A METHODOLOGY FOR INTERFACE DESIGN FOR OLDER ADULTS

Mary Zajicek
Department of Computing, School of Technology, Oxford Brookes University, Wheatley Campus,
Oxford OX33 1HX, UK

Keywords: User models, interface design patterns, design methodology, universal accessibility, older adults, speech systems, memory support, strategies at the interface.

Abstract: This paper puts forward a new design method based upon Alexandrian patterns for interface design for particular user groups. The author has created a set of interface design patterns for speech systems for older adults with the aim of supporting the dynamic diversity in this group. The patterns themselves reflect a significant body of research work with this user group uncovering important information about how they interact with speech systems. The design knowledge embedded in these patterns is therefore closely linked to knowledge about the user and enables interface designers to clarify which users are excluded from their software.

1 INTRODUCTION

This paper is concerned with a new methodology for interface design for older adults using speech systems, where carefully describing the users for whom the system is defined will promote a better fit between user and system.

The paper outlines the characteristics of older adults as computer users and introduces the idea of patterns to encapsulate design recommendations for speech systems for this user group. Older adults to a greater or lesser degree experience memory impairment and a reduction in their ability to build strategies at the interface, characteristics which are modeled within the patterns.

The paper discusses the challenge of complying with legislation which promotes software accessibility for all user types and patterns are suggested as providing a useful framework within which software developers can analyse their prospective user group in this regard.

Eight interface design patterns are analyzed in terms of the user models they implicitly and explicitly encapsulate, and also in terms of the users they disadvantage.

2 WHY WOULD WE WANT TO MODEL USERS' ABILITIES?

Recently significant legislation has been introduced in the US and UK aimed at encouraging designers to ensure that software does not exclude certain user groups. Age related impairments older adults often exclude older adults from the use of standard software. For example the print may be to small or the multi-tasking interface may rely to heavily on good memory or their ability to build complex strategies in order to use the software.

A better understanding of older adults as users through modeling their attributes will enable interface designers to clarify which users are excluded by their software.

3 THE ORIGIN OF INTERFACE DESIGN PATTERNS

The idea for patterns and pattern language originated in the domain of architecture; with the publication more than twenty years ago of Christopher Alexander's book *The Timeless Way of Building* (Alexander, 1979). He proposed that one could achieve excellence in architecture by learning and using a carefully defined set of design rules, or

Zajicek M. (2004).
A METHODOLOGY FOR INTERFACE DESIGN FOR OLDER ADULTS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 81-88
DOI: 10.5220/0002611000810088

patterns: and although the essence of a beautifully designed building is hard to define the application of patterns for room design etc can contribute to the design of the whole.

A pattern describes an element of design possibly together with how and why you would achieve it. For example Alexander created patterns which describe ways of placing windows in a room and designing a porch which achieves a successful transition between inside and outside a building. These include textual descriptions and diagrams or photos (Alexander, 1977).

Patterns for human-computer interface design were first discussed in the late nineties, and currently there exist a range of different pattern forms. Some pattern builders choose a purely narrative approach such as those found in the *Design of Sites* (Van Duyne et al, 2002) whereas others are more structured. Martin van Welie for example sets out patterns under the headings *Problem* and *Solution* Van Welie, 2002) A comprehensive list of pattern forms can be found at Sally Fincher's *Pattern Form Gallery* (Fincher, 2003).

The pattern form used in this paper, is based on Jennifer Tidwell's *UI Patterns and Techniques* (Tidwell, 2002) where the pattern has four sections, *Use When*, *Why*, *How*, and *Examples*. A fifth section entitled *Tradeoffs* has been included from the claims approach, as there are always tradeoffs when designing speech dialogues and these should be made explicit.

A full day workshop, 'Perspectives on HCI Patterns: Concepts and tools' at CHI 2003 was attended by several of the pattern developers referenced above, where they came up with the pattern Language Markup Language (PLML)(Fincher, 2003) which promises to provide a generic pattern format.

4 THE OLDER ADULT AS A USER

Adults as they get older experience a wide range of age related impairments including loss of vision, hearing, memory and mobility, the combined effects of which contribute to loss of confidence and difficulties in orientation and absorption of information. Significantly, age related impairments affect people at different rates and even any one individual from day to day. The need for Design for Dynamic Diversity to accommodate this dynamic diversity of ability in older adults was first proposed by Newell and Gregor (Newell & Gregor, 2000) and is demonstrated in interface design by Gregor Newell and Zajicek (Gregor et al, 2002).

Furthermore, gathering interface requirements from older adults requires considerable skill and understanding of the user group. Newell and Gregor also proposed (Newell & Gregor, 2000) that standard User Centered Design techniques, which rely on relatively homogeneous user groups for user testing, should be replaced by User Sensitive Inclusive Design, which seeks out diversity, in order to ensure that systems are truly usable by older adults result.

In summary then interface design for older adults is more complex than for standard groups, making optimum interface design more difficult to achieve. It is therefore particularly important that instances of design which work well for older adults should be carefully documented and passed on for other designers to use.

5 ADVANTAGES OF INTERFACE DESIGN PATTERNS FOR OLDER ADULTS

Guidelines provide a useful form of 'advice' for designers and are necessarily generalized for a range of applications. However the information embedded in the guideline has been distilled either from a form of craft knowledge, or theory or through experimentation. If the guideline comes with the 'reason' attached together with an example of the use of the guideline, the designer is a stronger position to utilize the information.

The W3C, Web Access Initiative Guidelines (Web Access Initiative Guidelines, 1999), which were developed for Web designers so that they could make their Web pages more accessible for non-standard users, are accompanied by the reasons for the guidelines, which enables the designer to be aware of who she or he is excluding if they do not follow the guidelines.

Designers therefore, especially those designing systems for older adults, would benefit from access to the information, experimental or otherwise, that gave rise to the guideline.

Academic papers of course exist which describe the experiments from which the guideline was distilled, but these contain more information than the designer requires. The argument here is that the information relevant to good design practice should be set out in a structured and informative way for easy access by the interface designer.

Interface designers are rarely older adults themselves and therefore have no concept of how it would feel to access a computer when you are experiencing the combined effects of memory, sight,

and mobility loss coupled with reduced confidence that comes with slower processing of visual, spatial and verbal information. Furthermore, the dynamic diversity of ability in older adults poses particular challenges for interface designers.

A robust set of design patterns with a linking language is therefore a particularly important requirement for those designing systems for use by older adults. A set of clear and informative patterns together with information on how the patterns may be used together in a system i.e. the pattern language, would enable interface designers to access best practice and help them to create sympathetic and successful designs for older adults.

Importantly the patterns will reflect the experience of older adults through experimentation and observation, which the designers themselves are lacking. This in itself will nurture good design and provide a framework in which mistakes need not happen.

6 EXAMPLES OF INTERFACE DESIGN PATTERNS

Patterns for speech systems possess different properties compared with the more visually orientated graphical user interface patterns of Tidwell and van Welie (Tidwell, 2002),(Van Welie, 2002), and indeed the architectural patterns of Alexander (Alexander, 1977). Speech dialogues use two forms of input, speech and keypad, and output in the form of a speech message. The usability of the dialogue hinges on its structure and the quality of the output messages. Patterns relevant to speech systems therefore must include those concerned with the construction of output messages, and also those related to dialogue structure.

This section presents several patterns which deal with the quality of output messages in speech systems for older adults, and can be formed in either pre recorded or synthetic speech. They are categorized according to function, *Menu Choice Message*, *Confirmatory Message*, *Default Message*, *Context Sensitive Help Message*, *Talk Through Message*, and *Explanation Message* together with the dialogue structure patterns *Error Recovery Loop* and *Partition Input Message*. This is not a complete list of patterns for speech systems for older adults and can be developed further with additions and refinements.

6.1 Menu Choice Message

This pattern encapsulates design knowledge derived from experiments carried out with older adults using the voice Web browser BrookesTalk which offers a menu selection of functions in a Voice Help message which talks novice users through their interaction with the browser (Zajicek & Morrissey, 2003).

It was found that older adults were confused by long messages with many options and that they remembered more information from short messages. This phenomena was nor seen in younger people. Older adults also tend to remember the first and the last elements of a menu better.

Users were found to respond more favorably to key press menu selections expressed by mnemonic letter keys such as *A* for *address*, compared with function keys which caused much confusion. Finding letters on the keyboard was a major problem for older people using the voice Web browser.

Pattern name:	Menu Choice Message
Use When:	When the range of choices offered in the dialogue is small. Large numbers of choices should be broken up into sub menus. Keep the message as short as possible.
Why:	Experiments show that older adults are confused by long messages and forget the possible options or remember only the first or last.
How:	Short messages offering at most three selections should be used only with very well defined options where the user using pre-existing knowledge (not strategies that are worked out at the time) can see easily which of the selections will lead them to their goal. Example 1. demonstrates how this can be done. When there are three options, place the most commonly selected options last and first. Use mnemonic letters to identify menu items for key press entry, as in Example 2.
Example:	1. "Would you like to deposit or withdraw money?", rather than 'Would you like to perform a credit or debit transfer?" 2. "You have no page loaded. Would you like to Enter an address, press A Perform a search, press S

Tradeoffs:	Menus usefully group end goals in a speech dialogue, and the smaller the menu the greater the number of interactions that will be needed to reach the end goal. The tradeoff is that reducing the number of options in the menu lengthens the interaction. Despite this drawback, experiments carried out with the VABS (Zajicek et al, 2003) showed that the short messages approach was most successful with older adults.

6.2 Confirmatory Message

The importance of confirmatory messages was demonstrated during experiments carried out with older adults using the voice Web browser BrookesTalk (Zajicek & Morrissey, 2001). Here confirmation was produced by a personal helper who answered *yes* or *no* to users' questions about their interaction. In the domain of speech dialogues the principle of confirmatory action is expressed in confirmatory messages that reassure the user that the interaction is going well.

Pattern name:	Confirmatory message
Use when:	After the user has input data, or made a choice or performed some other action that they might not be sure of.
Why:	To confirm that data has been input correctly or to draw the users attention to the progress of the dialogue. Research has shown that user confidence can be increased by confirmatory action and can aid the construction of conceptual models.
How:	After a data input event try to arrange for the output of a confirmatory message that contains the input data. In Example 1. the words in italic represent input data that has been embedded in a confirmatory sentence. The message confirms that the system still 'knows' that it is talking to Mary and mentions 7 pm so that the user can check the input. The words evening and appointment serve to frame the concept of the reminder. Confirmatory messages can also be used as in Example 2. to confirm that the dialogue is proceeding satisfactorily.
Example:	1. '*Mary* you have successfully booked a call reminder at *7 pm* on the evening before your appointment" 2. "You have successfully logged on to the Voice Activated Booking System and have asked for a call reminder for your appointment on Wednesday the 8th November. What time would you like your call reminder?"
Tradeoffs:	Confirmatory messages increase the user's confidence in their interaction and aid the construction of conceptual models of the speech system. The tradeoff is between increasing confidence and adding to the length of the interaction. Confirmatory messages demand yet more attention of the user and can be irritating to confident users. Research has shown however, especially for systems designed primarily for first time users, that confirmatory messages should be used.

6.3 Default Input Message

Default input messages proved to be very useful in the VABS system where users' input reminder call times were not recognized. Users were usually happy with the default time of 7 pm for a reminder call as shown in Example 1.

Pattern name:	Default Input Message
Use when:	When a user is required to input data which has been misrecognised or when the user has failed to understand that data must be input. This can be used only when the input data is relatively imprecise and a default which would just about do is possible as in Example 1.
Why:	Because the system requires some input data in order to continue the interaction, or where the user has consistently avoided inputting the required data and is likely to terminate the interaction.
How:	The default input message should appear when an input has been misrecognised a certain number of times.
Example:	1. "Would 7 pm on the evening before your appointment be a good time for a reminder call?"

Tradeoff:	The default input is possibly not the input that the user intended and they may or may not be satisfied with it. The possibility of not being satisfied is the tradeoff against the frustration of trying several times to enter the correct input or leaving the interaction with the task incomplete.

6.4 Context Sensitive Help Message

The VABS features several help areas within the dialogue. Users can say *Help* at any time during their interaction and a message, which is specifically relevant to the area of the dialogue they are currently using, will be output. Context sensitive help was invoked several times in the VABS [16] and was considered to be very useful.

Pattern name:	Context Sensitive Help Message
Use when:	When help in the context of the current task would be useful.
Why:	This message is a direct response to a user asking for help which maps closely to human-human dialogues when help is requested, and represents the most efficient way in which help can be provided. Context is important as it ensures that the message will be as short as possible as the message will contain only information relevant to the task in hand. Older adults experience difficulty in absorbing spoken information, and this type of message will provide only information which is necessary at the time.
How:	Identify areas in the interaction which relate to identifiable tasks and create explanatory messages for each task. Arrange for an input of *Help* to trigger the relevant context sensitive help message.
Example:	1. "This system uses voice recognition to understand your commands that should be spoken clearly. Giving your name helps the system to locate your current sessions and gives you access to more functions. You need to contact the center during office hours to register your name". 2. "This system is designed to offer users the ability to book or cancel a computer taster session. Speak your commands clearly and try to use the words given in the question".
Tradeoffs:	There are no direct usability tradeoffs. This is the best way to provide help. Unfortunately some older adults forget that they have the option to ask for help so alternative methods which do not require the user to take the initiative, must be used as well. The only tradeoff is in the extra programming effort required to set up the help areas and messages.

6.5 Talk Through Message

Experiments with Voice Help for BrookesTalk showed that speech output that talks older adults through their interaction could enable people to use software where they hadn't been able to use it before (Zajicek & Morrissey, 2001).

Pattern name:	Talk Through Message
Use when:	When it would be helpful to be told where you are in the interaction and what you can do next.
Why:	Older adults find difficulty in building strategies at the interface, mainly because this activity relies on short-term memory. Memory loss means that it is difficult to remember what you did last time and to build up a model of how the interaction works.
How:	Where a new task is about to be tackled, talk through messages should be inserted to help with orientation to the new process. It is best to arrange for more competent users to switch off the messages when they are no longer needed.
Example:	1. 'You have arrived at the point where you tell us when you would like to come for your IT Taster Session. Sessions run from 10:30 to 15:30 every hour. Please say something like 'ten thirty' to see if the session is available".
Tradeoffs:	Talk through messages irritate confidant users and slow them down. They slow down even novice users and make for more speech output to listen to, but are considered to be most worthwhile for older adults.

6.6 Explanation Message

Older adults find speech interaction confusing. While younger people can adapt to the ways computers behave older people find it more difficult. Any explanation of what is happening can be useful.

Pattern name:	Explanation Message
Use when:	Use when the dialogue is behaving in a non-intuitive way or in a way that does not map onto usual human-human dialogue.
Why:	Interaction with a speech dialogue is often not intuitive to older adults, so it should explain itself as much as possible.
Example:	1. "There will be a short delay while your name is found on our database"
Tradeoff:	Extra messages make the dialogue longer. The tradeoff is between making the dialogue longer or making it clearer.

6.7 Error Recovery Loop

Errors and error recovery represent the main usability problem for speech systems. Standard menu driven systems often start with a long set of instructions in a bid to avoid errors happening. Older users are not able to remember these messages, which also slow down the dialogue, rendering them useless. The pattern described here directs designers to embed instructions in an error recovery loop: in effect to wait for the error to happen and then try to recover it.

This approach is most useful in dialogues which are used mostly by experienced users who are unlikely to require any instruction and will if they use the dialogue successfully never have to listen to an error recovery message.

Pattern name:	Error Recovery Loop
Use when:	When errors in data input are likely to occur.
Why:	Because older adults cannot remember lengthy preliminary spoken instructions about data input. It is best to let them try to input data and if it goes wrong invoke an error recovery message.
How:	Count how many times a data input occurs and on each count invoke an increasingly detailed error recovery message. In the examples below Example 1. simply gives instructions for efficient input, but the more detailed Example 2. provides information about which might help the user work better with the system.
Example:	1. "Your name has not been recognized. Please speak slowly and clearly into the telephone. 2. "The system is trying to match your name against the names it holds in the database. Please try to speak your name in the same way that you did when you registered for the Voice activated Booking System.
Tradeoffs:	This form of error recovery does not prepare the user in advance for possible errors, as they have to create the error before it is invoked. The tradeoff is against providing long instructions before the user embarks on a task.

6.8 Partitioned Input Message

The interaction paths taken through the VABS system by older adults was compared with the optimum path for each task (Zajicek et al, 2003) It was found that data input tasks showed the greatest deviation from the optimum route and this was because of misrecognition of utterances. Misrecognition of input causes considerable frustration in speech interaction and often leads to abandonment of the dialogue.

The Partitioned Input Messages in effect perform binary chops on the possible entry data (Brownsey et al, 1993) For example when a time for a reminder call is required instead of being asked to enter the time, the user would be asked 'Would you like your reminder call in the morning or afternoon?' as normally occurs when the session organizer sets up the reminder call. If the answer were morning the system would then respond 'Before eleven o'clock or after?' The dialogue would continue to halve the search area until a time is selected.

Pattern name:	Partitioned Input Message
Use when:	Use for any discrete input data which has up to sixteen possible values. Sixteen values requires four questions be asked to reach the correct value. More than sixteen values would require too many questions. This message type is particularly useful when the required input might be confusing. Example 1. tries to cope with IT Taster

	Sessions on the VABS which are held on the half hour, while most users suggest a time on the hour if asked for input (they also ignore instructions asking them to use the half hour)
Why:	This message replaces data input which is the most difficult part of speech dialogues. It is extremely error prone and older adults find difficulty in recovering from errors.
How:	Set up messages that divide the number of possible input in two each time as shown in Example 1.
Example:	1. (i) "Would you like to attend for taster sessions in the morning or afternoon of Wednesday 7th June?" (ii) "Would you like to come before or after 11 in the morning?" (iii) "Would you like to come at 11.30 or 12.30?"
Tradeoff:	This method takes longer than direct data input but the tradeoff is that it reduces the frustration of misrecognition.

7 PATTERNS AND USER BEHAVIOUR

Eight interface design patterns are analyzed in terms of the user models they implicitly and explicitly encapsulate, and also in terms of the users they disadvantage.

Figure 1 sets out the patterns described above with reference to the user models they support and which users are disadvantaged by the pattern. With the following key:

A – Models dynamic diversity
B – Models memory impairment
C – Models strategy building impairment
D – Good HCI design for all users
E – Could slow down younger users

Pattern Name	**A**	**B**	**C**	**D**	**E**
Menu Choice Message		X	X	X	
Confirmatory Message			X	X	
Default Input Message			X	X	
Context Sensitive Help	X	X	X	X	
Talk Through Message		X	X		X
Explanation Message		X	X		X
Error Recovery Loop	X	X	X	X	
Partitioned Input		X	X		X

Figure 1: Properties of the patterns

With this categorization in place together with the patterns themselves inexperienced software designers can reference the patterns of the user groups they support.

Such designers will be able to see easily to what degree they are addressing concerns of universal accessibility and also be aware of the tradeoff in supporting one user model while at the same time disadvantaging another. For example we can see in Figure 1 that some interface design patterns that are useful for older adults lead to sub-optimal design for younger users.

Most importantly of all designers will be able to search out those design patterns which provide the optimal solution by supporting dynamic diversity and hence all users.

8 CONCLUSIONS

Explicit information about the accessibility of particular software design features by different sectors of society is able to support those who are required to make challenging design decisions concerning design for universal access, and who need to be aware of which users will be excluded from their software.

The legislative demand for interface design for a wider proportion of society will complicate the design decisions that software developers will have to make and in these cases the use of patterns categorized according to user model can be of great benefit.

ACKNOWLEDGMENTS

The author would like to thank Age Concern Oxfordshire for their valuable support for the work described above.

REFERENCES

Alexander, C., 1979, The Timeless Way of Building, Oxford University Press.

Alexander, C., Ishikawa, S., Silverstein, M., 1977, A Pattern Language: Towns, Buildings, Construction, Oxford University Press.

Brownsey K. Zajicek M. Hewitt J., 1994, A structure for user oriented dialogues in computer aided telephony, Interacting *with Computers* 6(4), 433 – 449.

Fincher, S., 2003, CHI 2003 Workshop Report, *Interfaces* No. 56, Journal of the BCS HCI Group.

Fincher, S., 2003, HCI Pattern-Form Gallery, http://www.cs.ukc.ac.uk/people/staff/saf/patterns/gallery.html, (last accessed 23.4.3).

Gregor, P., Newell, 2002, A., Zajicek, M., Designing for Dynamic Diversity – interfaces for older people, Proceedings of 5th ACM/SIGAPH Conf. on Assistive Technologies.

Newell A.F. & Gregor P., 2000, User Sensitive Inclusive Design – in search of a new paradigm, Proc A.C.M. Conference on Universal Usability, Washington, DC Nov. pp39-44.

Tidwell, J., 2002, UI Patterns and Techniques, http://time-tripper.com/uipatterns/about-patterns.html (last accessed 22.4.3).

Van Duyne, D, Landay, J., Hong, J., 2002, The Design of Sites: Patterns, principles and processes for crafting a customer-centered web experience, Addison Weseley.

Van Welie, M., Interaction Design Patterns, http://www.welie.com/patterns/index.html, (last accessed 12.5.3), 2002.

Web Access Initiative Guidelines, 1999, http://www.w3.org/TR/WCAG10/, (last accessed 12.5.3).

Zajicek M. & Hall, S., 2000, Solutions for elderly visually impaired people using the Internet, In S. McDonald, Y. Waern, G. Cockton (eds) *People and Computers XIV – Usability or Else!*, Proceedings of HCI 2000, pp 299 – 307

Zajicek M., Morrissey W., 2001, Speech output for older visually impaired adults, Proceedings of IHM-HCI 503 - 513

Zajicek M., Morrissey, W., 2003, Multimodality and interactional differences in older adults, Special Issue "Multimodality: a Step Towards Universal Access' of *Universal Access in the Information Society*,(ed. N. Carbonell) Springer.

Zajicek M., Wales, R., Lee, A., 2003, Towards VoiceXML Dialogue Design for Older Adults, In Palanque P., Johnson P., O'Neill E (eds) *Design for Society*. Proceedings of HCI 2003