Design Principles for Children's Technology

Sonia Chiasson and Carl Gutwin
Department of Computer Science, University of Saskatchewan

HCI-TR-2005-02

ABSTRACT

Designers of children's technology and software face distinctive challenges. Many design principles used for adult interfaces cannot be applied to children's products because the needs, skills, and expectations of this user population are drastically different than those of adults. In recent years, designers have started developing design principles for children, but this work has not been collected in one place. This paper takes a first step towards this goal: based on an analysis of a wide range of research into children's technology, we present a catalogue of design principles for children's technology that are oriented towards the needs of designers.

Keywords

Children's technology, design principles, interface design

INTRODUCTION

Children now represent an important user group for software and technology, and as a result, more attention is being paid to the specifics of how to design for children. This user group is unique in several ways: Their goals while using computers are typically education or entertainment rather than productivity; they have a wide range of skills and abilities; and their experience with computers begins early and continues through their lives. Children are not miniature adults [5], and design principles formulated with adults in mind cannot simply be scaled down – children have their own needs and goals which cannot necessarily be met by adult tools.

These differences lead to the need for different designs. For example, in products whose aim is education or entertainment, user motivation and engagement are as important as task efficiency. Value is only attained if users spend time with the product and it keeps their attention. Designers of children's technology must concentrate not only on the mechanics of their interfaces but also on features that will keep children engaged. Similarly, where products aimed at adults typically make assumptions about their users – like the ability to read, type, and understand abstract concepts – products created for children must carefully take into account their intended audience, since a preschooler who uses a computer will have vastly different skills, abilities, and expectations than will a ten year-old. A child's stage of development will dictate what can reasonably be expected from them in terms of interaction. Any successful product will need to be adapted to the particular needs of its users, including modes of communication, input methods, tasks, and appearance.

Early designers assumed that taking an interface that works for adults, adding a few animations and bright colours, automatically made it appropriate for children [6], and children were rarely involved in the development of products for kids. More recently, however, designers have recognized the differences in this user group, and have worked to give children a voice in the design process. As a result, there are now a considerable number of design guidelines specifically for children.

However, these principles have often been difficult to find. Research into design for children crosses over into several disciplines; human-computer interaction, education, and psychology researchers have all made significant contributions in the area. This distribution of information means that it is difficult to find relevant knowledge when one sets out to design a new product. The problem is compounded by the fact that some well-accepted interface guidelines must be 'unlearned' or adapted for children, but designers may not think to question their assumptions when creating a new product.

To be useful to designers, information from these disparate sources must be brought together. This paper takes a first step to solving this problem: it collects and organizes design principles and insights, gathered from several sources, into an initial catalogue of design principles for children's technology.

ORGANIZATION OF THE CATALOGUE

Children's development can be categorized into three main areas: cognitive, physical, and social/emotional. Cognitive development addresses the mental and intellectual growth of a child. Physical development deals with the development of fine and gross motor skills as well as coordination. Social and emotional development are closely tied: social development involves the formation of relationships with others, and emotional development refers to a child's ability to understand, regulate, and express their own feelings as well as capacity for empathy and compassion. In order to properly meet children's needs and expectations, children's technology must take into account and support these development areas.

The design principles collected below recognize different levels of cognitive development through different literacy levels, different levels of task guidance, and through tailoring to different stages of mental development and imagination. Physical development is reflected in the types of input techniques that can be used and in the use of tangible interfaces. Support for different emotional and social development is seen in the different opportunities to form relationships with others and with the computer. Some of the design principles fall under multiple categories as they address several needs, but we have placed them in the category that best suits their main purpose.

A CATALOGUE OF DESIGN PRINCIPLES

The following sections describe design principles arising from research in HCI, education, and psychology. While many of the design principles can be adapted to meet the needs of children from different age groups, in most cases user testing has been done with a specific age range. Where available, the age range for each research project is included in parentheses.

Cognitive Development

Literacy

Most adult user-interfaces assume that users are proficient readers with fairly extensive vocabularies; most children, however, have not reached this proficiency level. Older children may not fully understand text-based instructions, while young children may not even know the alphabet yet. Conventional interfaces include menus and help functions that are text-based, making them inappropriate for young users. Interfaces that require textual input can also be problematic. Children can be very creative spellers, making it difficult for an interface to recognize text input [7]. Since reading and writing levels vary significantly, children's interfaces must be designed with a narrow age-group in mind to adequately meet the needs of its users.

Steiner and Moher [28] found that graphical metaphors are helpful for children's interfaces. Their graphical interface resembling a storybook helped children (4-7yrs) infer the purpose and operation of their storytelling software. Children could create images in the top portion of the storybook page and write the corresponding story in the lower half. The familiar storybook layout helped children learn to use the software quickly.

More recently, Druin et al. [7] investigated digital libraries for children (5-10yrs) and discovered that typical text-based query interfaces were insufficient for the needs of young users. They developed SearchKids, a graphical search interface that allows querying, browsing, and reviewing of search results through graphics. It uses content-specific metaphors, such as a zoo for navigating information about animals, and allows children to form queries by dragging representative icons. With SearchKids, children could successfully navigate a large information space that was previously inaccessible to them. Their research reinforces the idea that content-specific, graphical metaphors are appropriate representations for children and that visual interfaces with minimal text are most useful.

Hanna et al. [10] present several interface design guidelines for children's technology based on years of experience with developing children's software. To deal with varying literacy levels, they suggest presenting instructions in an age-appropriate format and including the option of having text instructions read aloud, as most children are not accustomed to reading on a screen. They also suggest that instructions should be easy to remember and should avoid making use of concepts unfamiliar to children (for example, referring to left and right portions of the screen for young children). Alternately, on-screen characters can speak instructions with corresponding animations. This method is especially useful because it directs attention and helps in understanding.

The common theme through these projects is that text is not an effective means of conveying information for children. Visual or audio cues prove more valuable, as long as the information presented is clear and age-appropriate.

Interfaces should be strongly visual, avoiding text as much as possible and reducing cognitive load. [7]
Content-specific metaphors are useful in helping children navigate interfaces [7,28]
Instructions should be presented in an age-appropriate format [10]
Instructions should be easy to comprehend and remember [10]

Table 1. Literacy: summary of design principles

Feedback and Guidance

Children expect to see the results of their actions immediately. If nothing happens after their input, children may repeat their action until something does occur (possibly causing a chain of unexpected and unwanted events). Although constant auditory and visual feedback can be annoying for adult users, children often expect it.

Being able to use a system without instruction is also important for children's interfaces. Children cannot be expected to read a manual to learn how to use a product; the product must either be entirely intuitive or provide some form of guidance through tasks.

Children may forget how to accomplish tasks requiring several steps or even simple tasks that are done infrequently. Danesh et al. [4] point to the need for scaffolding, which supports children through the necessary steps. In their development of Geney, they found that children (10-13yrs) would forget how to beam information between handheld devices and needed to be reminded to point the devices towards each other. They used a wizard-like interface to successfully constrain this process.

Hannah et al. [10] add that activities should start out simple, then increase in complexity and difficulty as the child masters the required skills. Feedback is also very important and should guide children through learning new concepts. Sedighian and Klawe [26] found that gradually removing feedback and cues in an educational game encouraged children (12-14yrs) to take on increasing cognitive responsibility, and stimulated engagement with the underlying formal mathematical concepts. The initial levels of their Super Tangrams game provided many visual cues to help predict the outcome of the user's actions; cues were then gradually removed as the children advanced through the game, requiring them to think about the concepts and predict the results themselves.

Steiner and Moher's work on the Graphic StoryWriter [28] demonstrates the need for immediate feedback. Their initial system had no visual or audio feedback to signal that an object was properly selected. As a result, children (4-7yrs) repeated the selection action in hopes that something would happen. They also often clicked on buttons multiple times, leading to unexpected results when the series of commands executed. Subsequent versions which included feedback were much easier for children to use. Said [25] also found that children (9-14 yrs) expect their actions to be immediately reflected on the screen and that feedback should be given in a timely fashion.

On-screen icons need to represent familiar items and be intuitive for children. For example, use a stop sign for stopping activities and make buttons have a 3-D appearance so they appear clickable [10]. Visual or audio feedback should be present when children move their mouse over clickable portions of the screen to indicate what is clickable and what is not. For audio feedback, it is desirable to have a short delay so that children can deliberately activate it; otherwise they tend to hear random audio because their cursor has already moved somewhere else. Feedback should also clearly show when the computer is busy processing requests so that children know to wait for something to happen. This can be achieved through on-screen icons or audio feedback but should be easy for children to understand. Conversely, if the computer has been waiting for input for an extended period of time, it should also indicate this with some feedback (such as humming or toe-tapping) [10].

Children are impatient and need immediate feedback showing that their action have had some effect, otherwise they will repeat the action until some outcome is perceived [25,28]
Interfaces should provide scaffolding and guidance to help children remember how to accomplish tasks [4]
Activities should allow for expanding complexity, and should support children as they move from one level to the next in use of the product [10,26].
Icons should be visually meaningful to children [10]
Rollover audio, animation, and highlighting should be used to indicate where to find functionality [10]
The interface should provide indication of the current state of the system, whether it is busy processing or waiting for input from the user [10]
Interfaces should track and display children's exploration of environments if it is important for them to remember where they have previously visited [31]

Table 2. Feedback and Guidance: summary of principles.

Using a prototype system that simulated a walk through the woods, Strommen [31] discovered that children (6-8yrs) explored interfaces in a non-systematic way and often did not recognize places they had previously visited. Children explored the virtual environment by walking along a set of paths, looking for animals. At the junctions, children rarely looked at all possible paths, nor were they able to distinguish which paths they had previously followed. If it is important for children to track their movement through a virtual environment or for them to explore the environment in its entirety, then the system should provide a means of tracking and displaying this information.

Mental Development

Younger children have difficulty with abstract concepts, and may not have the in-depth content knowledge required for navigating complex interfaces. Their usual approach is trial-and-error: once they find a method that works, they are unlikely to look for a more efficient strategy or for advanced options.

Druin et al.'s [7] work with SearchKids highlights how children (5-10yrs) think of information spaces and how they mentally organize such information. While they may not be able to think of appropriate search terms for a query, or efficiently navigate a categorized structure, they do understand icons representing what animals eat, where they live, and their appearance. By using a visual representation of animal characteristics, children were able to create complex queries and successfully navigate a large information space.

Children's interfaces need to take into account the fact that children may not yet understand abstract concepts [7]
Children's interfaces should not make use of extensive menus and sub-menus as children may not yet have the ability to categorize or have the content knowledge required to navigate efficiently [7]
Children are accustomed to direct manipulation interfaces, their actions should map directly to the actions on the screen. If other styles are used, expect that most users will require training and that some will be unable to grasp how the interaction works [15,21]

Table 3. Mental Development: summary of principles.

Children learn the laws of cause and effect early in childhood. They expect their actions to have a direct effect on their environment. For this reason, input devices should have direct mappings to the actions on the screen. Johnson et al. [15] investigated the use of a stuffed toy as an input device for controlling an on-screen character. Moving the stuffed toy resulted in corresponding actions from the on-screen character. When a direct mapping was used, the children (4+yrs) quickly grasped the idea and had no problems understanding how to control the on-screen character. An alternative approach interpreted the motion of the stuffed toy and mapped this to a context-appropriate action on the screen. While this resulted in smoother on-screen motion, the children needed training to learn how to control the character, and some were unable to grasp the abstraction. Abstractions therefore need to be used with care as they are not intuitive for many younger children.

Plaisant et al. [21] also uncovered the need for direct correspondence between body motion and motion of a controlled robot. In their experiment, children wore various sensor controls, mapping their motion to the actions of a robot. When arbitrary mappings were used, the children found it difficult to control the robot; when they adjusted the location of the sensors so that a more direct mapping applied, the children easily learned to control the robot.

Imagination

Children are good at playing make-believe, and most will readily immerse themselves in pretend situations, acting as if they were presented with the situation in real life [31]. Strommen's [31] "Woods Visit" software enabled children (6-8yrs) to explore a virtual forest from a first-person point of view. He found that the children would physically duck their heads when passing under virtual low-hanging branches. One child even stopped playing and sat quietly in her chair; when questioned, she explained that she was being quiet to see if any animals would come out. Children accept on-screen characters, such as these animals, as social actors and will interact with them in a social manner without hesitation.

When metaphors are used, children expect the on-screen objects to behave as they would in real-life. Rader, Brand, and Clayton [22] explored children's understanding in a visual programming environment. The children (6-11yrs) could create objects then assign rules to control their behaviour. They discovered that while children understood that rules controlled behaviour, they also expected their objects to have the properties of their real-life counterparts. For example, when the object was a raindrop, the children expected that it would fall towards the ground, even though the corresponding rule dictated otherwise.

Danesh et al. [4] successfully used a pond metaphor in their design of Geney, a collaborative game where children (10-13yrs) use handheld computers to learn about genetics. The goal was to mate fish with different characteristics to produce offspring that had a specific genetic makeup. Children easily grasped the pond metaphor, helping them navigate the system and understand the process.

Care should be taken when using metaphors for interfaces as children readily immerse themselves in the environment. While this leads to more intuitive interactions; it may also lead to expectations that exceed the bounds of the interface [4,22,31]

Table 4. Imagination: summary of design principles.

Physical Development

Motor Skills

Children's fine motor control skills develop over time. Until fully developed, they may have difficulty controlling the mouse and targeting small areas on the screen. For example, tasks requiring them to hold down mouse buttons while moving the mouse or requiring them to hold down mouse buttons for extended periods of time are tiring and difficult for children [10,13,28]. Typing can also be an obstacle for children as their usual strategy is hunt-and-peck, turning even simple sentences into time-consuming tasks. An option for these children are pen-based interfaces, although these technologies can be error-prone for children's writing [23]. For young children whose coordination is not yet fully developed, touch screens offer a simple alternative to a mouse and keyboard [7]; however, this solution is not always feasible, so other ways of simplifying mouse and keyboard interactions are also desirable.

Children can learn to use a mouse as an input device, but a simplified interface is easier for them to operate. For example, one-click interfaces are simpler than those requiring double-clicking or dragging. Furthermore, children may confuse the mouse buttons. This problem is more prominent in preschool-aged children, but older children may also experience frustration if accidentally clicking on the wrong button produces unexpected results. Therefore, children's interfaces should have the same functionality for each mouse button whenever possible [7,12].

Dragging items across the screen is a common requirement in graphical user interfaces but may be problematic for children. Alternatives to dragging, such as clicking to attach an object to the cursor then clicking to drop the object in the desired location (also known as sticky-drag-and-drop) requires less manual dexterity (4-7yrs, 9-13yrs) [10,13,28]. Strommen [31] further discovered that continuous motion is easier for children (6-8yrs) when they click once to start the motion and again when they want to stop, rather than having to continuously move the mouse or hold down a mouse button.

The impact of having simplified mouse interactions can be considerable. Inkpen [13] found that while playing the same game, children (9-13yrs) were more motivated and solved significantly more puzzles using a point-and-click interface versus one that used drag-and-drop.

Make mouse interactions as simple as possible. One-click interfaces are easier than dragging or double clicking [7]
Make all mouse buttons have the same functionality [7,11]
Touch screens are good for young children who have difficulty using a mouse [7]

Young children have difficulty targeting small objects on the screen. Items should be large enough and distanced from each other to compensate for some inaccuracy in targeting [7,11,28]
Dragging movements are difficult for young children. Dragging should be accomplished by clicking on the object to attach it to the pointer, then clicking again to drop it in the desired location [10,13,28]
Interfaces should not require children to hold down mouse buttons for extended periods of time, especially if simultaneous mouse movement is necessary [13]
Using a mouse, continuous unidirectional motion on the screen is easiest for children when a "click-go-click-stop" interface is used, where children click the mouse to start the motion and click again once they want to stop [13]
Marquee selection should be accomplished by drawing an initial selection area on the screen then allowing users to shape it to the desired size by "pushing out" the edges of the area, rather than the traditional method of choosing one corner of the rectangle and dragging to its opposite corner [1]

Table 5. Motor Skills: summary of design principles

Children may lack the fine-motor control needed to target small items such as the icons and buttons used in traditional interfaces. Interfaces for children should include on-screen items large enough to compensate for some inaccuracy in targeting. Icons should also be spaced on the screen to minimize the chance that children accidentally press the wrong one (4-7yrs, 5-10yrs, 4-5yrs) [7,11,28].

Many children's interfaces avoid multiple selection tasks, because it is difficult for children to use traditional techniques such as drag-selection. The main difficulty for children is in picking an appropriate starting point so that all desired objects are included in the rectangle. Berkovitz [1] proposed an alternative where users click on a selection tool, then click the area where they want to select. A selection rectangle appears; users can then push out the edges of the rectangle until all desired objects fall within the rectangle. His studies show that this is a more intuitive interaction for both children (6-12yrs) and adult users, even with the extra step of clicking on the selection tool.

Tangibility

Children enjoy playing with stuffed toys and use them in many make-believe situations. Merging this play with computer interfaces leads to interesting toys. Strommen and Alexander's [32] work showed that computerized stuffed toys can successfully engage children in social interactions and be a play partner for young children. However, these 'smart' toys need not necessarily be cuddly in order to be engaging for children. Simply having a computerized, tangible device with which to interact can lead to valuable exploration and learning.

Direct manipulatives enhance traditional children's toys by adding computational power. These new toys encourage learning of concepts usually reserved for older children and help explore scientific phenomena through active participation [19,24]. Resnick et al. [24] successfully built a wide variety of direct manipulatives, including programmable beads to teach children about dynamic patterns, and wearable badges that can simulate the spread of viruses. Wyeth and Purchase [36] developed Electronic Blocks – sensor, logic, and action blocks which young children (3-8yrs) could combine to create increasingly complex constructions. Children found the blocks fun and challenging as they tried to find combinations to produce desired behaviours.

Alternative input devices such as pressure mats and video-tracked props encourage children to physically interact with their environment and become more involved in the interaction. Stanton et al. [27] extended KidPad so that it could be controlled through sensors embedded in floor mats (a magic carpet) and physical props such as large cardboard shapes. They found that children (5-7yrs) collaborated more and were more engaged using these input devices. However, the physical devices required more involvement and work on the part of the user and slowed down the interaction. These trade-offs were deemed acceptable, however, and even beneficial to the types of tasks undertaken by the children. Surprisingly, small changes in the appearance of the magic carpet and the props led to significant changes in the way children interacted with the devices. For example, when squares were used to indicate the position of the carpet sensors, the children jumped vigorously on them, but when arrows were used, they carefully placed a foot on the arrow.

A third research project resulted in the creation of Curlybots, small robots whose motion is programmed by physically moving the robot. It can loop through recorded segments, resulting in complex patterns of motion. Frei et al. [9] found that Curlybots succeed in engaging even young children (4+yrs) in exploring advanced computational concepts that would traditionally only be learned at a later age. Physical devices that allow play and experimentation help children build mathematical intuition in ways unsupported by conventional computers.

Children like tangible interfaces because they enjoy being able to physically touch and manipulate the devices [6,9,24]
Direct manipulatives allow children to explore and actively participate in the discovery process [9,19,24,36]
Physical props and having large input devices encourages collaboration [27]
Superficial changes to the design can produce very different physical interactions. Different interfaces emphasize different actions [27]

Table 6. Tangibility: summary of design principles.

Social/Emotional Development

Motivation and Engagement

Adult interfaces usually try to help users be as efficient and productive as possible. It is assumed that they have basic computer skills and that they have a task in mind where the computer is a tool to complete this task. Children, on the other hand, use technology for educational, social, and entertainment purposes. In order to be successful, a product needs to keep their interest and attention. This may mean sacrificing efficiency or turning away from adult design principles that advocate lean, simple interfaces.

One way to address engagement is by supporting the idea that children need to feel empowered and in control of the interaction. Researchers [8,10] agree that that successful interfaces give children control over the environment and let them set the pace of the interaction. Giving children the power to make decisions in this limited environment also allows them to learn about consequences of their actions in a safe setting. Said's [25] work on engagement in multimedia games also found that children (9-14yrs) wanted environments that let them be in control, set and achieve goals, and be part of the action.

Many systems for children aim to teach or provide practice with particular skills. In these cases, the value provided by the systems is only achieved when children spend time on and pay attention to the task at hand. Several means of motivating children through computer systems have been investigated. One is to provide entertainment either by designing a novel task or by embedding fun features so that children can take breaks from the main task. Another means of motivation is to provide animated on-screen agents to guide, encourage, or entertain children during the task. Third, extrinsic rewards can also encourage children to continue working at a task.

One concrete way researchers have found to motivate children is through the use of entertainment click-ons (or 'hotspots'). These are active regions on the screen that reward users who click on them by displaying an animation, sound effect, or other multimedia response. Children spend a significant amount of time finding and revisiting click-ons, usually as a break from other tasks. Super et al. [33] studied the success of entertainment click-ons in an educational mathematics game called Counting on Frank. They found that children (8-12yrs) used the click-ons most after periods of math-focused activity and that click-ons offering multiple responses, humour, and multimedia were most enjoyable and popular. They also determined that screen position and type of on-screen object affected the likelihood of finding the hotspot.

A second motivational tool for children's software is having an on-screen animated character. Lester et al. [17] investigated the use of animated pedagogical agents by including such an agent in an educational product teaching children about plants. Five versions of Herman the Bug were used, ranging from a muted agent to a fully expressive one. They found that simply having an animated character on the screen while children (12-14yrs) worked on an educational task positively affected children's experiences and encouraged them to use the software more frequently. Benefits such as increased learning occurred when the character was expressive and offered domain-specific advice and explanations. Hannah et al. [10] adds that on-screen characters should not be intrusive; their comments should be appropriately timed to complement current activities or to prime children for what is about to happen.

Another key factor in children's applications is the use of extrinsic rewards [10]. While designing tasks that offer intrinsic rewards is ideal, where solving a problem or learning a new skill is reward in itself, often this motivation is not enough to sustain children's interest. In children's technologies, extrinsic rewards often come in the form of multimedia messages, scoring systems, and bonus activities. These rewards should be consistent and available even if children repeat the same problem or activity levels, as they will often fail at more advanced levels and need to re-experience success to gain confidence for moving forward.

Pausch, Vogtle, and Conway [20] also uncovered the importance of having extrinsic rewards such as a scoring system in children's games. Their study examined the effectiveness of alternative input devices: children (6-17yrs) played a version of the well-known Pong game that originally did not display a score on the screen. They found that the children wanted a score display and in its absence kept score themselves by counting. The score display was later added to keep children focused on the task.

Technologies should give children the ability to define their experiences and be in control of the interaction [8,10,25]
Entertainment click-ons are an effective tool for engaging children. Multiple response click-ons are most popular while humorous and multimedia click-ons are most enjoyable [33]
Providing occasional entertaining diversions keep children engaged and motivated during learning tasks [33]
Animated pedagogical agents are useful for learning environments; even those who do not provide any advice or interaction are perceived positively [17]
Expressive, domain-specific agents are useful due to pedagogical benefits and positive affective impact [17]
On-screen character interventions should be supportive rather than distracting [10]
Activities should be inherently interesting and challenging so children will want to do them for their own sake [10]
Supportive reward structures that take into account children's developmental level and context of use help keep children engaged [10,20]

Table 7. User Experience: summary of design principles

Social Interaction

Social interaction is an important part of children's lives and this interaction in increasingly taking place online. While safety concerns must be addressed, children's technology can encourage and facilitate this interaction. Whereas children were previously limited to playing with other children in their neighbourhood, they can now easily interact with people from around the world to play, share, explore, and experience new ideas [8]. Technology also gives children the chance to interact in ways that are less intimidating than face-to-face, opening new opportunities for those who are shy, self-conscious, or unable to interact through traditional means [8]. Children now expect social interaction to be part of any online activity, even those that are traditionally solitary activities. Kaplan et al. [16] found that older children (10-14yrs) wanted the ability to communicate digitally with others while reading and working within a digital library, even when they were located in the same room. In the absence of such features, they resorted to alternating between two applications – the digital library and an instant messenger.

Research has also been conducted into how to promote positive affect and encourage social responses using the idea of Computers As Social Actors (CASA). Here, the computer is an active participant in the interaction rather than simply a tool. Strommen and Alexander's [32] work with digitized stuffed toys shows that children positively respond to praise, humour, and affection from the toys. A second project [18] looking at how children respond to help from a digital toy showed mixed results, including instances where the children were frustrated by inappropriate comments from the toy. Apparently, children enjoy social interactions with computers, as long as the interaction follows social conventions and meets their expectations.

Turkle [34,35] found that children are comfortable attributing psychological characteristics to machines and feel that their computers are very close to being alive. These beliefs may lead to higher expectations for their interactions with computer systems. Further work investigating CASA and children was conducted by Chiasson and Gutwin [2]. It was found that while advocates of CASA and the Media Equation claim that even the most basic interfaces elicit social responses from users, this may not be the case for children.

Children's technology should facilitate social interactions between children [8,16]
Children's technology should account for children's beliefs about computers and interact in a socially consistent manner [2,18,32,34,35]

Table 8. Social Interaction: summary of design principles.

Collaboration

Even when they have their own computers, children naturally group around one machine to work together. They enjoy playing together and like to share their experiences with friends and family [8]. They are often more successful as a result of this collaboration [4].

Giving children each their own mouse when collaborating encourages participation and cooperation. It also leads to greater user satisfaction [13,29,30,31]
Groupware interfaces should provide mutual awareness at all times [3]
Interfaces should support both "give" and "take" transfers of control to accommodate different interaction styles [13]
Single-Display Groupware is useful for children's co-located collaboration as they naturally group to one computer even when they have the opportunity to use separate machines [14,29]

Table 9. Collaboration: summary of design principles.

Typical desktop computer configurations lead to inefficient collaboration as children struggle for control of input devices and try to be active participants in the interaction [31]. Passive participants often point to the screen and give directives, but soon lose interest. One branch of HCI research addresses these issues through Single-Display Groupware, where several users share an output screen, but each has their own input devices. Single-Display Groupware systems can lead to greater user satisfaction, enhanced collaboration, and can allow users to work independently or together on a task [14,29,30].

Immaturity in group dynamics and negotiation leads to collaboration problems among children. With adult systems, including awareness information of what others are doing helps resolve some of these issues. However, at times even awareness is not enough to resolve conflict between children (10-11yrs) as neither will give up control of the item in question even though they know this would resolve the issue [3].

Gender differences are also apparent in interaction styles when children (9-13yrs) work together. Girls take turns by giving control to the other player, while boys take control from the other player when they feel it is their turn [13].

CONCLUSION

The catalogue presented here brings together a variety of disparate research on children's technology, and provides one of the first collections of design principles that are specifically oriented towards designing for children. The catalogue can be used by designers as either a formative guide or as a basis for evaluation of existing systems. However, our collection represents only a first step: there are a number of ways that the catalogue can be improved and refined. First, the age ranges seen in current research vary considerably. Since it is clear that many design principles are age-specific, more work must be done to test the principles with different age groups, and fill in holes in the catalogue where there is no information for children of a particular age.

Second, the majority of the research that we found identifies principles relating to cognitive and physical development rather than emotional and social development. For example, principles regarding literacy levels and mouse movement are clearly defined and can be readily incorporated into new interface designs. These are basic principles that any children's interface should follow. However, research that addresses emotional and social development are less well defined and are more difficult to use. Existing principles in this area must be adapted to fit new interfaces and their inclusion is more dependent on the individual goals of the interface. Those addressing emotional needs are especially problematic. More work is required in determining what interfaces best address the emotional development needs of children. Next, these need to be generalized into basic principles for interface design

Third, the set of design principles presented here provide a guideline based on current technology; however, as technology evolves, so will the design principles. For example, as new input devices are adopted, principles specific to mouse-based actions will require change. Finally, more work is needed to devise design and evaluation methodologies (such as heuristic evaluation) that incorporate and make use of these principles. Despite the need for more research, this catalogue gathers a valuable set of information for designers of children's technology. The common theme through the principles presented here is the need to adapt interfaces to how children naturally behave, to accommodate for children's developing skills and knowledge, and to make products that are enjoyable from a child's point of view. It should now be obvious that design principles for adults do not apply for children's interfaces, and designers must look at children to determine what best meets their needs.

REFERENCES

1. Berkovitz, J. Graphical Interfaces for Young Children in Software-based Mathematics Curriculum, *Proc. ACM CHI 1994*, 247-248.
2. Chiasson, S., and Gutwin, C. Testing the Media Equation with Children, *Proc. ACM CHI 2005*.
3. Cockburn, A, and Greenberg, S. Children's Collaboration Styles in a Newtonian MicroWorld, *Proc. ACM CHI 1996*, 181-182.
4. Danesh, A., Inkpen, K., Lau, F., Shu, K., Booth. K. Geney: Designing a Collaborative Activity for the Palm Handheld Computer, *Proc. ACM CHI 2001*, 388-395.
5. Druin, A., A Place Called Childhood. *ACM Interactions*, v.3, 1996, 17-22.
6. Druin, A., Children as Our Technology Design Partners, in *The Design of Children's Technology*, Allison Druin ed., Morgan Kaufmann, 1999.
7. Druin, A, Bederson, B., Hourcade, J.P., Sherman, L., Revelle, G., Platner, M., and Weng, S.. Designing a Digital Library for Young Children, *Proc. ACM JCDL 2001*, 398-405.
8. Druin, A., and Inkpen, K. When are Personal Technologies for Children? *Personal and Ubiquitous Computing* v.5, 2001, 191-194.
9. Frei, P., Su, V., Mikhak, B., and Ishii, H. Curlybot: Designing a New Class of Computational Toys. *Proc. ACM CHI 2000*, 129-136.
10. Hanna, L., Risden, K., Czerwinski, M., Alexander, K.J. The Role of Usability Research in Designing Children's Computer Products. in *The Design of Children's Technology*. A. Druin ed., Morgan Kaufmann, 1999.
11. Hourcade, J.P., Bederson, B, Druin, A., and Guimbretiere, F. Differences in Pointing Task Performance Between Preschool Children and Adults Using Mice. in ACM ToCHI, 11, 4, 2004, 357-386.
12. Hourcade, J.P., Bederson, B., and Druin, A. Preschool Children's Use of Mouse Buttons, *Proc. CHI 2004*, 1411-1412.
13. Inkpen, K. Drag-and-Drop Versus Point-and-Click Mouse Interaction Styles for Children. *ACM ToCHI*. v.8, no.1, 2001, 1-33.
14. Inkpen, K., Booth, K.S., Gribble, S.D., and Klawe, M. Give and Take: Children Collaborating on One Computer, *Proc. ACM CHI 1995*, 258-259.
15. Johnson, M.P., Wilson, A., Blumberg, B., Kline, C., and Bobick, A. Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic Characters. *Proc. ACM CHI 1999*, 152-158.
16. Kaplan, N. et al. Supporting Sociable Literacy in the International Children's Digital Library. *Proc. ACM IDC 2004*, 89-96.
17. Lester, J., Barlow, S., Converse, S., Stone, B., Kahler, S., and Bhogal, R. The Persona Effect: Affective Impact of Animated Pedagogical Agents. *Proc. ACM CHI 1997*, 359 - 366.
18. Luckin, R., Connolly, D., Plowman, S., and Airey, S. Children's interactions with interactive toy technology. *J.Computer Assisted Learning*, v.19., 2003, 165-176.
19. Mikhak, B., Martin, F., Resnick, M., Berg, R., and Silverman, B. The Children's Wearable Machines: Handheld and Wearable Computers too. *Proc. IEEE HUC 1999*, 31-43.
20. Pausch, R., Vogtle, L., and Conway, M. One Dimensional Motion Tailoring for the Disabled: A User Study. *Proc. ACM CHI 1992*, 405 – 411.
21. Plaisant, C., Druin, A., Lathan, C., Dakhane, K., Edwards, K., Vice, J.M., and Montemayor, J. A Storytelling Robot for Pediatric Rehabilitation. *Proc. ACM ASSETS 2000*, 50-55.
22. Rader, C., Brand, C., and Lewis, C. Degrees of Comprehension: Children's Understanding of a Visual Programming Environment, *Proc. ACM CHI 1997*, 351-358.
23. Read, J., MacFarlane, S., Casey, C. Oops! Silly Me! Errors in a Handwriting Recognition-based Text entry Interface for Children. *Proc. NorCHI 2002*, 35-40.

24. Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., and Silverman, B. Digital Manipulatives: New Toys to Think With, *Proc. ACM CHI 1998*, 281-287.
25. Said, N. An Engaging Multimedia Design Model, *Proc. ACM IDC 2004*, 169-172.
26. Sedighian, K., and Klawe, M. An Interface Strategy for Promoting Reflective Cognition in Children, *Proc. ACM CHI 1996*, 177-178.
27. Stanton, D. et al. Classroom Collaboration in the Design of Tangible Interfaces for Storytelling, *Proc. ACM CHI 2001*, 482-489.
28. Steiner, K., Moher, T. Graphics StoryWriter: An Interactive Environment for Emergent Storytelling, *Proc. ACM CHI 1992*, 357-364.
29. Stewart, J., Bederson, B., and Druin, A. Single Display Groupware: A Model for Co-present Collaboration, *Proc. ACM CHI 1999*, 286-293.
30. Stewart, J., Raybourn, E., Bederson, B., Druin, A. When Two Hands are Better Than One: Enhancing Collaboration Using Single Display Groupware, *Proc. ACM CHI 1998*, 287-288.
31. Strommen, E Children's use of mouse-based interfaces to control virtual travel, *Proc.ACM CHI 1994*, 405-410.
32. Strommen, E., and Alexander, K. Emotional Interfaces for Interactive Aardvarks: Designing Affect into Social Interfaces for Children, *Proc. ACM CHI 1999*, 528-535.
33. Super, D., Westrom, M., and Klawe, M. Design Issues Involving Entertainment Click-ons. *Proc. ACM CHI 1996*, 179-180.
34. Turkle, S. *Life on the Screen: Identity in the age of the internet*, Touchstone Press, 1995.
35. Turkle, S.. Cyborg Babies and Cy-Dough-plasm: Ideas about Self and Life in the Culture of Simulation. In *Cyborg Babies: From Technosex to Technotots*. R. Davis-Floyd and J. Dumit, eds., Routledge, 1998.
36. Wyeth, P. and Purchase, H. Using Developmental Theories to Inform the Design of Technology for Children. *Proc. ACM IDC 2003*, 93-100.