Designing for ADHD: in search of guidelines

Lorna McKnight
ChiCI Group, University of Central Lancashire
Preston, UK. PR1 2HE
+44 (0)1772 895152
lmcknight@uclan.ac.uk

ABSTRACT
Children with ADHD (attention deficit hyperactivity disorder) are a group at risk of marginalisation and isolation from their peers, as they are often placed in the same educational settings as more able children, despite having quite different needs. This paper explores how software designers who are not familiar with this group of users could use their existing knowledge of usability issues to design software that avoids excluding these children, while still designing software that is usable for all. A series of 15 guidelines are proposed from a survey of literature and recommendations, which were found to compare closely with usability guidelines in many cases.

Keywords
Usability, ADHD, children, software design, guidelines.

INTRODUCTION
One of the fundamental principles of Human Computer Interaction is to 'know your users'. When designing for a particular user group, software designers are expected to take into account the needs and desires of that group, to produce software that will be appropriate for them in terms of usability and functionality. However, users within any group will be varied, and designers can often be unaware of the specific needs of minorities within these groups.

This paper is written in the context of the UMSIC project, which aims to support social inclusion in primary school children through the design of music-making software on mobile devices. This requires bringing together expertise from a wide range of disciplines, such as interaction design, software development, psychology and learning technologists. It also involves developing an understanding of some of the groups of children who are likely to struggle with inclusion by their peers, to ensure that any products developed address their needs suitably.

In the UK, children with non-extreme cases of ADHD (attention deficit hyperactivity disorder) are normally placed in mainstream education rather than specialist schools, meaning that they often have to use the same learning materials and technologies as the other pupils. However, they may have special education needs that are not supported by these tools. In the interests of fostering social inclusion, no child should feel that they are forced to use technologies that disadvantage them, or resort to using different tools from the rest of the class. This means that designers of software for children need to take the needs of this specialist user-group into account, so as to avoid marginalisation of the disadvantaged minority.

For software designers concerned with usability in general, the literature is extensive and well-established. For designers of interactive software for children, there is a growing supply of literature available (e.g. see [10], [7]), allowing practitioners to increase their knowledge of child-specific issues, and to support newcomers to the field. However, designing for a more specific minority, such as children with ADHD, may seem like a daunting task. If the numbers of child-computer interaction specialists worldwide are limited, then it is to be expected that researchers with specific expertise for minority groups would be even rarer. Added to this, a standard class in a school will usually have very few pupils with special education needs (SEN pupils), so it can be hard to gain experience with these users. It is understandable therefore that designers may wish for a set of guidelines to steer their designs, and provide some initial support in this demanding task – however, such guidelines can be difficult to find.

ATTENTION DEFICIT HYPERACTIVITY DISORDER
ADHD has been cited as the most common behavioural disorder in the UK, with some estimates suggesting that it affects between 3-9% of children [12]. It is often characterised by a difficulty in holding attention to a task, or in controlling impulsive behaviour. Because of the impact that attentional difficulties can have on academic performance, ADHD can also be classed by some as a learning disorder [3]. Diagnosis is usually performed by a professional healthcare expert such as a child psychologist, and it would be easy for a casual observer to dismiss ADHD symptoms as misbehaviour, day-dreaming, or lack of attention, which are already common in young children. In fact, children with this condition may be receiving a range of treatments, such as specialist support, behaviour management practices, psychotherapy, and in some cases perhaps even medication. A report by the National Attention Deficit Disorder Information and Support Service

(ADDISS) also highlights the difficulties that ADHD children may have in social situations due to their condition [3], meaning that social inclusion is likely to be difficult.

There is some question as to whether computer applications are most suitable for children with ADHD. Guidelines for behaviour management often suggest reducing technology use and increasing the amount of exercise. However, one study does suggest that collaborative work on a computer was successful for increasing peer acceptance of a child with ADHD [17]. Also, since technologies are unlikely to be removed from schools any time soon, it seems unfair to exclude already-marginalised children from the use of the same tools as their peers. It therefore seems most suitable to consider how such tools could support this group, but without also disadvantaging the majority. It is at this stage that guidelines would be useful for supporting this process.

GUIDELINES FOR ADHD

For most researchers, a search for guidelines on designing for ADHD might normally begin by searching academic databases (e.g. ACM Portal, ScienceDirect, IEEE Xplore or ISI Web of Knowledge). However, searching these sources was not found to yield much relevant information: the literature from the HCI community on this topic is minimal or non-existent, while other literature seems mostly concerned with diagnoses and clinical trials, which is perhaps not the easiest for designers to digest. Therefore, the search for information was widened to include general guidelines for teachers and parents, to try and better understand the issues that need to be considered with this group of users. Information was collated from a wide variety of sources (e.g. [12] [8] [1] [2] [4] [15] [5]), and grouped into clusters of similar recommendations. Some topics were then excluded from this list of information as they seemed beyond the control of software designers, for example recommendations concerning medication, diet and sleep routines. Finally, a list of the 15 most relevant guidelines were derived from all the information available.

It is worth noting that these guidelines were not produced with software in mind, and are concerned with behaviour management techniques and the design of educational materials. It is also worth noting that many of these recommendations are based on advice from support agencies, which do not always cite scientific basis for their suggestions, so this paper makes no strong claims as to their effectiveness. However, from looking at the guidelines, similarities could immediately be seen between these recommendations and what is already known from usability guidelines for software design. Therefore, this section will discuss the implications of these guidelines for software design, highlighting where each guideline is supported by usability literature, or where it could cause conflicts.

1. Design materials so the layout is neat and uncluttered.
Although intended for paper materials, this guideline could easily apply to software interfaces. Indeed, it is already in line with usability guidelines on uncluttered interfaces and minimalist design to reduce confusion and search times needed to find information (e.g. [18]: G6.1 ‘avoid cluttered displays’, G16.7 ‘display only necessary information’). This guideline therefore seems appropriate for all users.

2. Provide a ‘calm’ environment, with soothing colours. No decorations or distractions.
Although intended for designing a home or classroom environment, this suggestion could apply to a software environment. The advice on minimal distractions is already in line with standard usability guidelines (e.g. [18]: G11.6 ‘use attention-attracting features [only] when appropriate’, G14.1 ‘use simple background images’), and this seems suitable for all users to avoid confusing images and unnecessary distractions. The advice on ‘calm’ and ‘soothing’ colours may be of note to graphical designers however, particularly for the design of children’s software where bright attractive colours are often used.

3.Provide a high-reinforcement environment – reward good behaviour and completion of all tasks that are asked of the children, using positive language.
The use of reward structures and congratulations on task completion is often seen in children’s games, and this seems a common feature that is likely to be suitable for all users. The use of positive language is also in line with usability guidelines (e.g. [18]: G15.10 ‘write instructions in the affirmative’). However reward structures are seen far less in software tools – for example word processors, drawing packages, calendars/planners etc. Designers could consider building these features into software, but there is a risk of the tools becoming patronising for other users if overused.

4. Organise items in an orderly way.
The need for organisation and consistency is one of the most common rules of interface design (e.g. [16]: the golden rule to ‘strive for consistency’; [13]: the heuristic of ‘consistency and standards’; [18]: G6.2 ‘place important items consistently’, G6.7 ‘align items on a page’, G11.2 ‘format common items consistently’, G11.4 ‘ensure visual consistency’, G16.4 ‘group related elements’). This helps users to find information more easily, without distractions or confusion. This guideline therefore seems highly appropriate for all users.

5. Distinguish important information by putting it in bold or colour. Signpost sections and group related information into panels.
This guideline relates to guidelines 1 and 4 on organisation and consistency. Important information should be easiest to find, and users should be able to identify the key features of an interface quickly. This idea is already amply covered by the usability literature (e.g. [18]: G6.2 ‘place important items consistently’, G7.2 ‘differentiate and group navigation elements’, G11.5 ‘use bold text sparingly’, G11.10 ‘emphasize importance’, G16.4 ‘group related elements’, G16.9 ‘use color [sic] for grouping’); therefore this seems suitable for all users.

6. Use large print (12-14 point) and a clear sans-serif font such as Arial.

While sans-serif fonts are often used for children, research has not shown any reliable and clear benefit in this form of typeface over serif fonts for readability [19] [14]. In terms of size, 12-point is often suggested for use on a screen (e.g. [18]: G11.8 'use at least 12-point font'), and specifically for children one study recommends the use of between 8 and 12-point fonts on mobile devices [6], although it is worth remembering that the point size is a relative measure, and absolute size will vary depending on the font used. While the literature on this topic is divided as to the best form of font to use, it is worth designers bearing in mind, as always, the readability of any fonts that are used, and ensure that they are clear on the type of display they are designed for.

7. Help pupils follow text by writing/highlighting alternate lines in different colours.

This is a feature that is often seen in many types of software, and 'zebra tables' [11] are often recommended by designers where lists of information are used. However, designers should be cautious about conflicting with other guidelines about simple displays without too much colour and distractions. Where possible, long lists of information should perhaps instead be avoided, and there is some evidence that it would be better to separate such information into different screens if it is needed (e.g. [18]: G8.4 'use paging rather than scrolling', G8.5 'scroll fewer screenfuls'). However, most software designed for children aims to avoid using large amounts of text, due to the difficulties associated with reading on computer screens.

8. If the pupil needs to work through a series of questions, help them keep their place by using a marker.

Users quite often need to work through a series of screens, or a sequence of tasks. Making it clear what point in the process they are at could help them keep track of their progress through a game or long activity (e.g. [18]: G7.4 'provide feedback on users' location'), and where this does not conflict with other guidelines (e.g. avoiding a cluttered interface) this sort of information could benefit all users.

9. Use brief and clear instructions.

The use of simple and unambiguous language is already highly recommended in usability literature (e.g. [18]: G15.2 'avoid jargon', G15.3 'use familiar words', G15.7 'limit the number of words and sentences'), and particularly guidelines on designing instructions for children [7] [9]. Clarity and concise language is likely to be beneficial for all users, so this guideline seems appropriate for all.

10. Allow ample rest periods and exercise breaks.

This guideline may seem out of the control of software designers, and more for parents, teachers and other carers or educators. However, it is an important point to consider, for the sake of better understanding the needs of the users. Software could aim to encourage this, if desirable, by suggesting that users take a break after long periods of usage, or even by limiting usage to a certain length of time per day. However, another suggestion might simply be that designers should support the desire of their users to take breaks by allowing quick and easy saving and exiting from the software, and resuming without hassle. This is likely to prove beneficial to all users, particularly children who need to stick to a strict timetable within school hours.

11. Have a work station that is enclosed, in a soundproof environment, with few distractions around.

Related to guideline 2, the key issue here is of minimising distractions. It is important for designers to be aware of the context in which their software will be used – for example, audio output may require the use of headphones. In particular, software designed for mobile devices does not tend to assume this sort of isolated, static environment. However, there is no reason why users could not work in an enclosed space with a mobile device – in fact, a portable device could enable users to move to a private location or another space that is most suitable for their needs, rather than being constrained by the layout of desktop machines.

12. Keep technology shut away unless it's being used.

Related to guideline 10, designers should be aware that teachers/parents may not want children playing for long periods of time, or may need them to finish a task at the end of a lesson. Work or games should be easy to save and finish at any point, and resume easily later. As a suggestion, mobile devices may perhaps be more suitable for this user group, as they can be physically put away in a cupboard or drawer more easily than a desktop computer can.

13. Keep to a routine, e.g. don't change teachers.

Again, this seems more of an issue for carers and educators than designers. However, it is worth noting as a potential issue for researchers and developers wishing to work with children – sensitivity needs to be used in introducing new faces and new technologies, frequent updates and changes to software may be undesirable, and the implications of taking technology away at the end of studies should also be carefully considered. Appreciating the need for routines may help in gaining co-operation from parents and teachers.

14. Minimise surprises.

This guideline is linked to the guidelines on maintaining a routine and providing consistency. Unexpected behaviour is usually undesirable in software, and generally linked to lower task efficiency. However, this may cause difficulties with game design, where a lack of surprise could easily lead to tedious gameplay. Designers should avoid unwelcome surprises, and where they are aiming to surprise users they should consider why they wish to do so, and whether it is an essential feature for engagement or simply a distraction.

15. Maintain eye contact.

While obviously difficult for software to achieve, this is cited as an important strategy in maintaining attention. While software could use eye-tracking techniques, or on-screen characters who look directly at the child, it may be that there are simpler methods for determining if the child's

attention has wandered, e.g. detecting a long delay on input, or repeated input that is irrelevant to the task. On detecting a lack of attention the software could give a notification asking for input, or suggest tasks to attempt instead. It may be worth designers considering such features for all users, as such delays may also indicate boredom or confusion.

DISCUSSION

This paper has described a guideline-gathering exercise, which has yielded 15 guidelines that may be of use to software designers. Obviously these guidelines are far from exhaustive and untested in the context of software design, and much more research would be needed in order to put these forward as recommendations on designing for ADHD.

However, the most useful discovery from this exercise has been in noting how many of these guidelines relate closely to established usability guidelines. An expert designer who is sensitive to usability issues should be capable of designing a product that is at least not damaging for this user group, simply by applying common-sense rules. More importantly, it should also be possible to design a system that is suitable for ADHD and non-ADHD children, thereby reducing the risk of marginalisation of this group through unsuitable software. All that is needed is an understanding that usability principles may become more critical to follow in this instance – for example, a cluttered interface is not recommended for any users, but is usually tolerated, whereas if the user suffers from attentional difficulties then it becomes a much more serious issue. Designers should also be aware that features added purely for entertainment value may in fact be damaging to usability, and will have to consider the value of all features they add to an interface.

As a final note, it is worth remembering that users, whether they have ADHD or not, are all different, and their needs and desires will vary greatly between cultures, ages, and abilities. While guidelines such as these may be useful in assisting non-experts to guide their designs, they may also wish to consider allowing the users (or a parent or teacher) to tailor the system to their personal requirements. Guidelines are always by their nature a general rule-of-thumb rather than a system requirement, and would only ever be a starting point to guide initial designs. As always, user testing is highly recommended as a more reliable method of testing suitability with a target user-group.

ACKNOWLEDGMENTS

This work was conducted as part of the UMSIC project (www.umsic.org). Thanks also go out to the rest of the ChiCI Group for their help and support.

REFERENCES

[1] ADDISS: The National Attention Deficit Disorder Information and Support Service, UK. www.addiss.co.uk/index.html

[2] ADHD Support, Wayne PA. www.adhdsupport.com/adhd-organizational-tips.aspx

[3] Bilbow, A. (2005) *School Report: perspectives on ADHD.* ADDISS: UK. Retrieved April 2010 from www.addiss.co.uk/schoolreport.pdf

[4] Bussing, R. et al. (2006) What 'Dr. Mom' ordered: a community-based exploratory study of parental self-care responses to children's ADHD symptoms. *Social Science & Medicine,* 63 (4).

[5] Coventry City Council (2010) Attention Deficit Hyperactivity Disorder (ADHD): strategy sheet. Coventry: UK. Retrieved April 2010 from www.coventry.gov.uk/ccm/cms-service/download/asset/?asset_id=18830021

[6] Darroch, I. et al. (2005) The effect of age and font size on reading text on handheld computers. *LNCS.* Springer.

[7] Druin, A. (ed.) (1998) *The Design of Children's Technology.* Morgan Kaufmann.

[8] Dunn, S. & Prever, M. (1997). *Understanding ADHD.* Mind: UK. Retrieved April 2010 from www.mind.org.uk/help/diagnoses_and_conditions/adhd

[9] Gilutz, S. & Nielsen, J. (2002) *Usability of websites for children: 70 design guidelines.* Nielsen Norman Group.

[10] Markopoulos, P. et al. (2008). *Evaluating Children's Interactive Products.* Morgan Kaufmann.

[11] Miller, D. (2004) *Zebra Tables.* Retrieved April 2010 from www.alistapart.com/articles/zebratables

[12] National Health Service (2008) *Attention-deficit-hyperactivity-disorder.* Retrieved April 2010 from www.nhs.uk/Conditions/Attention-deficit-hyperactivity-disorder/

[13] Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons.

[14] Poole, A. (2005) *Literature Review: Serif vs. Sans Serif Legibility*. Retrieved April 2010 from www.alexpoole.info/academic/literaturereview.html

[15] Rougeaux, R. (2005). *Designing a calm home for ADHD.* Retrieved April 2010 from http://newideas.net/adhd/parenting-adhd/calm-home

[16] Shneiderman, B. (1998) *Designing the User Interface (3rd Ed.).* Addison Wesley.

[17] Tan, T.S. & Cheung, W.S. (2008) Effects of computer collaborative group work on peer acceptance of a junior pupil with attention deficit hyperactivity disorder (ADHD). *Computers & Education*, 50 (3).

[18] US Department of Health and Human Services (2004) *The Research-Based Web Design & Usability Guidelines.* Retrieved April 2010 from http://usability.gov/guidelines/guidelines_book.pdf

[19] Walker, S. and Reynolds, L. (2002) Serifs, sans serifs and infant characters in children's reading books. *Information Design Journal*, 2 (2).