Utterance-Based Systems: Organization and Design of AAC Interfaces

Timothy Walsh
Dept. of Computer and Information Sciences
University of Delaware
Newark, DE 19716
1-302-831-2712
walsh@cis.udel.edu

ABSTRACT
Electronic Augmented and Alternative Communication (AAC) systems provide a method for people with severe speech impairments to communicate by selecting what they desire to say, and have the system "speak" it for them. AAC interfaces present many unique design challenges due to the wide variation of physical ability among AAC users. A goal of technological aids for AAC users is to bridge the communication gap, increasing both rate and comprehensiveness of communication [2]. Here we focus on the design of an utterance-based system developed for literate, high-functioning adults interacting in public goal-directed situations with unfamiliar partners. This research involves the design and development of a coherent and intuitive AAC interface for an utterance-based system built upon theoretical evidence and observation of commercial-grade AAC interface software.

Categories and Subject Descriptors
K.4.2 [**Computers and Society**]: Social Issues ---Assistive technologies for persons with disabilities; D.2.2 [**Software Engineering**]: User Interfaces --- Human Computer Interaction.

General Terms
Design, human factors, reliability

Keywords
AAC, communication, interface design

1. INTRODUCTION
AAC is a field of study that attempts to provide technology/intervention to aid communication for those with a disability that makes it difficult to speak in an understandable fashion.. AAC devices are assistive communication devices that aid in the expression of thought for those affected by a disability such as cerebral palsy, amyotrophic lateral sclerosis, traumatic brain injury, assorted muscular dystrophies, and congenital deafness. Assistive communication devices are designed for a wide range of users that vary significantly in their age, as well as cognitive, linguistic, and motor abilities.

A typical AAC device consists of an *interface* (AAC interface) and a *language set*: the *interface* defines the way the user interacts with the device, while the *language set* defines the selectable language elements provided by the interface. Most AAC devices translate the user's input to the interface into machine vocalized speech. Those with high motor abilities are able to utilize an interface similar to a standard keyboard which requires the user to accurately and quickly interact with the device's many compact buttons. On the other hand, those with limited motor abilities benefit from larger or sparser keys to allow for easier selection. In cases of severe motor impairments, an AAC device's physical interface may only consist of a single button.

AAC interfaces are a form of communication assistive technology that aims to increase communication rate and informativeness of AAC users in a conversational context [2]. A major theme among this type of interface design is to maintain a good balance of accuracy of the message and the time required to convey said message. This balance of *speed* and *accuracy* is maintained through the complexity of the elements of the language set, as well as how each element is organized and presented to the user. Standard letter-by-letter keyboard interfaces are extremely slow for AAC users with severe motor impairments whose communication rate might average 10-15 words per minute[1][3, 7]. This slow rate of input motivates techniques that optimize the amount of information conveyed per user input: by increasing the value of user input to an entire words or phrases, each interaction with the device conveys a greater amount of information. Increased information per element of the language set yields a different set of issues as the user must be able to quickly and intuitively access each language element to make the system effective. Systems that employ higher complexity elements of the language set require a higher cognitive workload. Additionally, the more information that is conveyed per user input, the lower the probability that the input accurately expresses the intended thought of the user.

Utterance-based AAC interfaces allow for entire phrases or *utterances* to be stored as the elements of the language set. These systems require that utterances must be stored in advance, and then accessed at the point the user wishes to communicate. An utterance-based system implements a three-phase system use model to address the design challenges regarding speed and accuracy; preparing the user by incorporating prestored text into the interface that the user would anticipate (*prepare*), organizing this prestored text via some method so that the user can remember both what messages are stored and how to get to them at time of need (*system storage*), and making the interface simple to use by providing methods for quickly accessing relevant prestored text and allowing quick alterations to the message to more accurately convey meaning (*use*).

2. BACKGROUND
There has been much previous work in the area of utterance-based AAC systems (see Todman et al. 2008 for an overview). Some of these have been shown to be effective in certain contexts (*e.g., social conversation between unfamiliar*

[1] In comparison to spoken communication at 130-200 words per minute or more.

partners), but the organization and use of these systems remains an open area of research.

The work of Hoag et al. focuses on the preparation and system storage phases of the utterance-based system sequence [4]. By investigating effects of various prestored text on the attitudes and behaviors of unfamiliar communication partners in goal-directed public situations, Hoag et al. have recommended an organization strategy for utterance-based systems [6]. They focus on systems to be used in public, goal-oriented situations with unfamiliar partners (*e.g., going to a restaurant and interacting with a waiter*). A fundamental aspect of the design process was the understanding the relationship of prestored messages to discourse context. Building the model relied on findings from user studies carried out by Bedrosian et al. [1] that looked at the perceptions of the communication partner when messages that were not completely appropriate with respect for the discourse context used. These studies give insight on the implications of AAC devices for both the AAC user and the conversation partner; specifically on the conversation partner's reaction to flawed messages produced by the AAC user. Flawed messages were categorized as either *repetitive*, *too much information*, *not enough information*, or *partially relevant*. Messages that were *partially relevant* were rated significantly worse than those of the other three categories.

Because of this, their AAC system design follows Vanderheyden [11] who suggests a schema driven organization based on schemas as described by Schank & Abelson [9]. This project is an implementation of the Bedrosian et al.'s design where the focus is on the design of a simple interface to incorporate issues of human computer interaction to develop an optimal access to prestored text to both accurately and quickly convey messages. Associating prestored text with the events in a familiar sequence enables the user to progress through the interface as s/he participates in the event. This results in a reduced need to search for a message, an overall increase of speed of communication and accuracy [6].

3. DEVELOPING THE INTERFACE

This research took place in three stages of development:

1. Researching the topic and exploring existing AAC interface design.
2. Designing an AAC interface based on discoveries made, incorporating elements crucial to utterance-based design.
3. Implementation (and revision) of the utterance-based AAC Interface design proposed.

3.1 Research and Documentation

Developing an interface for AAC users motivated the need for a requirement analysis for users of said product. This core software engineering principle yields a better understanding of the user needs, ultimately resulting in a better product [5]. Understanding AAC, AAC users, and issues of interface design became the initial step in compiling such an analysis. Further research into similar interface design was accomplished through documenting and carefully observing the DynaVox™ Series5 Speaking Software (with InterAACT[2]) AAC interface. This analysis yielded a *model* of what an AAC user is: a precise understanding of the specifications an AAC interface must fulfill in order to meet the requirements of the AAC user population.

With a firm set of specifications that an AAC interface must fulfill, background in specific-user interface design was researched. The following four fundamentals were derived via the *Seven Principles for Transforming Difficult Tasks into Simple Ones* [8]:

1. *Consistency throughout interface* – the user should feel familiar with the interface no matter the goal the user is attempting to accomplish; simplify the structure of the task.
2. *Logical graphical user interface system* – the interface should function in some manner that would be reasonable for the user to expect.
3. *Uniform functionality* – the components of the interface are strictly defined and will adhere to the guidelines as such; exploit the power of constraints and create a intuitive mapping.
4. *Intuitive organization of information* – the information that is presented to the user is organized in a way that the user will either intuitively grasp the organization or will be able to logically understand the method of organization used.

3.2 Designing an AAC Interface

The design of the AAC interface focused heavily on integrating these four fundamentals into the interface and addressing the augmentative and alternative communication challenges (*i.e., speed v. accuracy*). An utterance-based approach and event-oriented organization were chosen as the primary means of maximizing the user communication rate. Each element of the language set is an utterance, thus containing an entire expression of thought, and is organized according to the an event-oriented *schema*. Adapting an event-oriented structuring of information reduces the need to search for a message as the schema presents only messages relevant to the user's event context. An intuitive sentence building and editing system was also developed to incorporate many of the same design fundamentals. Sentence editing is one of the methods that addresses the loss of accuracy resulting from using such complex elements of the language set; users are able to select an utterance most relevant to their intended message and perform minor alterations to accurately express their thought.

The prototype interface was designed for a target population of adults that are high cognitive functioning, yet also suffer from a motor impairment that affects their communication ability. Additionally, the interface is intended to be used in sequential, goal-directed, public situations with an unfamiliar partner.

3.3 Implementation

The implementation of the software took place in two phases of development. The first phase was a rapid prototyping of the interface using the Python (2.6) programming language and PyGame graphics library; this development environment was chosen to rapidly create and quickly modify the interface. The prototype was used as the demonstration interface to gain feedback from speech pathologists, AAC users, and other researchers involved in interface design to ensure adequate development.

The second phase is a full implementation of the interface using the Java programming language (SE 6.0), incorporating all aspects of the design proposal. The second phase implementation is being developed on a more robust framework to provide a fully functional AAC interface that is comparable to the commercial-

[2] [1] http://www.dynavoxtech.com/products/interaact/

grade interfaces mentioned in Section 3.1. A vital component to the success of this system is to create an intuitive method of adding content for those who frequently work with AAC users (*e.g., speech pathologists*); this procedure is abstracted away from the system's program code to allow anyone to add content. The system content (*e.g., prestored messages, event-based and hierarchical organization*) is tagged in a custom markup language derived from utterance-based and AAC research.

4. AAC INTERFACE PROTOTYPE

Maintaining the design principles enumerated in Section 2.1, the interface must preserve consistency, implement a logical graphical user interface, have uniform functionality, and house an intuitive organization of the information presented to the user. In order to accomplish this, the interface design always has the same general layout; promoting user familiarity and communication ability. The layout will always have a *navigation panel* in a consistent location that is used to move throughout the interface and *navigate* the interface structure, as well as the organized information presented. Similarly, a constant and consistently present display window will show a message once a user has selected it, allowing the user to vocalize, edit, or clear the current message selected. The layout utilizes the remaining real estate of the screen to display the dynamic components of the interface. The consistent components serve as the familiar anchor for the user throughout the interaction with the device. Both the consistent and dynamic portions of the layout use graphical elements that have uniform functionality (a circular button will always perform *x* action, and a square button will always perform *y* action, etc.), this aids in the overall interaction with the interface, as the user will accurately predict functionality (reduces error, increases efficiency). Prior research determined that AAC interfaces that did not maintain a uniform functionality became confusing to use, as functionality became difficult to predict. AAC user input must result in predictable behavior as to minimize instances of correctly entered input resulting in undesirable outcomes. Figures 1 – 6 demonstrate what a typical interaction with this type of interface would look like.

4.1 Utterance-Based Approach

In order to increase the value of user input, the interface prototype employs an utterance-based approach: each input results in an entire utterance being queued for vocalization. The user is able to edit the utterance to more accurately convey their intended message, combine said utterance with another utterance, and vocalize the utterance to produce a spoken version of the message.

4.2 Event-Oriented Organization

Organization of the utterances and information presented to the user is extremely important to the success of the interface. The ability to quickly locate a desired phrase is an essential aspect of a well-designed and efficient AAC interface. Prior research indicates that counterintuitive organization of information and repetition of information presented to the user increase the amount of time needed to locate a the desired information.

The AAC interface prototype implements an event-oriented organization to create an intuitive and hierarchical information structure. By organizing an event into the scenes within an utterance-based system, the real-life actions of the user are reflected within the interface as both draw from the same mental schematization of events.This type of an organization allows the AAC interface to mimic the sequence of events as the user participates in them, thus only utterances relevant to the user's current sequence of events are presented to reduce the need to search for information. This system allows for a natural progression throughout the interface: entire groupings of utterances that correspond to an event – a *scene* – slide out of view to correspond with their completion (*e.g., as the user selects which appetizer they want, the appetizer scene moves out of view*). This progression preserves screen real-estate by ensuring that only relevant utterances are presented, and also automates navigation to reduce the required user input. Additionally, this organization's intuitive and hierarchical information structure reduces the amount of input needed when searching for a message that does not relate to the current schema. Distinct graphical elements are used to represent the schema and to help the user navigate within this organization structure.

4.3 Sequenced Messaging

Automation of user input is a technique frequently used to increase communication rate in AAC interface design. Using the context of the user's current and previous input, the input the user is attempting to generate can be predicted. Such interfaces utilize automation to predict user behavior, preferences, and any other information that requires input in order optimize the rate of communication.

Sequenced messaging is a strategy to increase communication rate, where a user vocalizes a series of sequential phrases that represent each element of a more complex utterance. Some (if not all) of these smaller phrases do not contain adequate information, but would cause the conversation partner to query for more information from the user in a predictable way. Using this system, the ensuing messages can be predicted based on the context of the previous message.

Sequenced messages are used to guide the interaction:

- The first utterance is used to set up the context of the interaction
- The second utterance is the response to the conversation partner's anticipated query to the initial utterance

Rather than take the time to select a message conveying all needed information, the system is set up to allow for quick selection of sequential pairs of utterances that, together, form the intended utterance and effectively convey the user's thought. The overall process spreads the amount of time required to convey a complex message across a series of utterances, optimizing both the speed and accuracy of the overall interaction by dispersing the total response latency of the user without sacrificing the content of the intended message. Such a strategy would be beneficial as sequential messages are categorized as *not enough information*, which is positively rated according to the user studies conducted by Bedrosian et al. The same mental schemas that are used to organize the utterances in the prototype interface's system are also responsible for the logic behind sequential messages: the flawed initial message presented by the user triggers an intuitive recovery response from the conversation partner to mend the conversation by querying for additional information. Sequential messages allow for an overall more interactive experience as the strategy reduces the amount of time the conversation partner spends waiting.

5. FUTURE PLANS AND EVALUATION

The second phase interface is currently in development, focusing on incorporating all of the aspects of the original prototype with their respective feedback; as well as implementing features that have not been completed and features that need attention such as

the integration of message construction, expansion of the message editing system and facilities, and intelligent methods of entering and storing messages and scripts into the system. While the second version of the interface is being worked on, feedback on various aspects of the design is sought out to further the ease of use of the interface. The completion of the full implementation will coincide with a series of planned experiments with AAC users to properly evaluate the interface.

6. REFERENCES

[1] Bedrosian, J., Hoag, L., & McCoy, K. (2003). Relevance and speed of message delivery trade-offs in augmentative and alternative communication. *Journal of Speech, Language, and Hearing Research, 46,* 800-817.

[2] Beukelman, D. R., & Mirenda, P. (2005). *Augmentative and alternative communication: Supporting children and adults with complex communication needs (3rd ed.)*. Baltimore, MD: Paul H. Brookes Pub. Co.

[3] Copestake, A. (1997). Augmented and alternative NLP techniques for augmentative and alternative communication. In ACL-97 workshop on Natural Language Processing for Communication Aids. Madrid, 37–42

[4] Hoag, L. A., Bedrosian, J. L., McCoy, K. F., & Johnson, D. E. (2008). Hierarchy of conversational rule violations involving utterance-based augmentative and alternative communication systems. *Augmentative and Alternative Communication*, 24, 149-161.

[5] Liskov, B., & Guttag, J. (2001). *Program Development in Java. Boston*: Addison-Wesley.

[6] McCoy, K. F., Bedrosian, J. L., and Hoag, L. A. (2010). Implications of Pragmatic and Cognitive Theories on the Design of Utterance-Based AAC Systems. In Proceedings of the First Workshop on Speech and Language Processing for Assistive Technologies (SLPAT), a NAACL 2010 Workshop, Los Angeles, CA, June 2010.

[7] Newell, A., Langer, S., and Hickey, M. (1998). *The role of natural language processing in alternative and augmentative communication*. Natural Language Engineering 4, 1 (March), 1–16.

[8] Norman, D., (2002). *The Design of Everyday Things*. New York: Basic Books.

[9] Schank, R. C. & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Erlbaum.

[10] Todman, J., Alm, N., Higginbotham, J., & File, P. (2008). Whole utterance approaches in AAC. *Augmentative and Alternative Communication*, *24,* 235-254.

[11] Vanderheyden, P. B. (1995). Organization of pre-stored text in alternative and augmentative communication systems: An interactive schema-based approach. Technical Report #AAC9501, Applied Science and Engineering Laboratories, Wilmington, DE.

Figure 1: The user is asked to select a schema that represents his or her current event or intended destination.

Figure 2: The user has entered the *Restaurant* schema in the *Enter* scene and can select utterances from their current or possible future events.

Figure 3: Selecting the utterance *"I would like an appetizer please"* from the *Appetizer* scene automatically advances the schema to present the user with more relevant scenes and utterances.

Figure 4: The user has now vocalized *"I would like an appetizer,"* which is the first utterance in a pair of sequential messages. The second utterance (*"Nachos please"*) can be customized to more effectively obtain the intended goal.

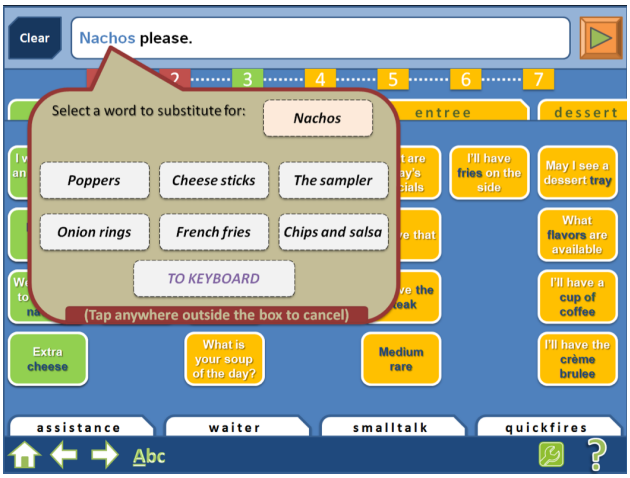

Figure 5: Selecting the *slot-filler* 'Nachos' in the utterance *"Nachos please"* prompts the user to select an appropriate replacement or manually enter a new phrase.

Figure 6: After the user has performed an edit to an existing utterance, the system will ask if this change should be adapted as the default.