

1. Accessibility and user interface design

There are a number of myths about Accessibility and websites:

- ◀ It means having a special text only 'dumbed down' website,
- ◀ Accessibility is hard to do and costs a lot,
- ◀ A site which has to be made accessible means it's boring.

But the truth is that designing for accessibility is not any of this. You can easily have a feature rich, attractive website that also works for people with accessibility needs. But let's start at the beginning and find out what this accessibility issue is all about.

1.1. What exactly is this accessibility issue?

According to ABS statistics, at any one point in time, around 20% of the Australian is experiencing a permanent disability of some sort. At least half of these people will have difficulty using a computer and would benefit from accessible design. Many other people benefit from accessible design:

- ◀ The 8% of the male population who has some form of colour blindness (less than 1% of females are colourblind,
- ◀ Anyone with a temporary injury who has trouble using a mouse or clearly seeing a screen (could be up to 10% of the workforce at any one time),
- ◀ Touch typists who choose not to use a mouse,
- ◀ User of PDAs and web-enabled phones.

Research by Forrester (for Microsoft, 2003) shows that 44% of working age computer users in the USA customise or modify in some way the computer interface (e.g. larger fonts, changes to keyboard or mouse operation) and that 17% use assistive technology (see below). While some of these people have a recognisable disability, the main driver is ease of use. See www.microsoft.com/enable/research/phase2.aspx for more information.

As you can see, building accessible online solutions is more than just providing the necessary features to support the technologies used by people with a disability, such as visual impairment, motor disabilities or intellectual disabilities. It is also about supporting older people (with variable font size), people in rural locations (with a light page weight) or people with colour blindness (by not using colour alone as a highlight).

The Australian Banker's Association (<http://www.bankers.asn.au/>) has published voluntary guidelines for the design of ATMs, EFTPOS, Telephone and Internet Banking services (<http://www.bankers.asn.au/Default.aspx?ArticleID=344>). Although they're

'voluntary', it makes more sense that as you upgrade your services, you design for accessibility from the beginning.

1.2. How do people with accessibility needs use online services?

People with accessibility needs use different assistive devices, some are built in to their PCs, while others are based on external hardware or additional software. They include:

- ◀ Built-in customising options (such as changes to font sizes and styles, changes to mouse appearance and operation, changes to keyboard operation, changes to the operation of warnings and notifications),
- ◀ Simple built-in aids (such as magnification or read aloud options),
- ◀ Alternative input devices (voice input programs, alternative keyboards, alternative mouse devices),
- ◀ Vision technology (screen readers, screen magnifiers, text-to-speech software, refreshable Braille devices).

You can see pictures of these at <http://www.utoronto.ca/atrc/reference/tech/techgloss.html>.

1.3. Coding for accessibility

To support people who use assistive devices, you must do the following:

- ◀ All functions must work using a keyboard as well as a mouse (including, scripts, hovers, onmouseover etc.),
- ◀ Content must make sense when rendered linearly. Assistive technology works with the code, so that the underlying code must make sense, not just the visual presentation,
- ◀ Non-text elements (images, multimedia etc.) must have text equivalents,
- ◀ Structure must be created with code not visual presentation (e.g. headings, lists, tables, forms).

The first place to start is the W3C's Web Accessibility Initiative's website (www.w3c.org/wai/). There is a wealth of information on guidelines, checklists and techniques to assist you while developing your website. You can also visit our industry website at www.it-test.com.au.

In essence, making something accessible is about correctly coding your site, using the appropriate HTML tags and other features. As with the development of any application, if you design and code for accessibility from the beginning, it's much easier and cheaper than having to do it in retrospect because someone takes you to task.

We've assembled a brief list of the top mistakes of designing and developing accessible solutions:

- ◀ Information presented in tables that are not properly marked up with headers so they are incomprehensible for assistive technologies,

- ◀ Information presented in complex multilevel tables when a series of simple tables would work much better for everyone and reduce the need for complex accessibility markup,
- ◀ Key functionality that only works for users with a mouse,
- ◀ Functionality that works with a keyboard, but is practically unusable,
- ◀ Pages using CSS for positioning, but with the order of the divisions in the underlying code completely at odds with the visual layout,
- ◀ Links that do not clearly describe their destination (4 'log on's, a few 'click here's and lots of 'more's is a common scenario),
- ◀ Structural elements (lists, heading, forms etc.) created with visual presentation, not using proper structural code,
- ◀ Images without alternative text so any one or any thing surfing without images (such as Search Engines and people who are blind) get no information.

1.3.1. Accessibility and online security

Several banks are experimenting with onscreen keypads (number, QWERTY or both) for security. These use rotating keys in an attempt to combat key loggers, and can also change location on the screen. Whatever their merits as a security solution, rotating keys immediately make the online banking environment inaccessible to anyone who cannot use a mouse (older people, people with arm or hand injuries or people with vision impairment).

Other security options being considered include randomly generated GIF file images of numbers or letters. These CAPTCHAs (Completely Automated Public Turing-test to tell Computers and Humans Apart) have caused considerable accessibility problems, but there may be a workable solution. The problem is covered in <http://www.w3.org/TR/2003/WD-turingtest-20031105/>, while the solution is covered in <http://www.standards-schmandards.com/index.php?2005/01/01/11-captcha>. You can read a general discussion about CAPTCHAs here <http://en.wikipedia.org/wiki/Captcha>.

1.3.2. Accessibility and Usability

An online banking application is accessible when a person with an impairment is able to use data, information and services as effectively as someone without a disability - not necessarily in the same way, but equally effectively. Compliance with technical rules and standards is necessary, but not sufficient.

For example, a scripted flyout menu needs to be accessible without a mouse, but only the first level navigation should be active for a keyboard user. The link should take the user to another page containing the second and third level navigation. One major finance site requires 280 hits of the tab key to get through every flyout option before it is possible to access information from the last main first level element in the navigation (and a further 137 tabs if you wanted to get to the last visible first level element). While this site may be technically accessible, someone with a physical injury or impairment will find it practically unusable.

1.3.3. Accessibility and rich internet applications

Rich Internet applications are a growing part of online banking. These can cause accessibility problems if they are not implemented carefully. But, there is substantial documentation on how to build these applications properly.

Macromedia has worked particularly hard to develop accessibility techniques for Flash. These are at: <http://www.macromedia.com/macromedia/accessibility/features/flash/> and <http://weblogs.macromedia.com/accessibility/>.

AJAX technologies are getting a lot of attention and they have the potential to solve some difficult accessibility problems when pages change state. Like Flash there is good documentation on how to make the resulting online service accessible. You can get more information here

<http://www.adaptivepath.com/publications/essays/archives/000385.php>, here <http://en.wikipedia.org/wiki/AJAX> and here http://sourcelabs.com/ajb/archives/2005/05/ajax_mistakes.html.

1.4. Conclusion

There are plenty of resources to help to design your site right the first time to work for people with accessibility needs. You also need to test your site to ensure you've coded it right. Tools like Bobby (<http://www.watchfire.com/products/desktop/bobby/default.aspx>) help, but you need to engage expert consultants to assist in the testing process.

It's always better if you design it right the first time.

2. About the primary Author

Craig is the founder and Managing Director of The Performance Technologies Group (PTG Global), with over 15 years in user experience, user interface design and change management.

Craig runs the R&D function at PTG, having produced a number of world firsts including XPDesign – the first systematic methodology for user interface design and Certified Usable – the first guarantee for usability and user experience.

Craig has been the primary architect behind many of Australia's most popular websites including CBA, Virgin Blue and ASIC and works on cutting edge technologies such as touch, medical and special-purpose applications.

Craig holds a Masters qualification in organisational psychology, is a member of the APS and the APS College of Organisational Psychologists and is a Registered Psychologist in NSW. He is also an Associate of the University of NSW and Macquarie University.



Contact Craig on:

Email: craige@ptg-global.com

Phone: +61 (0)2 9251 4200

Mobile: +61 (0) 416 266 216

Address: Level 16, 207 Kent St, Sydney, NSW, 2000, Australia