Engageability: a new sub-principle of the learnability principle in human-computer interaction

B CHIMBO, J H GELDERBLOM AND M R DE VILLIERS[1]

Abstract

The learnability principle relates to improving the usability of software, as well as users' performance and productivity. A gap has been identified as the current definition of the principle does not distinguish between users of different ages. To determine the extent of the gap, this article compares the ways in which two user groups, adults and children, learn how to use an unfamiliar software application. In doing this, we bring together the research areas of human-computer interaction (HCI), adult and child learning, learning theories and strategies, usability evaluation and interaction design. A literature survey conducted on learnability and learning processes considered the meaning of learnability of software applications across generations. In an empirical investigation, users aged from 9 to 12 and from 35 to 50 were observed in a usability laboratory while learning to use educational software applications. Insights that emerged from data analysis showed different tactics and approaches that children and adults use when learning unfamiliar software. Eye tracking data was also recorded. Findings indicated that subtle re-interpretation of the learnability principle and its associated sub-principles was required. An additional sub-principle, namely engageability was proposed to incorporate aspects of learnability that are not covered by the existing sub-principles. Our re-interpretation of the learnability principle and the resulting design recommendations should help designers to fulfill the varying needs of different-aged users, and improve the learnability of their designs.

Keywords: Child computer interaction, Design principles, Eye tracking, Generational differences, human-computer interaction, Learning theories, Learnability, Engageability, Software applications, Usability

Disciplines: Human-Computer Interaction (HCI) Studies, Computer science, Observational Studies.

1. Introduction

The classic principles and guidelines for software design were aimed at improving work performance and productivity, aspects relevant mainly to adult users. Many of these principles cannot be applied to children's products, since their needs, skills and expectations differ greatly from those of adult users (Chiasson and Gutwin, 2005). It is generally known that

[1] B Chimbo, J H Gelderblom and M R de Villiers are members of the academic staff in the School of Computing, University of South Africa, Pretoria, Their respective contact details are: chimbb@unisa.ac.za, gelderjh@unisa.ac.za and dvillmr@unisa.ac.za

adults and children learn differently, and our assumption is that this has not been taken sufficiently into consideration in software design. Children are a major user group and software aimed at them is mainly designed by adults. If these adult designers want to adequately address the needs of child users, they need a clear understanding of the differences between the requirements of child and adult users. Guidelines that distinguish between users of different age groups will support this. This study is a trans-disciplinary venture that crosses boundaries between domains to create a more holistic approach. Trans-disciplinarity implies the integrated application of various scientific theories, as is done in this research, which combines the disciplines of software design, learning and human-computer interaction. We explore frontiers of knowledge as we compare how children and adults learn to use software applications and, in particular, identify the value of engageability as an attribute that enhances learning by children.

Various human-computer interaction (HCI) researchers have proposed guidelines aimed at design for children (Fishel, 2001; Baumgarten, 2003; and Gelderblom, 2008). This study attempts to augment these ventures. It does so by applying the literature and integrating research methodologies from various disciplines, including HCI, adult and child learning, learning theories and strategies, usability evaluation, and interaction design. In an innovative approach, we investigated learning experiences by observing peer-teaching activities using both live observation and technological observation with usability testing equipment. In peer teaching, a participant who was familiar with an application taught a novice participant how to use it. Self-teaching activities were also observed and for these we added eye tracking recordings to our observations.

Dix, Finlay, Abowd and Beale (2004) provide designers of software interfaces with a comprehensive set of high-level directing principles with the aim of improving the usability of systems. They divide their principles into three categories, namely learnability, flexibility and robustness. Learnability is one of the quality components of usability; it refers to a measure of the degree to which a user interface can be learnt quickly and effectively. In their definition of learnability, Dix et al. (2004) include five sub-principles, namely predictability, familiarity, synthesisability, generalisability and consistency. These aspects are discussed further in Sections 2.2 and 5.1 to 5.5.

The focus of the research on which this article is based, was to investigate the learnability principle to establish whether it should be applied differently to software for varying age groups, thereby supporting designers in meeting different generational needs. The results culminated in the reformulation of the learnability principle to distinguish between its meaning for adult and child users. In pursuit of our goal, we compared the way in which children (aged 9 to 12) and adults (aged 35 to 50) learn to use unfamiliar software applications.

We set out to achieve the following:

- identify interface aspects which are complex for adults, but not for children, as well as those which are complex for children, but not for adults;
- identify patterns in the learning behaviour of adults and children, respectively; and
- use the above to compare how children and adults learn to use software, thereby broadening the definition of the learnability principle.

The formal research questions that underlie the study are presented in Section 3. A series of empirical studies was conducted with child and adult participants, using software aimed at children and software aimed at adults respectively. Data was collected through interviews, observations, video recordings and eye tracking.[2]

A comparative analysis of the meaning of the learnability principle for adults and children, based on their use of four software applications – Timez Attack, Story Book Weaver, MS Outlook and MS PowerPoint – indicated that there are differences in the way adults and children learn. (The four applications are described in section 3.1.) Notable differences were observed. For example, adults read instructions during initial stages of using the software, whereas children start to explore the software in a trial-and-error manner, without consulting instructions. The differences revealed aspects of software interfaces that adults and children approach differently, which has implications for the interpretations of the learnability principle. There are differences that are not covered by the current definition of the principle. This gap led to the proposal of an additional sub-principle, namely engageability that incorporates aspects of learnability that are not covered by the five existing sub-principles. Engagement is crucial for learnability. What makes a product engaging for children is different from what would make it engaging for adults.

The re-interpretation and expansion of the learnability principle and the design recommendations are intended to help designers of computer software address the varying needs of different-aged users, thereby improving the learnability of their designs.

We next provide a review of relevant research and theory, covering work from different disciplines including theories of learning, adult learning, HCI and interaction design. The research design and methodology are explained followed by a discussion of the research results. The latter include a discussion of the meaning of learnability for children and adults and a reformulation of the learnability principle. The findings from the research are summarised and conclusions are drawn. Finally, we explain the contribution of the research, some limitations and aspects that require further investigation.

2. Literature review

As explained in the introduction, there is need to augment and refine the existing principles for the design of technology specifically targeted at children. Our focus is on principles to improve the learnability of software applications. This section gives a brief introduction to the fields of HCI and interaction design and then explores existing research concerning the learnability principle and, more generally, the research and theory related to how children and adults learn.

2.1 HCI, interaction design and child-computer interaction

The term HCI was adopted in the mid-1980s to denote a new field of study concerned with studying and improving the effectiveness and efficiency of computer use (Kotzé and Johnson, 2004). HCI is concerned with the design, implementation and evaluation of interactive systems, with specific consideration of what a user needs to accomplish when using the

[2] Eye tracking is a technique which determines eye movement and eye-fixation patterns. A full definition is given in Section 3.2.

system (Dix et al., 2004). HCI is a multi-disciplinary subject with computer science, psychology and cognitive science at its core (Dix et al., 2004).

Closely related to HCI is interaction design, which Preece, Rogers, and Sharp (2007:8) define as 'designing interactive products to support the way people communicate and interact in their everyday and working lives'. The focus is on how to design user experiences using a variety of methods. Interaction design places emphasis on user participation in the design process, but according to Preece et al. (2007), it is equally important to understand how people act and interact with one another, with information and with technology; and their abilities, emotions, needs and interests.

Historically, computers and computer applications have been designed for use by adults for assistance in work-related pursuits. In many accepted definitions of HCI and interaction design, there is a hidden assumption that the users are adults.

There are, for example, references to users' 'everyday working lives' (Preece et al., 2007) or the 'organization' they belong to (Dix et al., 2004). Children, however, make up a substantial part of the larger user population. Whereas products for adult users usually aim to improve productivity and enhance performance, products for children are aimed more at providing entertainment or engaging educational experiences.

Child-computer interaction has emerged in recent years as a specialised research field within human-computer interaction. This is manifested in the annual Interaction Design and Children conference that has been held since 2002. Until recently, only developmental psychologists, educationists, and market researchers were interested in children as users of interactive technology. Currently however, numerous researchers in the field of HCI are focusing their attention on design of children's technology.

Technology has become an important element of the context to which contemporary children belong and it is important to understand its impact on children. The development of any technology can be successful only if the designers truly understand the target user group. The present research contributes towards that understanding, by specifically investigating what learnability of software interfaces means for users of different ages.

2.2 The learnability principle

The aim of design guidelines, standards and design principles is to help designers to improve the usability of their products by giving them rules according to which they can make design decisions (Dix et al., 2004). These rules restrict the range of design options and prevent the designer from making choices which are likely to harm the usability of the product. Learnability is related to the time it takes users to learn to use the commands for a task or the effort of a typical user in performing a set of tasks on an interactive system (Nielsen, 1994; Santos and Badre, 1995; Shneiderman, 1997; Dix et al., 2004; Senapathi, 2005). Learnability comprises specific measurable attributes and can be evaluated by measuring them in a real-life context. It deals with the initial understanding of the system by the novice user, as well as the attainment of maximum performance once users have learnt how to use it (Preece et al., 2007). In the HCI context, learnability relates to interactive features that help novice users learn quickly and that expedite steady progression to expertise.

Various sub-principles that affect learnability have been identified (Preece et al., 2007; Dix et al., 2004; Senapathi, 2005; Aspinall, 2007). As mentioned before, Dix et al.'s classic set of principles are predictability, synthesisability, familiarity, generalisability and consistency. Predictability refers to the ease with which users determine the result of future interface interactions, based on the past interaction history (Dix et al., 2004). A predictable system is easy to learn. Synthesisability is the system's ability to provide an observable and informative notification about its internal changes of state (Aspinall, 2007). When a system is synthesisable, users can assess consequences of their actions. The familiarity principle relates to the ability of an interactive system to support the mapping of prior experiences onto a new system (Dix et al., 2004). Familiarity impacts on novices' initial perception and whether they can determine the required actions from their own prior experience. Generalisability supports users in extending their knowledge of interaction in and across other applications, to new, but similar, situations. To support generalisability, consistency is essential, with systems offering similar functionality in comparable situations (Dix et al., 2004; Preece et al., 2007; Nielsen, 1994).

2.3 Learning theories and definitions of learning

Learning is the process of transforming experience, skills and attitudes and involves various sub-processes (Siemens, 2006). It is broader than education and can occur outside educational settings. Learning theories describe ways in which people learn, providing conceptual frameworks for interpreting the learning process (Siemens, 2006). Three major learning theories are behaviourism, cognitivism and constructivism. Behaviourists see learning as a black-box approach (Driscoll, 2000). Behaviourism focuses on observable behaviour, influenced by rewards and punishment as stimuli, and views behavioural changes as a learning result (Venezky and Osin, 1991). Cognitivism relates to cognitive processes such as the formation of mental models, metacognition, and self-regulation. Cognitivists view learning as internal human information processing (Driscoll, 2000). According to De Villiers (2005), constructivism relates to personal knowledge construction and interpretation, active learning, anchored instruction, and multiple perspectives on an issue. The constructivist approach proposes that understanding and learning are constructed by personal reflection (Driscoll, 2000). None of these is exclusively right or wrong, although constructivism is presently accepted as the most relevant, and many current educational policies, models and practices focus primarily on constructivist learning.

Learning, whether deliberate or incidental, cognitive or practical, involves a change in knowledge, skills or attitudes. It occurs by using prior knowledge, conditions, and mental understanding to synthesize the skill or concept being acquired. Learning is flexible, occurring via different routes and learners do not always know exactly where they will end up (Rushton, Eitelgeorge and Zickafoose, 2003; Goffree and Stroomberg, 1989; Jarvis, 2006). Furthermore, learning can be enhanced by the development of specific skills.

Cotton (1995) distinguishes between three types of skills: psycho-motor skills that become automatic after repeated performance, perceptual skills controlled and conducted by the senses, and cognitive skills used in the process of acquiring knowledge. Using software applications is a composite skill that requires all three of these skills.

2.4 Comparison of adult and child learning

Learning in adults and children is qualitatively different, due to the maturation of the brain's learning capacity and to different life situations (Illeris, 2006). In childhood, learning is typically uncensored and trusting. Children develop their thinking abilities by interacting with other children, adults and the physical world. In adulthood, learning is fundamentally selective. Adults concentrate on learning things that concern work, careers, family and interests. Learning is motivated by a need to become more self-directed. Another important difference is that children have not completely developed cognitively, emotionally or physically.

Children learn best when learning is initiated by their inquisitiveness and interests, rather than imposed (Woolley, 1997). They naturally seek to make sense of experiences and to find order and patterns in their environment. They should know which forms of behaviour produce desirable effects and eliminate those that do not (Ramey and Ramey, 2004). Children are keen observers of other people and can mimic behaviour (Thornton, 2002). They learn by peer-interaction, playing or solving problems together. Play provides opportunities for exploration and trial-and-error in safe and enjoyable settings (Ramey and Ramey, 2004). When emotions are engaged, events and ideas are more strongly committed to memory. Children also learn a great deal from involvement with activities conducted by adults or more experienced children (Thornton, 2002).

Adults are more successful at developing a skill when they understand why they should acquire it (Knowles, 1998). An adult's ability and willingness to learn is largely affected by the value they place on the task at hand (Knowles, 1998; Merriam, Caffarella and Baumgartner, 2007). In order to motivate an adult, whatever they learn should be relevant to everyday life. They are driven to learn by the desire to know more, provided that it is clearly relevant or useful. Adults enter learning with a large quantity of experience that varies from individual to individual (Gravett, 2001). In the learning process, they draw on the resources they already have (Illeris, 2006).

As people mature, they become less dependent and more self-directed; thus, they develop a need to be recognized, and treated, by others as self-directed human beings (Merriam et al., 2007). Adults employ self-directed learning, taking responsibility for how they set their learning goals, locate appropriate resources, decide on which learning methods to use, and how they evaluate their progress (Brookfield, 1995). Adults who have decided to learn something, generally wish to see rapid results on their efforts.

According to Brookfield (1995) there is a form of competitiveness in the way adults approach learning. Learning, for them, is linked to quality of life and self-esteem. Many adults judge themselves according to how learned they are.

The differences between the learning of adults and children that are particularly relevant to our research are:

1. Children often depend on adults for material and psychological support during learning, while adults depend upon themselves (Cave, LaMaster and White, 2006).
2. Adults are largely self-directed in that they are responsible for deciding what, when, and how learning will occur. This difference is relative and varies according to context (Leberman et al., 2006).
3. Children perceive learning to be explicitly one of their major roles in life, while adults perceive themselves as doers who use learning to achieve success (Ference and Vockell, 1994). Adults learn best when they perceive the outcomes of the learning process as valuable and contributing to their own development and success.

4. Adults have more life experience than children, which provides a good foundation for learning. However, it can also be a hindrance and a child's lesser experiences can occasionally prove more beneficial. Adults can be less willing to explore new ways of doing things (Durham, 2010).

Having established the theoretical context of the research reported here, we next give a detailed account of how the research was conducted.

3. Research design and methodology

Following Terre Blanche and Kelly's (1999) classification of a research approach, our research can be described as descriptive, basic and qualitative. It is *descriptive* in the sense that it represents a picture of the specific details of a situation, social setting or relationship (Neuman, 2003; Terre Blanche and Kelly, 1999) – in this case the relationship between users and software applications. Rich descriptions of the behaviour of participants at the interface provided the starting point for data analysis. The research is *basic* as it seeks to improve understanding of basic principles – here, the principles of interaction design. Qualitative and quantitative data collection methods were used, but the analysis was mainly *qualitative*. The analysis resulted in the interpretation of observed differences as a series of insights into the different meanings of learnability.

The research was guided by the following questions:

1. Does the learnability of software interfaces have a different meaning for child and adult users?
2. Should the learnability principle be adapted or interpreted differently to be applicable to software applications aimed at users from different age groups?

The participants in this research were children between the ages of 9 and 12 years and adults aged from 35 to 50. By the age of 9, children are able to complete a task by following specific directions and they have an adequate attention span (Hanna, Risden and Alexander, 1997). A total of 28 participants were involved – 12 children and 16 adults. Table 1 shows the participant profile. Eight of the participants were experts in one or more of the software applications and twenty were novices. Five of the participants acted as both experts and novices, as there were software applications with which they were familiar and other applications which they had not used before. The children came from two primary schools in Pretoria, South Africa and the adults were employees from one university in Pretoria, South Africa. All participants were volunteers who responded to an invitation to participate.

Table 1 Participant profile

	Age range	Gender	Computer experience
Children	9 to 12	5 Male 7 Female	1 Low 4 Moderate 6 Moderately high 1 High
Adults	35 to 50	8 Male 8 Female	2 Low 2 Moderate 7 Moderately high 5 High

3.1 Software used in this research

The four software applications used in our experiments are described below:

1. Timez Attack is an educational software application, cleverly disguised as a captivating computer game. Timez Attack teaches the multiplication tables. Users navigate an avatar (a little green alien) through dungeons in search of golden keys to open doors. The keys take the form of multiplication problems. When a key is found, the program takes the user through a sequence of events that helps to systematically build up the answer to the problem.
2. Storybook Weaver Deluxe 2004 is a software application for creating stories. Users can choose from a large selection of backgrounds to create scenes on the pages of their electronic storybook. They can also select from thousands of story characters and objects to create illustrations.
3. Microsoft Outlook manages e-mail, calendars, contacts, tasks, to-do lists and documents or files on the hard drive.
4. Microsoft PowerPoint is a software product used to create computer-based presentations typically used when presenting a conference paper, teaching a class, introducing a product to sell, or explaining an organisational structure.

3.2 Experimental procedure (data collection)

A series of experiments was conducted in a formal usability laboratory. Each experiment involved a user learning a new application, either on his or her own, or with the help of an expert. The following eight combinations of participants were observed:

1. A child learning a new child product with the help of a child expert.

2. An adult learning a new child product with the help of a child expert.

3. An adult learning a new adult product with the help of an adult expert.

4. A child learning a new adult product with the help of an adult expert.

5. An adult learning a new adult product on his/her own.

6. An adult learning a new child product on his/her own.

7. A child learning a new adult product on his/her own.

8. A child learning a new child product on his/her own.

Data was collected through observation, eye tracking, video recordings and interviews. Observations took place in the usability laboratory where all sessions were video recorded. The advantage of video recording is permanence as it allows us to experience an event repeatedly by re-viewing it (Grimshaw, 1982). With each repeated viewing, we could adjust our focus somewhat and notice behaviours that were overlooked during previous viewings (Erickson, 1982; Fetterman, 1998). Replaying the recordings also allowed the researchers more time to contemplate the data before drawing conclusions. The repeated re-viewings were accompanied by extensive note taking. This yielded a rich set of data that was analysed qualitatively.

Usability testing is a usability evaluation method that measures performance as users do specified tasks on a target system. It is conducted in specialised usability laboratories with one-way glass, so that researchers can observe participants' interactions with a system, without participants being aware of them. The sessions are video-recorded so that researchers can re-view them.

Eye tracking was another method used to collect data while single participants taught themselves to use software applications. Eye tracking is a technique used to determine eye movement and eye-fixation patterns. It is the process of measuring the point of gaze, where the subject is looking, or the motion of an eye relative to the user's head (Pretorius, Calitz and van Greunen, 2005). In this research we used a Tobii 1720 eye tracker, which was installed in the usability laboratory.

Informal, semi-structured interviews were held at the end of each session. The purpose of these short interviews with both the novice and expert participants after the experiment, was to discuss issues concerning:

- how they felt during the experiment when they were observed and recorded,
- whether the software was easy or difficult to learn,
- parts of the software applications they enjoyed and those they did not enjoy, and
- whether they were able to master all the required activities.

3.3 Data analysis

In analysing the observation and interview data, we followed the five step process proposed by Terre Blanche and Kelly (1999):

1. Familiarisation and immersion: This step involved immersing ourselves in the material gathered, by reading it repeatedly, taking notes and drawing diagrams, with the objective of obtaining complete comprehension of the collected data. The outcome was textual descriptions of the specific observed behaviours that relate to learnability of the software interfaces.
2. Inducing themes: Induction is the inference of general rules or classes from specific cases (Terre Blanche and Kelly, 1999). Through this process we identified themes that emerged from the data, using a bottom-up approach. These themes formed the basis from which descriptions of the observations were refined and reorganised. Examples of themes identified are '[i]nterface aspects that adults struggle with but children do not' and 'differences in reactions to usability problems'.
3. Coding: In our research, coding overlapped with the process of inducing themes. It involved ongoing dissection of the data and linking pieces of information to the identified themes. During this process themes were sometimes expanded or combined in the light of new discoveries.
4. Elaboration: Elaboration involved exploring the newly organised material to identify similarities and differences in the data that may lead to new insights. In this case, insights related to the difference between adults' and children's ways of learning a new software application. We reorganised the accrued material resulting from analysis and

coding into a coherent discussion of insights that would inform the reformulation of learnability.

5. Interpretation and checking: In this phase, the final account of the study was compiled. It was necessary to ensure that no weak points, contradictions or gaps existed in the proposed framework.

The result of the thorough data analysis process was a list of insights into aspects of interaction with unfamiliar software interfaces that reveal differences in how children and adults approach these applications. Patterns that emerged from eye tracking studies were analysed by comparing the fixation paths and gaze plots of children and adults. These patterns served as further evidence for some of the insights that emerged from the analysis process described above. We elaborate on this in the subsequent discussion of the research results.

The next section describes these insights. The insights led us to the final outcome of the study, namely a reformulated definition of learnability.

4. Research results

Ten insights emerged regarding differences between adult and child participants' experience at the interface. This section is structured around these insights. Each insight is discussed with its justifying evidence, as well as the implications it has for software design.

Insight 1: Children are more accepting and accommodating of usability problems than adults.

We saw evidence of this in both the educational games, where there were clear usability problems. For example, in Timez Attack, a congratulatory message incorrectly appears on the screen before the player had actually achieved any milestone in the game. All the children ignored the message. The adults were taken aback or questioned why they were being congratulated before completing a task. In Storybook Weaver the buttons for selecting and adding objects to a story page do not follow general interface conventions. Children, however, were unperturbed when they had to select and add an object by clicking on the + button instead of the expected ✓ button, as shown in Figure 1 below.

.

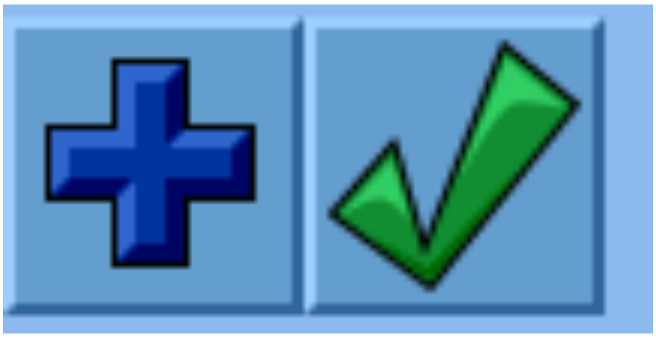


Figure 1 Buttons for selecting and adding objects

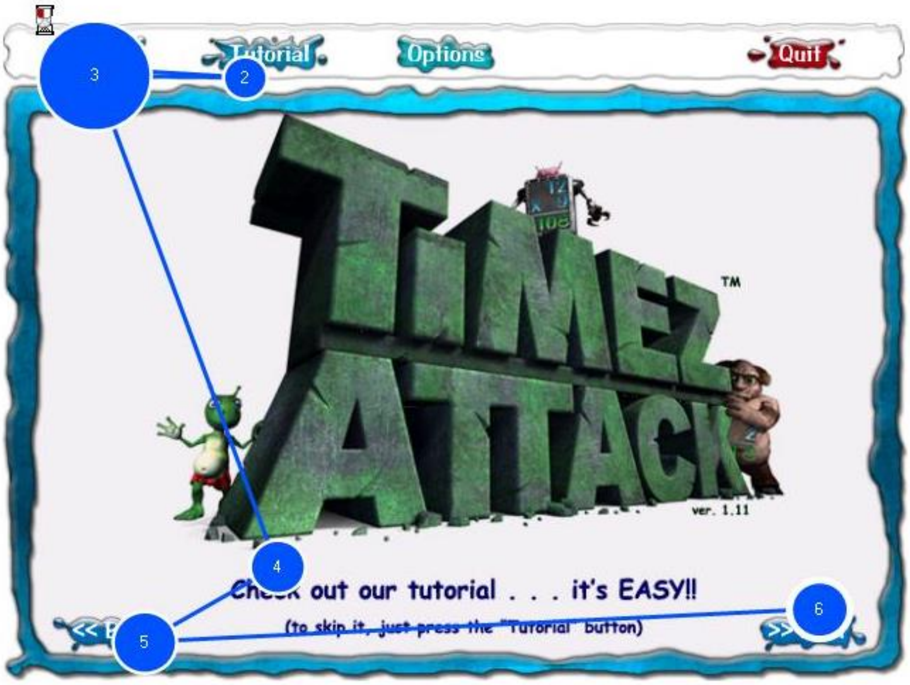
Children clicked on both buttons in an attempt to figure out the right one. Adults, on the other hand, generally appeared irritated when they did not get the desired outcome by clicking a tick (✓). Some of the comments from adults were:

> *'I am completely baffled and whatever this is, it is non-intuitive….*
>
> *'Oh Good Grief! What must I do with this thing?'*
>
> *'What must I do now?'*

We learn from this that designers should not assume that a child's impression of usability is valid. They can be very forgiving and may not comment on obvious problems when asked their opinions. It may be a good idea for designers to use adults to test products intended for children in order to discover potential usability problems that may be overlooked when testing with children.

Insight 2: Children and adults use different learning techniques when playing an unfamiliar game. Adults rely on instructions and are more systematic than children in how they approach learning a new application. Children are more willing than adults to use trial-and-error to achieve an objective.

Children displayed fearlessness when confronted with a new application. They just wanted to get on with it, while most adults first consulted the instructions or tutorials. Eye tracking results revealed that when the application was a game, children immediately searched for the button that would activate the game. In Timez Attack, for example, the longest fixations of child novices were on the Play button at top left on the screen (see Figure 2). Adult participants, by contrast, fixated on the instructions at the bottom of the screen. This particular adult did not look at the Play top left button at all, as shown in Figure 3.



Che out our tutorial . . . it's EASY!!

(to skip it, just press the "Tutorial" button)

Figure 2 Fixations of a child novice on the opening screen

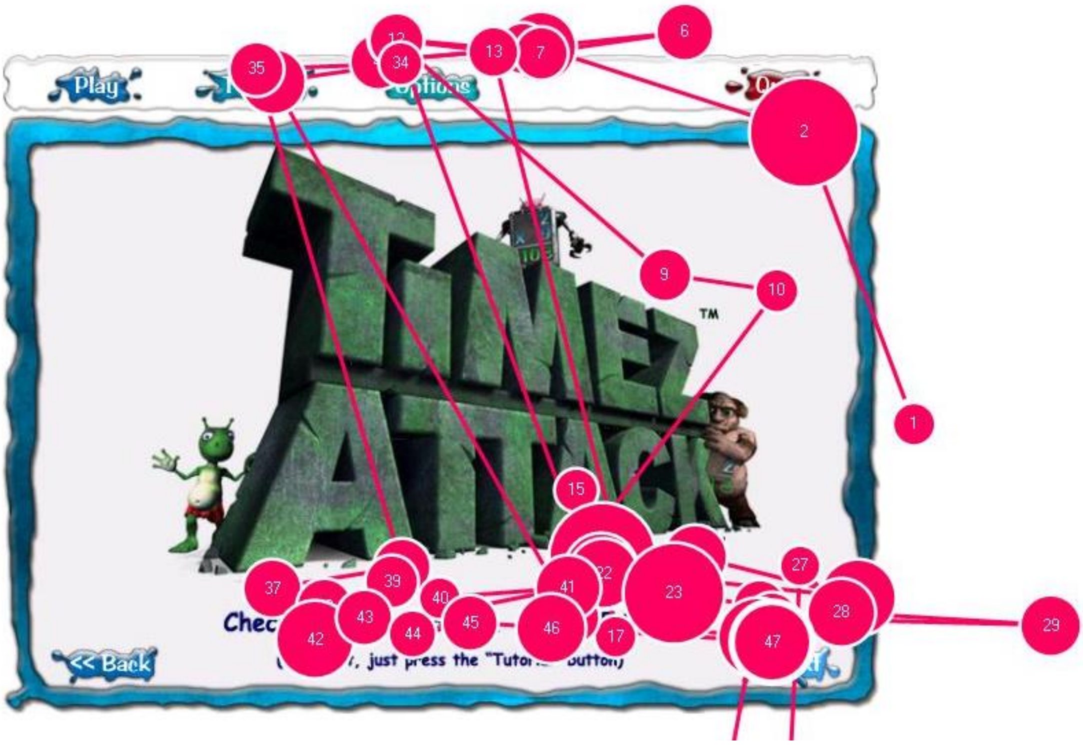


Figure 3 Fixations of an adult novice on the opening screen

Further eye tracking results, as well as results from the observations, confirmed this. Adults had considerably more fixations on the instructions during the first 20 seconds of playing Timez Attack, as previously mentioned, and as indicated by the concentration of fixations at the bottom of Figure 3 in the area where the instructions were located. The average number of fixations on instructions during this 20 second period was 9.75 for children and 33.25 for adults. Children ignored most of the on-screen instructions and seldom asked the available expert for help. Adults were much more prone to seeking guidance.

We learn from this that designers of software intended for children should not rely on written instructions to inform children how to use the software. They should use elements of predictability to make the software easy to use without instructional guidance. Children learn through trial-and-error and are not afraid to experiment. This means that children's products must be more robust and fault tolerant.

Designers should also make the instructions clearly detectable and well positioned on the screen, so that users (especially adults) can easily find and read them.

Insight 3: The life experiences of adults are much greater and more ingrained than those of children. Adults tend to have fixed patterns as a result of their life experiences and can be less open-minded during learning than children.

When playing a game that involved a combination of keyboard and mouse navigation, adult novices had difficulty, because they were accustomed to using the mouse for clicking buttons on menus rather than moving objects from one place to the other. They struggled to adjust to using the mouse for anything else (e.g. as a steering device). Moreover, when adults had already attached a meaning to an interface element, it was more difficult for them to accept a different use of the element, as in the case of the + and ✓ buttons discussed in Insight 1.

This insight was also supported by the way adult experts helped the novices, saying things that refer to prior life experiences, for example: "An e-mail carries the same principle as ... writing and posting a letter via the post office. The only difference is that this is done electronically".

We learn from this that software designers should thus design applications to accommodate the different learning styles of different age groups. There is a need to accommodate the more flexible approach of children, allowing them to achieve their goals in different ways. Designers should accommodate the varying life experiences of users and encourage all age groups to explore and discover other ways of doing things, but they should bear in mind that adults may want to follow a path that corresponds to their deep-rooted ways of doing things.

Insight 4: Children learn to use a new software application more independently than adults.

Child novices rarely asked for assistance. They discovered things independently without any need for guidance. Their natural curiosity drove them to make new discoveries. They were also quick learners. If they were not sure what to do they just clicked randomly and went ahead. Adult novices were hesitant to try anything independently and frequently asked for help. This included asking questions such as "What must I do now?"; "What do I write now?"; "What am I going to say?"; "Do I still not give the answer?"; "How did you throw the creatures?"; "When I choose the font size, how do I know that I have chosen the right font size?", and "Can we do it together again, then l can later do it on my own".

Software intended for children should facilitate independent learning. Software designed for adults, on the other hand, should provide detailed instructions on use of the application or should provide easy access to a "help" facility.

Insight 5: Child novices are faster than adult novices in gaining and mastering mouse and keyboard navigation skills during game play.

As mentioned earlier, in Timez Attack, all the adults struggled to use the mouse/keyboard combination to navigate the avatar through the dungeons. Examples of the emotional expressions of adult novices in reaction to mouse/keyboard navigation difficulties included: "Ooooohmmmm!! What now?"; "Oh man, what is the mouse doing?"; "Oh Good Grief! What must l do with this thing?"; and "I give up!!".

Out of the seven child novices who learnt how to play Timez Attack, only one experienced difficulty navigating characters with the mouse during the initial stages of the game, but improved at the end.

Designers of software applications should match the user interface controls to the ages of the users. For example, if adults find it difficult to coordinate the simultaneous operation of the mouse and the keyboard, software designers should provide additional support in assisting adults lerning to navigate. Furthermore, designers should not assume that because adults generally have better hand-eye coordination than children, that they will be better than (or just as good as) children at navigation.

Insight 6: Adults want to have a clear picture of the entire software application before they start using the application, whereas children just start using the application.

This is evident by the systematic way in which adult experts taught novices. The adult expert would begin by asking the novice if they knew anything about the application. Depending on the outcome the expert would then provide an overview of the purpose and working of the complete product. Only then would the expert start giving instructions for using the application and demonstrating it practically. Whilst demonstrating, the expert would ask the novice questions to check whether they understood the demonstration.

Child experts did not give an overview before instructing novices on how to use the software applications for the first time. The child expert allowed the novice to learn the applications by participating from the start. All the adult participants began by reading the tutorials of Timez Attack and StoryBook Weaver, while only one child participant read the Timez Attack tutorials. This confirms that adults want a clear and comprehensive picture of what is coming.

If a product is designed for use by both children and adults, designers should provide customisable guided tours of the application in order to accommodate the requirements of both types of users. If there is a chance that adults will use a product, tutorials that give a product overview should be provided, as adults prefer to read instructions before using the application for the first time. Children should have the option to omit instructions unless it is absolutely necessary for them to take note of them.

Insight 7: Children do not expect things to go wrong. Adults are more cautious, and tend to be more self-critical than children.

Evidence of the above insight is derived from the questions asked by the children. For example, “Does it matter which numbers I use on the keyboard?”; “Can I go the wrong way and see what happens?” and “I want to see what happens if I do not catch one of the snails?”

Adult novices hesitated to make any moves that they were unsure about. With all the applications demonstrated, adult novices often asked for reassurance. For example, “What if I click the wrong door?”; “How do I make the avatar walk to the left without making a mistake?”; “What if I make a mistake and fall in a dungeon?” This means that applications designed for children should show greater tolerance for incorrect operation than applications designed for adults. Designers may work through a risk assessment to ensure the applications and their implementations do not expose children to unacceptable risks. If an application requires a very specific sequence of actions, the interface should make this clear so that children do not follow the wrong trail.

Insight 8: When learning to use a new software application, children are more comfortable learning from their peers than from adults. Adults, on the other hand, are not affected by the age of the person who teaches them, as long as the person demonstrates clear knowledge of what he/she is doing.

Child novices appeared very relaxed when they were being taught by a child expert. They would frequently communicate with the child expert during a demonstration. When an adult expert was demonstrating, the children showed signs of intimidation by failing to ask questions as they did when they were being taught by a child expert. This was confirmed during the interviews.

When developing training materials for software applications, designers should be aware of the effectiveness of peer tutoring for children. For example, where voice-overs are used in automated demonstrations of software designed for children, it would be advisable for software application designers to use a child's voice.

When applications are developed for use in a school laboratory or in any situation where adult supervision will be present, the design should not presume that an adult will provide help when required. When struggling, the children may not ask for help spontaneously. They may rather try to get by on their own or ask their peers.

Insight 9: Children appreciate interactive demonstrations, especially if the expert teaching them is a child, as opposed to one-way demonstrations with which adults seem comfortable.

In the child-to-child demonstrations, novices were engaged from start to finish. Child experts let the child novices interact and participate in decisions right from the start of a session. Adult experts would give an overview of the application before starting a demonstration, and tended to explain their own choices in the beginning rather than letting the novice participate in the choices.

Designers are encouraged to develop instructional demonstrations that take full advantage of the needs and preferences of a particular user group. Where voice-overs are used in demonstrations, children appreciate listening to another child's voice, in line with this user group's preference for peer tutoring. Designers are also encouraged to embrace elements of interactivity that allow children to actively take part in the demonstration and to control the pace and progress of the demonstration.

Insight 10: Children accept what they are learning regardless of its purpose. Adults find learning to be purposeful if it has meaning and adds value to their lives.

Children and adults get more involved if they are using software that relates to them. Adults connect their learning of new software applications to life experiences that may include work-related activities, family responsibilities and even previous educational experiences.

Adult experts communicated a great deal of information on software applications such as Outlook and PowerPoint, because they were more well-versed with this software than with educational games such as Timez Attack and StoryBook Weaver. Adult novices showed great enthusiasm learning the former applications, since they knew that these would help them in their day-to-day work. Child novices enjoyed the experience of using all of the software applications that they were taught. Adult novices were more eager to learn PowerPoint presentations and Outlook than they were to learn educational games. Whilst the experts were demonstrating the two games, child novices paid intense attention to how the game was played. Adults showed more emotional engagement than children during their encounters with the software applications. Adults became easily frustrated when they were using an application that was not meaningful to them.

Application designers should include elements of user interface design that reinforce positive emotions in the user.

To conclude this discussion on the insights, we highlight certain findings. Adult learning is value-driven. Adult learners need to know why they should learn something before undertaking to learn it. Given the rationale for learning something, they will often invest considerable energy in investigating the increased benefits to be gained from the learning experience and the consequences of not learning it. Adults appreciate learning concepts that are relevant to their everday life and that they can use afterwards. Designers of software applications for adults should therefore make the value of learning a new application clearly apparent.

Our results thus show that the differences in how adults and children learn in general do not always apply when they are learning to use a new software application:

- Whereas adults usually rely on themselves in the learning process, when they learned to use the software games in this research, they depended more on the support of instructions and external assistance than children did.
- Adults were not noticeably more self-directed in learning these games than children.
- Adults' broader life experience did not have a clear effect on their learning of the games. In some instances it hindered, rather than helped, them. Children were more confident to learn through trial-and-error, while adults preferred to read instructions.

Section 2.3 introduced three major current learning theories. The way in which the adults in the study approached the teaching and learning of software applications indicated that they applied principles of cognitive learning. They developed mental models to guide their actions and used a self-regulated, systematic and holistic style as they sought fixed interaction patterns. Children, by contrast, demonstrated constructivist learning as they taught and learned independently, actively, and interactively. They obtained their knowledge by personal exploration and construction rather than from an overview and instructions.

The insights on the differences between the learning styles of children and adults serve as basis for our proposal in the next section regarding the definition of the learnability principle.

5. Implications for interpretation of the learnability principle

The research question addressed in this article relates to the possibility that learnability has different meanings for users of different ages. In the discussion that follows, we relate the sub-principles of learnability to the insights gained through this research, making suggestions for changes to accommodate users of different age groups. We end this section with the proposal of an additional sub-principle, namely, 'engageability'. In our discussion of the engageability sub-principle we do not only provide support from our results – we also consult literature on the role of engagement in human-computer interaction as well as in learning to strengthen our case for including it in the definition of learnability. The way in which the subprinciples of learnability (Dix et al., 2004) are represented in our insights is now discussed in detail.

5.1 Predictability

Predictability allows users to know beforehand what will happen when they click on a menu item or press a key. Insight 1 shows that children do not consider the meanings of interface elements as critically as adults. They merely try out options until one works. Adults, on the other hand, experienced confusion if a button or other function did not operate as they had expected.

Elements relating to predictability were also evident from Insight 2. Children used trial-and-error in learning, whilst adults read instructions first. This implies that children do not feel the need to predict the next step in interaction. However, predictability is also important for children because, if they use trial-and-error and do not succeed, they may lose interest. Adults are more systematic than children in the way they learn to use a new software application. Therefore user interface designers should be particularly aware of elements of predictability in user interfaces of applications designed for adults.

Insight 3 also refers to elements of predictability with regard to the way that adult novices use prior experience to predict future actions, whilst child novices are not necessarily guided by previous experiences.

5.2 Synthesisability

Synthesisability is the ability of the user interface to allow the user to construct a predictive mental model of how it operates (Hinze-Hoare, 2007). A synthesisable interface allows the user to understand which actions have led to the current state, what the system did to get there, and what the user should expect next.

Insight 3 relates to synthesisability. It refers to the life experiences of adult novices that are greater and more diverse than those of children and that allow them to develop fixed life patterns or mental models that may enable them to overcome the difficulties of learning to use a new software application. However, at other times, these fixed mental models may be detrimental to learning an unfamiliar software application. Fixed patterns as a result of prior experiences can make them less open-minded than children to new learning.

Insight 2 also relates to synthesisability in the sense that child novices, without much life experience to guide them in unfamiliar circumstances, work in a trial-and-error manner. They use what they learn in this way to construct cognitive maps of the operation of unfamiliar software.

5.3 Familiarity

Familiarity is the degree to which the user's experience and knowledge can be drawn upon to understand the operation of an unfamiliar system. Familiarity impacts on the novice user's initial perception of the system and on whether users can determine required actions from their own prior experience. When the new system is familiar, the user will relate it to similar systems or real-world situations, thereby reducing the cognitive burden involved in becoming adept in its use.

The link between Insight 3 and familiarity is obvious and the implication is that familiarity is more important for adult users than for children. Insight 10 also relates to familiarity, indicating how both child and adult novices get involved in learning a new software application if it relates to some tasks or activities that they encounter in real life. Users with previous experience of software applications in general will experience similar functionality while using the new software and will find it easier to learn than users with no prior experience. The experiments showed that children mastered games faster than adults, probably because they had played such games or watched them being played.

5.4 Generalisability

An interface is generalisable if users are able to use what they have already learnt to carry out new tasks. Insight 5 relates to generalisability, suggesting that child novices were faster than adult novices in mastering mouse and keyboard navigation skills. Besides youthful dexterity, as opposed to the general slowdown of motor co-ordination with age, the main source of performance advantage in mouse and keyboard mastery by child novices over adult novices could be found in the generalisability of mouse and keyboard skills mastered in the prior use of applications. User interface standards and guidelines promote generalisability. A user who knows how to use specific icons in one application should be able to apply this knowledge in a different application. Insight 1 exemplifies this point.

5.5 Consistency

Kristoffersen (2008) states that generalisability is sometimes described as a form of consistency, except that it applies more broadly to situations, rather than just operations. Insights 2 and 7 support Nielsen's (1994) assertion that consistent user interfaces encourage exploratory learning strategies. The propensity of child novices to use the trial-and-error approach to learning a new software application relates to the principle of consistency in the sense that consistent system feedback and responses to user action allows the child novice to learn how to play in this way. If system feedback and responses are consistent, child novices will have more success in their exploratory approach to learning.

6. Reformulation of the learnability principle

It is evident from the above discussion of the findings of this study that the sub-principles of learnability can be applied differently for children and adults. To summarise:

Predictability is more crucial in adult products than in those aimed at age group 9 to 12. Adults prefer a predictable system, since they like to be sure of what will happen next and what they are able to do next, while children are more willing to try out different actions and just explore. Children are less concerned about the effects of their actions than adults.

Synthesisability has a different meaning for the two user groups since their differing levels of experience influence the way they form mental models about the operation of a system. When designing for children, designers should thus be aware that children may construct different mental models from what they (the adult designers) would expect. A system that adheres to the principle of familiarity for adults may include elements with which children are not familiar. On the other hand, the fact that children are exposed to technology from early on, may mean that new input mechanisms that they have been exposed to through computer games may be unfamiliar to older people. The consequences of generalisability and consistency on learnability may be different for each user group, but there is no indication from the derived insights that the two user groups understand the meanings of the two principles differently.

The major contribution of this research is a suggested extension to the learnability principle. Below, we present engageability as an additional sub-principle.

6.1 Engageability

Engagement has been described from different perspectives: in the HCI context, engagement is a goal of interface design; it is also the main focus in the well-established frameworks of flow theory and play theory (O'Brien and Toms, 2008). Engageability occurs when a person loses himself or herself in an activity, losing all track of time and not noticing anything outside of the activity. Engageability is thus associated with flow (Csikszentmihalyi, 1990), which occurs when individuals are so engrossed in an activity that they do not want to stop. Furthermore, it is related to intrinsic motivation, which is motivation that is internal to a learner. In contrast with extrinsic motivation, such as marks and rewards, intrinsic motivation is inherent in the learning process, and occurs when learning is found to be fun (Alessi and Trollip, 2001).

In HCI, engagement has been referred to in a number of different ways: "as a process; a stage in a process, or the overall process; as an experience; as a cognitive state of mind ....." (Peters, Castellano and Freitas, 2009:1). Van Greunen, Van der Merwe and Kotze (2010) view engagement as a user experience goal that is required for positive interaction with a system. Engaging the user at the interface has been the topic of HCI research, and engagement is also a familiar concept in learning. In as much as researchers have written on engagement in learning, the ability to engage – or 'engageability' – has never been explicitly considered as a learnability principle in HCI.

We define engageability as the extent to which a software application can fully engage the user by providing a complete and satisfying user experience. Depending on relevance and context of use, a user interface that adheres to the engageability principle will enable users to be self-regulated, to define their own learning goals and to evaluate their own achievements. An engaging software application will naturally support users in knowing how to learn and transfer knowledge. Engageability also promotes collaborative use, supporting the possibility of users sharing the experience.

Support for engageability can be found in a number of insights discussed in Section 4. Insight 9, for example, relates to children appreciating interactive demonstrations, as opposed to the more direct demonstrations that adults appreciate. To make an application engaging for a child, they should not be expected to passively watch a demonstration before they can start using the system. Adults, on the other hand, may lose interest if they are unable to form a clear and holistic view of the system before they start to use it (Insight 6). Insight 10 is about children and adults engaging with the system and becoming more engaged and involved as they use software that relates to them.

Adults connect their learning of new software applications to life experiences and knowledge that may include work-related activities, family responsibilities and even previous education. The relevance of a system for the child or adult user respectively is thus an important aspect of engageability. Insight 8 demonstrates that learning with peers makes children more active participants in the learning process – that is, they are more engaged when learning with another child. Insights 2 and 4 imply that if children are forced to rely on instructions or tutorials to use an application they may well lose interest, while the opposite is true for adults. Thus, engageability has very different meanings for children and adults.

Table 2 is a matrix which links the sub-principles and the insights. It indicates relationships between the ten insights and the six sub-principles. Five are the original sub-principles proposed by Dix et al. (2004), while the sixth is the sub-principle of engageability suggested in the present study to describe situations where users are fully engaged in the environment. It is notable in Table 2 that engageability relates to five of the insights, more than the number to which any other sub-principle relates.

Table 2: Matrix of insights and sub-principles

Sub-principle	*Insights*									
	1 **Accepting and accommodating**	2 **Relying on instructions**	3 **Life experiences**	4 **Children more independent than adults**	5 **Gaining and mastering of skill**	6 **Having a clear picture**	7 **Adults are more cautious**	8 **Children learn from peers**	9 **Appreciating interactive demonstrations**	10 **Accepting what is being learnt**
Predictability	√	√	√							
Synthesisability		√	√							
Familiarity			√							√
Generalisability					√					
Consistency		√					√			
Engageability		√		√		√		√	√	

7. Conclusion

The objective of the overarching research – to compare the meaning of learnability for child and adult users – was achieved. The lack of relevant information in the literature dealing specifically with the learnability principle suggested a gap in the body of knowledge. This research was an attempt to fill this gap and improve the granularity in the description of the learnability sub-principles. The comparison of the meaning of the learnability principle for children and adults using an unfamiliar software application yielded significant results for software designers and HCI practictioners in general. The end result ultimately aided in the reformulation of the learnability principle in a way that distinguishes between the needs of adults and children. The way in which learnability and its sub-principles are currently defined, makes it difficult to clearly separate its application to different age groups.

A new sub-principle, engageability, was proposed to incorporate aspects of learnability that were not covered by the existing definition. None of the existing sub-principles captures the importance of the engagement factor in the learnability of software applications. The more users are captivated by an application, the more committed they will be to master it. We have also shown that children and adults are captivated (that is, engaged) by different, and sometimes even opposite, characteristics of a user interface. However, further research is necessary to get further support for this principle, for example, to determine what the impact of context of use would be on its effectiveness or applicability.

We conclude this article on a practical level by providing some recommendations for software design as drawn from the results of this research:

1. Software should be designed so that, even if instructions are available, users can use the software without instructional guidance. Software intended for children should not rely on written instructions only, but should allow for trial-and-error techniques to be followed. Software for adults, on the other hand, should provide detailed instructions and "help" facilities.
2. Designers should make instructions clearly detectable and well-positioned on the user interface so that users are more likely to read them when it is essential for successful interaction. If adults will use the product, tutorials with product overviews should be provided.
3. Designers should consider the coordination of mouse and keyboard use by both children and adults. It can take time to develop an understanding of the connection between moving a mouse and movement on a computer screen. Adults may also require help to master the motor control required when input devices are used for actions to which they are unaccustomed.
4. Applications designed for children should show greater tolerance for incorrect operation than applications designed for adults (that is, there should be flexibility for a trial-and-error approach to learning).
5. Designers of software applications intended for adults should make the value of learning an application apparent. Given the rationale for learning something, adults will be more likely to invest time in it.
6. Different user groups have distinctive skills and abilities, therefore, design principles for one group may not be applicable to the other group. Designers should acknowledge that they, as adults, may not understand the needs of child users. On the other hand, when designing for children, they should test the usability of their designs with adults, because children are more accepting of usability problems.

We believe that the findings and recommendation of this inter-disciplinary research will help to advance HCI practice and to improve the quality and learnability of software targeted at different age groups.

Bibliography

Alessi, S. M. & Trollip, S.R. 2001. *Multimedia for learning: Methods and development*. Boston, MA: Allyn & Bacon.

Aspinall, D. 2007. *Principles and guidelines interface design rules: Informatics.* Edinburgh: University of Edinburgh. Available from http://homepages.inf.ed.ac.uk/da/teach/HCI/rules.pdf [Accessed: 20 February 2009].

Baumgarten, M. 2003. Kids and the Internet: A developmental summary. *ACM Computers in Entertainment,* 1(1): 1-10.

Brookfield, S. 1995. Adult learning: An overview. *International Encyclopedia of Education.* Oxford: Pergamon Press.

Cave, J., LaMaster, C. & White, S. 2006. *Adult characteristics*. Available from < http://ed.fnal.gov/lincon/staff_adult.shtml> [Accessed: 15 May 2009].

Chiasson, S. & Gutwin, C. 2005. *Design principles for children's software*. Technical Report. HCI-TR-05-02, Computer Science Department, University of Saskatchewan.

Cotton, J. 1995. *The theory of learning: An introduction.* London: Kogan Page.

Csikszentmihalyi, M. 1990. *Flow: The psychology of optimal experience*. New York: Harper and Row.

De Villiers, M. R. 2005. e-Learning artefacts: Are they based on learning theory*? Alternation,* 12(1b): 345-371.

Dix, A., Finlay, J., Abowd, G. D. & Beal, R. 2004. *Human-Computer Interaction*. 3rd ed. New York: Pearson Education.

Driscoll, M. 2000. *Psychology of learning for instruction.* 2nd ed. Needham Heights, MA: Allyn & Bacon.

Durham, J. 2010. *Cognitive differences between adults and children*. Available from: <http://www.exploreadultlearning.co.uk/cognitive-differences-adults-children-learning.html> [Accessed: 10 September 2010].

Erickson, F. 1982. Audiovisual records as a primary data source. *Sociological Methods and Research,* 11(2): 213-232.

Ference, P. & Vockell, E. 1994. Adult learning characteristics and effective software instruction. *Educational Technology*, 34(6): 25-31.

Fetterman, D. M. 1998. *Ethnography: Step by step*. 2nd ed. Thousand Oaks, CA: Sage.

Fishel, C. 2001. *Designing for children*. Gloucester: Rockport Publishers, Inc.

Gravett, S. 2001. *Adult learning: Designing and implementing learning events; A dialogic approach,* Pretoria: Van Schaik Publishers.

Gelderblom, J. H. 2008. *Designing technology for young children: Guidelines grounded in a literature investigation on child development and children's technology.* PhD thesis, University of South Africa.

Goffree, F. & Stroomberg, H. (eds.) 1989. *Creating adult learning, theoretical and practical communications on educational design for adults*. Leyden: Spruyt, Van Mantgem & De Does.

Grimshaw, A. D. 1982. Sound-image data records for research on social interaction: Some questions and answers. *Sociological Methods and Research,* 11(2): 121-144.

Hanna, L., Risden, K. & Alexander, K. J. 1997. Guidelines for usability testing with children. *Interactions*, 4(5): 9-14.

Hinze-Hoare, V. 2007. *Review and analysis of Human Computer Interaction (HCI) principles*. Available from: < http://arxiv.org/ftp/arxiv/papers/0707/0707.3638.pdf> [Accessed: 10 August 2010].

Illeris, K. 2006. *How I learn: learning and non-learning in school and beyond.* Roskilde University Press.

Jarvis, P. 2006. *Towards a comprehensive theory of human learning,* London Routledge.

Knowles, M. 1998. *The adult learner: The definitive classic in adult education and human resource development*. Boston, MA: Gulf Publishing Company.

Kotzé, P. & Johnson, C. 2004. *Human Computer Interaction*: Study guide for INF120-8, Pretoria: UNISA Press.

Kristoffersen, S. 2008. Learnability and robustness of user interfaces: Towards a formal analysis of usability design principles, in *Proceedings of the ICSOFT 2008: 3rd International Conference on Software and Data Technologies.* Volume SE/GSDCA/MUSE: Institute for Systems and Technologies of Information, Control and Communication, 2008: 261-268.

Leberman, S., McDonald, L. & Doyle, S. 2006. *The transfer of learning. Participants' perspectives of adult education and training,* Aldershot: Gower Publishing Ltd.

Merriam, S. B., Caffarella, R. S. & Baumgartner, L. M. 2007. *Learning in adulthood: A comprehensive guide*. 3rd ed. San Francisco: Jossey-Bass.

Neuman, W. L. 2003. *Social research methods: Qualitative and quantitative approaches.* 5th ed. Allyn & Bacon: VIII, 608.

Nielsen, J. 1994. *Usability engineering*. Boston, MA: Academic Press.

O'Brien, H. & Toms, E. 2008. What is user engagement? A conceptual framework for defining user engagement with technology. Journal of the American Society for *Information Science & Technology*, 59(6): 938-955.

Peters, C., Castellano, G. & De Freitas, S. 2009. An Exploration of User Engagement in HCI, *Proc. of the Affective-Aware Virtual Agents and Social Robots (AFFINE) Workshop*, *International Conference on Multimodal Interfaces* (ICMI'09), ACM Press, Article No. 9, (2009). ISBN: 978-1-60558-692-2.

Preece, J., Rogers, Y. & Sharp, H. 2007. *Interaction design: Beyond human-computer interaction*. 2nd ed. New York: John Wiley & Sons.

Pretorius, M., Calitz, A. & Van Greunen, D. 2005. The added value of eye tracking in the usability evaluation of a network management tool. In: Bishop, J. & Kourie, D. (eds.). *Research for a Changing World: Proceedings of SAICSIT 2005: 1-10.* South Africa: White River. ACM International Conference Proceedings Series.

Ramey, C. T. & Ramey, S. L. 2004. *How children learn and how parents can help: Center for Development and Learning.* Covington, LA. Available from <http://www.cdl.org/resource-library/pdf/how-children-learn.pdf> [Assessed: 3 July 2009].

Rushton, S., Eitelgeorge, J. & Zickafoose, R. 2003. Connecting Brian Cambourne's conditions of learning theory to brain/mind principles: Implications for early childhood educators. *Early Childhood Education Journal,* 31(1): 11-21.

Santos, P. J. & Badre, A. N. 1995. *Discount learnability evaluation*. GVU Technical Report GIT-GVU-95-30. Georgia Institute of Technology.

Senapathi, M. 2005. A framework for the evaluation of CASE tool: Learnability in educational environments. *Journal of Information Technology Education*, 4: 61-84.

Siemens, G. 2006. 'Connectivism: *Learning theory or pasttime for the self-amused.*' Available from: < http://davidtjones.wordpress.com/2009/10/04/learning-theories-and-e-learning> [Accessed: 18 June 2010].

Shneiderman, B. 1997. *Designing the user interface: Strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co.

Terre Blanche, M. & Kelly, K. 1999. *Interpretive methods*. In Terre Blanche, M. and Durrheim, K (eds.). Research in practice: Applied methods for the social sciences. Cape Town: UCT Press.

Thornton, S. 2002. *Growing minds*. London: Harvard University Press.

Van Greunen, D., Van Der Merwe, A. & Kotze, P. 2010. Factors influencing BPM tools: The influence on user experience and user interfaces. *International Journal of Computing and ICT Research*, 4(1): 47-57.

Venezky, R. & Osin, L. 1991. The intelligent design of computer-assisted instruction. New York: Longman.

Woolley, J. 1997. Thinking about fantasy: Are children fundamentally different thinkers and believers from adults? *Child Development,* 68(6): 991-1011.