counter-balance this by making the learning systems we build simpler and thereby more effective. While complexity is unavoidable in some measure, where it is not fully justified it can lead to wasted effort and ineffectiveness. An approach in which simplicity and appropriateness are highly valued helps to ensure that effort at all levels is focused on the purpose of effective online teaching and learning. Prerequisite to a simplicity-oriented approach is a deep understanding of the nature both of the educational and technical issues involved: without expertise there is a risk of simplistic rather than simple and effective results.

Notes

(1) See the Merlin website at www.hull.ac.uk/merlin for more details.
(2) This is the RealMedia software available from www.real.com

References

De Bono, Edward. (1998). *Simplicity*. London: Viking.

Pickering, John. (1995). "Teaching on the Internet is Learning". *Active Learning*. Issue 2. July 1995.