

# Simplicity in cognitive assistive technology: a framework and agenda for research

Clayton Lewis

Published online: 8 December 2006  
© Springer-Verlag 2006

**Abstract** Technology offers substantial benefits to the many people with some form of cognitive disability. But the power of technology often comes in a package whose complexity is a barrier to many users, leading to calls for designs, and especially designs for user interfaces, that are “simple”. This paper analyzes the idea of simplicity, and suggests (a) that simplicity in a user interface is not a unified concept, but rather has distinguishable facets, and (b) that simplicity must be defined in terms of the cognitive capabilities of a user, so that what is “simpler” for one user may be “more complex” for another. Despite (b), the prospects for universal design in this area are good, in that interface technology with the flexibility needed to produce “simple” interfaces for a range of users with different cognitive strengths will be of value in addressing the overall design space of interfaces for a broad audience. While it is possible to sketch the outlines of a useful theory of simplicity, the sketch reveals much that is not fully understood. It also reveals opportunities to rethink the architecture of user interfaces in a way that will benefit user interface development generally.

**Keywords** User interface design · Cognitive disabilities

## Introduction

There are a great many people with cognitive disabilities

While definitions differ in detail, and while good data are not available for the world population, it is a reasonable estimate that more than 20 million people in the USA [4], and perhaps 400 million worldwide, have some form of cognitive disability. Cognitive disabilities affect mental functions like learning, problem solving, remembering, planning, and decision making, and may be caused by developmental disorders, brain injury, the effects of aging, or some forms of severe mental illness.

People with cognitive disabilities often have other disabilities, as well

It is not useful to try to separate sharply responses to cognitive disabilities from responses to other forms of disability. Many people with cognitive disabilities also have problems in hearing or seeing, difficulty in speaking, and/or difficulty with motor control, including in walking, pointing, or typing.

Computational technology offers a wide variety of benefits for people with cognitive disabilities

Capabilities of people with cognitive disabilities cover a very wide range. While for some people learning a simple cause and effect association (for example, that pressing a button controls a toy or a lamp) is a serious challenge, other people with cognitive disabilities can learn quite complex computational skills. For example, Hart [12] describes a study in which five students with

---

C. Lewis (✉)  
Coleman Institute for Cognitive Disabilities,  
University of Colorado, 4001 Discovery Drive, Suite 210,  
Boulder, CO 80303, USA  
e-mail: clayton.lewis@colorado.edu

autism, and moderate to mild cognitive impairment, successfully completed a high school level course on using the Excel spreadsheet program. Other people can carry out the steps of a task, but have trouble remembering what the steps are; they may be able to use a handheld prompting device that displays the steps in sequence, with audio or pictorial prompts. See [31, 32] for a survey of aspects of cognitive disability and how technology can address them.

As Wehmeyer et al. note, complexity of the user interface is often seen as limiting the use of technology by people with cognitive disabilities. Commercial software, and handheld devices, usually have screens cluttered with many controls and cues: buttons, hyperlinks, menus, icons, images, and text. Some people with cognitive disabilities find it difficult to locate the information and controls they need in the clutter. Similarly, performing a task may require carrying out a lengthy sequence of actions, often accompanied by the replacement of one complex display by others.

Design guidelines call for simplification of designs as an approach to making the functionality of such systems more accessible to people with cognitive disabilities. For example, the first guideline in [5] is “Web content should be simple”. But what is simplicity? And how can we realize it in interface design?

#### Simplicity is relational

This paper will argue that simplicity is not a structural property of an interface or system, but rather is defined by the relationship between the cognitive demands of using the system and the cognitive capabilities of the user. When the demands match the capabilities, we say the interface is simple.

#### Simplicity is multifaceted

Because different cognitive capabilities are in play in using an interface, there are different aspects of simplicity that need to be considered separately. That is, the informally defined term “simplicity” lumps together a number of different aspects of an interface.

There are a variety of different design tactics for making interfaces simpler

Each facet of simplicity can be addressed by one or more methods for changing cognitive demands. The result is a portfolio of design tactics.

Different simplicity tactics are needed for different users

Because simplicity reflects the relationship between demands and capabilities, and because people’s capabilities vary, so simplicity varies from one person to another.

Questions of understanding and implementation are abundant in designing for simplicity

Moving beyond an impressionistic, “common sense” idea of simplicity exposes gaps in available knowledge of cognitive functions and their role in human–computer interaction. Also, the practical need to respond with acceptable cost to the varying needs of a wide audience poses challenges to current approaches to software architecture.

#### A framework for thinking about simplicity

In the spirit of earlier efforts to understand the diversity of user interfaces and their functions [26, 14, 19, 29], Table 1 proposes an outline of the generic structure of a problem in human–computer interaction.

In the table, the left column represents a user, and the right column represents the world. Specific *target states* are identified in the world. Each target state represents a possible outcome in some activity in which the user is engaged. For example, if the user is engaged in a communication activity, target states might pick out situations in which Fred has received a particular message *M*, or that Fred has received a different message *M'*, or that Ethel has received *M*, and so on.

In between the user and the world stands a *system*. The system presents certain *cues* (for example, hyper-

**Table 1** Framework for discussing interface structures

User	System	World
Has an intention, a <i>target state</i> . Carries out <i>actions</i> , often guided by <i>cues</i> .	Presents <i>cues</i> to the user. Supports <i>actions</i> by the user. Responding to user actions, acts on the world so as to change its state.	Has a collection of possible <i>target states</i> : Possible state 1, Possible state 2,...Possible state <i>n</i>

In a successful interaction, the user carries out a sequence of actions, in response to which the system acts on the world so as to attain a state corresponding to the user’s intention

link labels, button labels, or menus), and the user performs certain *actions* allowed by the system. The system responds by acting on the world (and, in general, by changing the cues presented to the user). In an interaction, the user, perhaps guided by the cues, carries out a sequence of actions, and the response of the system to these actions is to change the world so as to realize one of the target states. In a successful interaction, the target state that is reached agrees (in some sense) with the *intention* of the user.

As just mentioned, it is not necessary that the user is guided by the cues (the user may not see them, or may ignore them), and in fact in some interfaces there are no or few cues. Also, there can be situations in which the user lacks a well-formed intention. These variations will not figure in the subsequent discussion.

### Clutter, breadth, and depth

In the case of almost any commonly used computer technology today, for example, a word processing program, a first observation is that the collection of target states in this picture is astronomically large. It necessarily follows that the space of available *sequences of user actions* must be equally large. If the user can have an intention that picks out any one of a large number of targets, and if the system is to support their realizing these intentions, they must be able to choose from an equally large number of candidate action sequences.

Suppose that the adopted design approach initially is to assign a single action to each of the possible targets for the interaction. Suppose further that it is required that the interface provides a cue of some kind for each action, as is done in modern interfaces. It is easy to see that for systems with large numbers of targets, clutter must result: the interface will have to provide a large number of cues.

In fact, this problem is so apparent, for all but very small target collections, that it becomes clearly necessary to associate sequences of actions with the targets, if we are to have any hope of providing cues for the required actions. That is, we will be driven to increase the *depth* of our interface, the number of actions required in sequence to select a target, as a way to reduce its *breadth*, the number of actions available at any one time [17].

A toy example follows to illustrate these ideas. Suppose a system supports exactly 128 target states. A maximally broad interface would present 128 actions, with their cues, for example, in the form of a menu with 128 choices. With this interface the user can accomplish their task with just one action, but will be looking

at a screen with 128 cues on it. A maximally *deep* interface, in this same situation, will use a sequence of seven menus, each with two choices. In this case, each screen is “simple”; it only needs to present two cues, one for each choice, but the user must work through seven of these screens to select the particular target they want.

It is not hard to see that there are intermediate designs, as well. For example, eight choices can be presented on a screen, rather than two, supporting the user’s task with three menus in sequence, say two eight-way menus followed by a final menu with two choices.

### Is depth “simpler” than breadth?

By focusing on the notion of “clutter” in this discussion, one may come to feel that the deeper designs are “simpler” than the broader ones. After all, the individual screens in the deeper designs can have many fewer cues on them, because of the limited number of actions they have to support. But there are countervailing considerations.

### *Devising good cues is hard in deep interfaces*

Consider the problem of finding someone in an address book. In a wide interface, one scans a list of people’s names and addresses, which are excellent cues for the specific action, selecting one of the entries. In a deep interface, the list of people has to be subdivided in some way, and then cues have to be devised for the actions of choosing among these subdivisions. If the division is made by (say) job category, the requirement holds that the user has to know the job category of the person in question. In real life, finding computer science departments on typical university websites is hard, if one sticks with the hierarchy of menus, because that department is sometimes in the engineering subtree and sometimes not. As Furnas and colleagues [9] have shown, finding good terms to describe conceptual categories, even ones as concrete as the headings in a yellow pages directory, is really impossible, in the sense that the variation in interpretation that will be given to the labels is unmanageably large.

### *Remembering intentions*

A second issue with deeper interfaces, one that is of particular importance for some people with cognitive disabilities, is the need to retain one’s intention over the life of a long interaction. In a broad interface, only

a few actions are needed to accomplish the task, while in a deep interface several actions will be needed. During the time it takes to complete these actions, and across the changes in display that will occur on each action (so as to swap in the cues for the actions at the next stage) the user has to keep their intention in mind. Maintenance of intention is difficult for many people; it is a commonplace of the cartoonist that an old person may go into the bedroom intending to get dressed, but gets into bed instead. See [24], Sect 4.1, for a survey of human errors attributable to failures of intention.

So, is breadth “simpler” than depth? No. The conclusion from the comparison is that which form of interface will be more effective depends on a variety of factors, including differences in the user’s cognitive capabilities, as well as task differences.

In the discussion so far two factors play a significant role, namely the quality of the cues available for the deep interface, which depends on the task, and the ability of the user to maintain intention over an extended interaction, which depends on the user. Additional factors influence the comparison, including:

#### *Discrimination*

In a broad interface the user has to be able choose which of many alternative cues is the appropriate one [15, 16]. How easy this is depends not only on the nature of the cues, which depends on the task, but also on the user’s particular cognitive capabilities.

#### *Response suppression*

In a broad interface the user has to avoid responding to inappropriate cues. Doing this is an aspect of *executive function* [13, 21]. As shown in many studies [20], response suppression is difficult in situations like the Stroop test, in which color names are printed in colored ink, with the color of the ink *not* matching the color name. The participant is asked to name the colors of the ink, which is very hard to do, because of the need to block the strong response to the written color name. Such difficulties may be a source of difficulty in dealing with “clutter”, when a desired control has to be chosen from among others that are also frequently used.

#### *Visual search*

Detecting an item of interest against a background of distractor items is difficult, especially when the distractors are visually similar to the item of interest. For some people with cognitive disabilities, this kind of

visual search is especially difficult [6, 23]. A deeper interface, in which each display contains fewer distractors, because it presents fewer cues, can be easier to use.

Notably, people with autism can show superior visual detection performance in some tasks [22], but this effect seems to depend on the kind of stimulus cues that are available [11]. This clearly shows the difficulty in devising a single interface that will be effective for a range of users and applications.

#### *Display size*

Broad interfaces are not appropriate for small screens, as there isn’t room for the cues. User characteristics influence this, in that a person with a visual impairment, like many people with cognitive disabilities [23], may need to have cues that are physically large, meaning that the number of cues that can be presented on a given screen is reduced. Similarly, if a user has a motor impairment, on-screen targets for touch selection or mouse selection may have to be larger, again limiting the breadth of interface that is possible.

#### *Response repertoire*

A dramatic example of the influence of user characteristics (though not cognitive characteristics) on interface structure is the use of scanning interfaces. Some users with severe motor impairments can physically carry out only a tiny repertoire of actions, that may include (for example) sucking a tube. In a scanning interface they are presented with a sequence of cues, all associated with the same action, not different actions as is usual in graphical user interfaces. As the cues are presented, in sequence, they make their response only when the desired cue appears. This arrangement permits a large range of alternatives to be controlled with a very limited response repertoire.

#### *Differing target frequencies*

This discussion of the depth-breadth tradeoff in user interface structure has introduced the framework for describing interfaces, and has developed the point that interface structure simplicity depends on the individual user and their specific cognitive capabilities. Differences in intention maintenance, or visual search, could tip the balance either way.

Trading off depth and breadth is only one way to influence the effective simplicity of a design for a user. A variety of other tactics play off variation in how

often different tasks must be performed, or, in the presented framework, how often different target states must be reached. A design in which common targets can be reached very quickly can be advantageous, even if rare tasks are harder. The principle here may be familiar from the design of Morse code, in which different numbers of dots and dashes are assigned to the different letters of the alphabet. The commonest letters have the shortest codes, reducing the overall number of dots and dashes that must be used to send a passage of text.

An extreme form of this tactic is pruning out target states altogether from a design, so that there is *no* sequence of user actions that will reach them. Of course this can make way for a simpler interface, since it allows for a reduction in depth *and* breadth. But it is only viable if the excluded targets are really very rare.

Even after any really rare target states are pruned, the remaining target states often differ enough in frequency to make it worthwhile to use a design in which commoner targets are reached by shorter action sequences. At the same time, often there is also value in assigning more conspicuous cues to the actions used to reach the commoner targets. The structure of the graphical user interface, or *GUI*, for any common application will show these effects. The menu items associated with less commonly-needed actions are buried in subsidiary menus, so that they are not seen at a higher level of the interface, and so that more actions are needed to select them. The payoff for this demotion is that the cues for actions that *are* commonly needed can be made more prominent, and fewer overall actions are needed to reach the associated common targets. For users for whom the assumed task frequencies are accurate, the resulting interface is simpler, both in the sense that it is effectively narrower, and in the sense that, at the same time, it is shallower. Less clutter is achieved without paying the price of increased depth, for common targets.

Given that the desirability of exploiting differing target frequencies is accepted, there are a number of rather different design tactics for managing it. In particular, there is a spectrum of methods for determining target frequencies, ranging from static methods, in which frequencies are determined (usually implicitly) once and for all, and set into the design, to fully dynamic ones, in which the frequencies are allowed to differ for different users and situations, and the interface actually shifts in response to observed frequencies. A hypothetical email program can be used to illustrate the range of possibilities.

### *Static designs*

Suppose the designer determines that the use of blind carbon copies is quite rare. Then they will not include a bcc field in the standard header for new messages. Unless they have pruned blind carbon copy altogether from the range of reachable targets (that is, a target like “Fred received message M, and so did Ethel, but Fred does not know Ethel got it” is simply excluded) there will be some action the user can take that will expose the bcc field, or allow this function to be controlled in some other way. Even if a user needs to use bcc all the time, with this design it will always be complicated to do so.

### *Dynamic designs*

Here the designer contrives for the system to detect that a user is often using bcc. When this happens, the interface is changed so as to make bcc more accessible, for example, by starting to show the bcc field in the headers of new messages without the user having to take any action to request it. For the user who needs bcc often, this interface will be “simpler”, in that fewer actions are needed to control that common (for them) function.

So, is the dynamic design preferable? It depends. Note that the interface as the user sees it is actually changing in this design. For some users, for whom learning what cues to look for, and how to respond to them, is difficult, these changes could cause real trouble. It appears that people are able to *proceduralize*, or automate the application of, their knowledge, when they work in a consistent task environment [1], so that they become faster and more accurate by not having to examine the features of the environment each time they perform the task. When the environment changes, as it does in these dynamic designs, proceduralization can be blocked, or can create interference in responding to the changed interface. Some readers may have had experience with systems that reorder menu items, based on frequency of use, and may have been annoyed by not finding a menu item where they expect it, or, worse, wrongly choosing the item that is now in the position formerly occupied by their familiar choice.

The Apple Mail program keeps track of the folder to which the last message was moved, and makes this folder the default destination. This is a significant convenience for users who normally move nearly all messages to the same folder. But it happens regularly that a user (the author, in fact) moves a message to a



different folder, and then misses the fact that the default destination has been reset, leading to a number of messages being placed in the wrong folder before the problem is detected.

How important these issues are depends on the cognitive capabilities of the user. A user for whom proceduralization is slow and difficult may be little affected. On the other hand, a user for whom interpreting common cues, such as words, is difficult may be forced to rely more on proceduralization, and hence be more affected. That is, someone who has difficulty making a menu selection based on reading the textual menu prompts may be set back when the position of items changes, more than someone who can read the prompts would be.

### *Configurable designs*

An intermediate point on the static–dynamic spectrum of designs for variable target frequency is *configuration*. In this case, the system does not measure target frequencies and automatically adjust, but the structure is not fixed, either. Rather, the user can restructure the interface so as to change how easily accessed, and how prominently cued, different functions are. This kind of design is now the rule for common application GUIs, and for handheld devices like cellular phones and PDAs. To illustrate using the bcc example, I recently sent a message with a bcc, for the first time since adopting my current mail application. To do this I changed the configuration of the GUI, so that now the bcc field is exposed in the headers of messages I compose. In Microsoft Word it is possible to control quite flexibly which of the very many available controls are actually shown. For example, one can suppress choices relating to tables, or to macros, or to other functions a particular user may never need.

This design approach can be seen as a compromise between fully static and fully dynamic designs. On the one hand, the design can respond to differences among users in what they need to do, or even to changes in what a given user needs to do, unlike a strictly static design. On the other hand, users who do not want their interfaces to change do not have to make changes, or they can make them temporarily.

But there is a price to be paid for configuration. The complexity of the configuration apparatus itself shows up not only in the GUI itself, where the configuration actions, and their cues, must be exposed somewhere, but also in places like manuals and help screens. That is, configuration features themselves are a source of

clutter. This problem will be further discussed later in this paper.

### *Context-dependent designs*

Also intermediate between static and dynamic designs are those in which the frequency of targets is estimated (usually implicitly) based on the activity the user is engaged in. A familiar example is the way the reply function is structured in email systems. Without considering context, if I send an email message it could be to anyone. But if I am currently viewing a message, it is quite likely that I will want to send a message to that person, that is, that I want to reply. The interface is structured so as to make it much easier to send a message to the author of the message that I am reading, than to anyone else. We can see this as an instance of the variable target frequency principle at work: the interface is structured to make a common target easy to reach, with the determination of the common target being based on task context.

Context-dependent features are widespread. Consider that an application keeps track of the last folder in which you opened a document, and will propose that as the folder to look in when you next ask to open something. Your cell phone probably keeps a list of recent calls, and makes it easier for you to contact the people connected with those calls than other people.

Do context-dependent features simplify an interface? Yes and no. Yes, in that (if done well) they shorten the action sequence for common tasks. No, for some users, in that they may introduce a degree of variability into the interface, which, as previously discussed, may create problems for users for whom coping with variation is difficult, or who must rely on learning standard procedures to compensate for difficulty in interpreting cues.

### *Activity integration*

The email reply example brings out another aspect of some context-dependent interface structures. There is a sense in which the reply design often gets a user action for free, namely, the user action of selecting the message to which a reply is wanted. Often, the user is reading mail, using whatever features are provided for doing that, and carrying out whatever actions are needed. In mid-flight, seeing a message to which a reply is wanted, the user can now start the reply process without the need to carry out *any actions* to select the to-be-replied-to message. The effective simplicity of

generating the reply is thus significantly increased, by taking advantage of the larger activity context.

Of course there are times when one needs to reply to a message when one is *not* already looking at it, and then one has to generate extra actions. But much of the time these actions are not needed.

### Sharing the load

Configurable interfaces open up an additional possibility for simplification. As mentioned above, configurable interfaces commonly carry baggage associated with the configuration facility itself. But what if the configuration facility is not included in the application GUI at all, but is accessed by a third party? This allows an attractive division of labor. If assistance is available from someone who can configure the application appropriately, the user gets the benefit of a stable interface, and one with no configuration-related clutter. (Actually, there is still some price to pay, in that the documentation and help available for the configurable interface will almost certainly be more complex, because of the variation in the application that the documentation has to describe. In principle, documentation could be dynamically modified to reflect changes in configuration. In practice it is hard to make instructions and explanations sufficiently modular to make this work.)

This option seems attractive for someone for whom the functional limitations of a fully static interface are unacceptable, but for whom coping with an interface that changes would be difficult.

Once a third-party is introduced, one more simplification tactic can be seen. Instead of fully pruning rare target states, rendering them completely unreachable, a design can include a path for reaching them through third-party effort. We will see an example of this in a case study, below.

### Modality effects

Is an interface that allows to choose email recipients by clicking a picture of them (like that in [27, 28]) simpler than a conventional interface? An overall argument is that an interface is simple for a user if the demands of the interface are well matched to the capabilities of the user. One of the demands of an interface is interpreting the cues it presents. So, for a user who can recognize a picture, but not read a name, an interface that uses picture cues will be simpler.

It appears that being able to interpret cues in a modality is not an all or nothing thing. Multimodal

presentation, as used by the information site TheDesk (<http://www.thedesk.info>) for presenting information about Medicaid programs to people with cognitive disabilities, includes both text and a digitized speech track. This is a help to people who can comprehend a message better if it is both spoken and written than if it is presented in either modality alone [3].

Further, within the domain of images, pictures, and symbols, there are wide differences in interpretability for different people. Commercial devices that use non-textual cues use a wide variety of different picture and symbol sets (for example, some based on relatively detailed drawings or even photographs, and others based on schematic, high contrast, black and white symbols [2]). While some of the differences in effectiveness of these images are likely due to differences in vision among people (for example, astigmatism), some may be due to differences in cognitive processing. Some users seem better able to use abstract shapes as cues than seemingly more meaningful pictures (Bodine, personal communication.)

While they fall outside the current focus on cognitive capabilities, there are analogous design issues associated with actions, as well as with cues. As previously discussed, people differ in what their action repertoire is, just as they differ in whether or not they can read text. An interface that requires actions outside the user's repertoire, or actions that can be made only with difficulty, will not be usable.

The simplification tactics just enumerated are based on the argument that simplification is restructuring an interface so as to shift the demands of using it from areas of relative weakness to relative strength for users. Clearly, this idea is applicable to meeting the needs of people of all kinds, not just those with cognitive disabilities. This is the general message of universal design, the effort to design systems that are usable by the broadest possible audience [7, 18, 25]. It follows that these tactics can be employed in general interface design, not just in design for people with cognitive disabilities.

### Case study

Many of the simplification tactics previously discussed can be illustrated by an analysis of the Digital Mailbox, a device designed by CaringFamily, LLC (see <http://www.caringfamily.com>) to support electronic communication to and from elders who do not have or wish to have computers. The designers felt

that this was an application calling for maximum attention to simplicity in the user interface.

The Digital Mailbox consists of a printer and scanner, connected to the Internet via a custom telephone interface. Family members (say, the adult children of an elder, and grandchildren) enroll in a service that allows them to send email to an address for the elder. Their mail, which can contain pictures, is sent to the Digital Mailbox, where it is automatically printed on a daily schedule.

The following discussion focuses on the means by which the elder can send email out, using a radically simplified interface: a single button. That is, the elder sends mail from the Digital Mailbox with no commands, dialing, or menus, using a single button press. How is this done?

Let us consider two cases, replies, and original messages. When a message is received by the elder, it is printed out automatically. To reply, the elder writes a note on the printout, places it in the scanner, and presses the single button. The system recognizes the printout on which the message is written, and automatically routes the reply to the author of the original message.

To send a message that is not a reply, the elder selects a piece of pre-printed stationery, addressed to the intended recipient (the stationery has a thumbnail photo, as well as the name, of the addressee on it). They write a message on the stationery, insert it in the scanner, and press the single button. The system recognizes the stationery and routes the message to the intended recipient. The system also prints a new sheet of stationery to replace the one that was just used, maintaining the elder's stock.

The following are the simplification principles that this design illustrates.

#### Target state pruning and sharing the load

In the interaction as described, there are many target states reachable by conventional email that are just not available to the Digital Mailbox user: they can send messages *only* to enrolled family members, from whom they have received messages, or for whom they have pre-printed stationery. For some elders, this drastic pruning is acceptable. For others, an expansion is available via sharing the load: they can send a message to an enrolled family member, asking that it be sent on to someone else.

#### Context-dependent design

The design incorporates the logic of ordinary email reply, as discussed above.

#### Activity integration

The interface for the Digital Mailbox is actually broader than superficially appears. It is perfectly true that there is only one button, but it is *not* true that pressing that button is the only action needed to send the message. In fact, *selecting the stationery* is also required, and is cued by the printing on the stationery. Activity integration between the natural act of selecting the stationery, and the actual sending process using the Digital Mailbox, allows the sending process itself to be extremely simple.

What happens if the elder does not select stationery, but instead writes on ordinary paper, and scans that? Sharing the load enters here, again. The message is sent to a family member who is the default recipient, who routes it based on any available indication of the intended recipient, for example a salutation.

#### Modality effects

The use of pictures of the addressees on the stationery, and (something not emphasized above) the heavy use of pictures as message content, makes the system less reliant on text.

Preliminary experience with the digital mailbox design has been positive. Systems have been installed for 13 elders in one informal trial, and 473 messages have been sent by these elders using the one-button interface, over periods ranging from 4 to 8 months. Formal field trials are starting as of this writing.

#### Research opportunities

The Digital Mailbox is a simple example, in more than one sense. Partly because of radical target state pruning, that is to say, quite limited functionality, extreme simplicity is within reach. Because of that, no adjustment of the design to better fit the particular cognitive capabilities of different users has been needed. The needs of people with serious motor or visual impairments (beyond supplying expanded fonts when needed) are not addressed in this example. Applying the ideas proposed in this paper more widely will require a good deal of work.



### Assessment of cognitive capabilities

As we have seen, providing a “simple” interface for someone requires knowing what their cognitive strengths and weaknesses are. In particular, components of executive function [13], such as the ability to maintain an intention, or the ability to suppress responses to irrelevant cues, influence what level of depth and breadth will be effective for someone. Similarly, how readily someone can proceduralize their interactions, and, on the other hand, how much they have to rely on proceduralization, because of difficulty in processing cues, will influence what degree of dynamic adjustment in an interface will work for them. We lack good ways of assessing these strengths, and developments along this line are needed. Better understanding of the organization of the cognitive functions involved [13, 21] will also help. Increased work on cognitive modeling of the performance of people with cognitive disabilities would also be valuable. The success of Anderson and colleagues [1] in modeling learning, especially proceduralization, in a complex human-computer interaction task suggests what could be done.

### Software architecture

There are many opportunities to increase the flexibility of software, so as to reduce the cost of adapting it to the needs of particular users. As mentioned earlier, existing interfaces have a high degree of configurability, but support for dynamic reconfiguration is limited. Support for configuration by third-parties is uneven. Architectural support for modality effects, that is, for changing the modalities used for cues, and for actions, is the focus of some work [30] but more is needed. We cannot easily replace textual cues by pictorial ones, for example, or even replace cues in one language by cues in another without a lot of work. The geometry of the screen is a big limit in this respect, and the question arises of whether there could be a way to manage screen space so as to provide more flexibility. In general, it is not yet usually possible to plug together devices that offer (for example) scanning interfaces and devices running generic application functionality, and even when it is, issues of physical packaging, portability, and the like, often emerge, crossing the line from software to hardware.

### Advances in textual and nontextual communication

Many interfaces make heavy use of text, both for cueing and for content. For people for whom text is

difficult to process, this is a huge barrier. Recent advances in language engineering, for example in machine translation [10], suggest that human language processing techniques may become available that have the potential to simplify to make text it easier to comprehend. At the same time, advances in the inexpensive capture, processing, and storage of images suggest the opportunity to explore new, more flexible and powerful, ways to communicate without text [8]. Similar advances in processing speech offer additional possibilities.

### Activity structure

The reply example shows how significant simplification can be obtained by a combination of context-dependent structure in an interface, where commoner targets are assigned shorter, better cued actions sequences, and activity integration, where work done in one part of an activity is leveraged to simplify another part. Doing this more generally requires analyzing desired user activities, with a particular view to discovering what aspects of context predict the frequency of next steps (as when reading a message predicts that replying is likely enough to be worth supporting). Many similar integrations have been discovered (for example, integrating dialing a call with reading a reminder that one should make a call) but more systematic ways of identifying likely next steps of actions are required.

### Sharing the load

One could say that the whole spectrum of care for people with cognitive disabilities represents sharing the load, but technology may be designed as to better support this. New communication media, like text messaging, have changed the temporal and also social structure of communication, so that more people are “online” with communication partners more of the time. If these technologies can be used to provide support for people with cognitive disabilities, they have the potential to foster greater independence.

### Inclusion in research

Though the thrust of this paper has been theoretical, progress in human-computer interaction rests heavily on empirical research, and user testing of product designs. Sadly, people with cognitive disabilities are rarely included in these studies. This has the dual impact of reducing the extent to which the needs of people with cognitive disabilities are reflected in

research and development, and limiting the understanding that the research and development community has of people with cognitive disabilities. There is a great opportunity for progress on this.

## Conclusion

“Simplicity” is a misleadingly simple concept. Analyzing the role of simplicity in user interface design leads to recognition of its relational nature, as “simplicity” consists of a good fit between the demands of an interface and the capabilities of a user. On these bases, a range of design tactics can be identified for enhancing this fit. Possible enhancements of current knowledge and skills can also be identified, for example in assessing cognitive functions and in structuring more flexible interfaces.

**Acknowledgments** The ideas in this paper were stimulated by discussion at a symposium on cognitive technologies at IBM Watson Research Center, October 6–7, 2005. I thank the organizers and attendees. I also thank Dr Cathy Bodine, Director of the Rehabilitation Engineering Research Center on Advancing Cognitive Technologies, and Dr Michael Williams, CTO of Caring Family LLC, for many useful conversations. I am grateful to the National Institute for Disability and Rehabilitation Research and to the Coleman Institute for Cognitive Disabilities for research funding. I thank three anonymous reviewers for many useful suggestions.

## References

- Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y.: An integrated theory of the mind. *Psychol. Rev.* **111**(4), 1036–1060 (2004)
- Beukelman, D.R., Mirenda, P.: Augmentative and alternative communication. Paul H Brookes, Baltimore (1998)
- Blattner, M., Glinert, E.P.: Multimodal integration. *IEEE. Multimedia.* **3**, 14–24 (1996)
- Braddock, D., Rizzolo, M.C., Thompson, M., Bell, R.: Emerging technologies and cognitive disability. *J. Spec. Educ. Tec.* **19**(4), 49–56 (2004)
- Bohman, P., Anderson, S (2005) A Conceptual framework for accessibility tools to benefit users with cognitive disabilities. WebAIM: Web Accessibility in Mind. Retrieved from <http://www.webaim.org/techniques/articles/framework/>. Cited 6 Feb 2006
- Carlin, M.T., Soraci, S., Goldman, A.L., Mcilivane, W.: Visual search in unidimensional arrays: a comparison between subjects with and without mental retardation. *Intelligence.* **21**, 175–196 (1995)
- Center for Universal Design (1997) The principles of universal design. North Carolina State University. Retrieved from [http://www.home.earthlink.net/~jlminc/tools\\_principles.html](http://www.home.earthlink.net/~jlminc/tools_principles.html). On Cited 6 Feb 2006
- Danielsson, H., Jönsson, B.: Pictures as language. International Conference on Language and Visualisation, Stockholm, Sweden, 8–9 November (2001)
- Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T.: The vocabulary problem in human-system communication. *Commun. ACM.* **30**(11), 964–971 (1987)
- Geer, D.: Statistical machine translation gains respect. *IEEE. Comput.* **October**, 18–21 (2005)
- Greenaway, R., Plaisted, K.: Top-down attentional modulation in autism spectrum disorders is stimulus-specific. *Psychol. Sci.* **16**(12), 987–994 (2005)
- Hart, M.: Autism/Excel study. In Proceedings of ASSETS '05 (The 7th International ACM SIGACCESS Conference on Computers and Accessibility), pp.136–141. Baltimore, MD, 9–12 October 2005. ACM, New York (2005)
- Hedden, T., Gabrieli, J.D.E.: Insights into the aging mind: a view from cognitive neuroscience. *Nat. Rev. Neurosci.* **5**, 87–96 (2004)
- Hutchins, E., Hollan, J., Norman, D.: Direct Manipulation Interfaces. In: Norman, D., Draper, S. (eds.) *User centered system design*, pp.87–124 (1986)
- Kitajima, M., Blackmon, M.H., Polson, P.G.: A comprehension-based model of Web navigation and its application to Web usability analysis. In: McDonald, S., Waern, Y., Cockton, G. (eds.) *People and Computers XIV—Usability or Else!* (Proceedings of HCI 2000, University of Sunderland, St. Peter's Campus, UK, September 5–8), pp.357–373. Springer, Berlin Heidelberg New York (2000)
- Kitajima, M., Blackmon, M.H., Polson, P.G.: (2005) Cognitive architecture for website design and usability evaluation: Comprehension and information scent in performing by exploration. In Proceedings of HCI-International 2005,
- Larson, K., Czerwinski, M.: Web page design: implications of memory, structure, and scent for information retrieval. Proceedings of CHI'98 human factors in computing systems, pp.25–32. ACM Press, New York (1998)
- Law, C.M., Jacko, J., Peterson, B., Tobias, J.: Universal designs versus assistive technologies: research agendas and practical applications. In Proceedings of ASSETS '05 (The 7th International ACM SIGACCESS Conference on Computers and Accessibility), pp.2–3. Baltimore, MD, 9–12 October 2005. ACM, New York (2005)
- Mackinlay, J.D., Card, S.K., Robertson, G.G.: A Semantic analysis of the design space of input devices. *Hum-Comput. Interact.* **5**, 145–190 (1990)
- MacLeod, C.M.: Half a century of research on the stroop effect: an integrative review. *Psychol. Bull.* **109**(2), 163–203 (1991)
- Miyake, A., Friedman, N.P., Emerson, M.J., Witzki, A.H., Howerter, A.: The unity and diversity of executive functions and their contributions to complex “frontal lobe” tasks: a latent variable analysis. *Cognitive. Psychol.* **41**, 49–100 (2000)
- O’Riordan, M.A., Plaisted, K.C., Driver J., Baron-Cohen S.: Superior visual search in autism. *J. Exp. Psychol. Human.* **27**(3), 719–730 (2001)
- Oross, S.O. III, Woods, C.B.: Exploring visual perception abilities in individuals with intellectual disabilities: assessment and implications. In: Soraci, Jr S., Murata-Soraci, K. (eds.) *Visual information processing*, pp. 35–79. Praeger, Westport CA (2003)
- Reason, J.: *Human Error*. Cambridge University Press, Cambridge (1990)
- Rose, D.H., Meyer, A., Hitchcock, C.: (eds.): *The Universally Designed Classroom: Accessible Curriculum and Digital Technologies*. Harvard Education Press, Cambridge MA (2005)
- Shneiderman, B.: Direct Manipulation: a step beyond programming languages. *IEEE. Comput.* (1983)

27. Sutcliffe, A., Fickas, S., Sohlberg, M.: Personal and Contextual Requirements Engineering. In Proceedings of 13th IEEE International Conference on Requirements Engineering, Paris (2005)
28. Sutcliffe, A., Fickas, S., Sohlberg, M., Elhardt, L.A.: Investigating the usability of assistive user interfaces. *Interact. Comput.* **15**, 577–602 (2003)
29. Tweedie, L.: Characterizing interactive externalizations. In Proceedings of ACM CHI'97, pp.375–382. (1997)
30. Wang, J., Mankoff, J.: Theoretical and architectural support for input device adaptation. In Proceedings of Conference on Universal Usability, pp.85–92. (2003)
31. Wehmeyer, M.L., Smith, S.J., Palmer, S.B., Davies, D.K., Stock, S.: Technology use and people with mental retardation. In: Glidden LM (ed.) International review of research in mental retardation 29, pp.293–337. Academic Press, San Diego CA (2004)
32. Wehmeyer, M.L., Smith, S.J., Palmer, S.B., Davies, D.K.: Technology use by students with intellectual disabilities: an overview. *J. Spec. Educ. Tec.* 19(4), (2004)