Enhancing Alternative and Augmentative Communications Devices with Context Awareness Computing

Adam Loup[1], Lindsay Blue[2] and Shengru Tu[3]
[1]*Geocent, LLC, New Orleans, Louisiana, U.S.A.*
[2]*Hubbard & Tennyson, LLC, New Orleans, Louisiana, U.S.A.*
[3]*Department of Computer Science, University of New Orleans, New Orleans, U.S.A.*

Keywords: Context Awareness, Alternative and Augmentative Communications, AAC Device, Framework, Android.

Abstract: The paper reports the development of a software framework that assists programmers to enhance Alternative and Augmentative Communications (AAC) devices by applying context awareness technologies. Based on the situational contexts including the location and the categories of the conversation, the enhanced system can be highly personalized and can filter, sort the words internally and highlight the words in display. The objective is to lessen the burden of device-aided communication by proactively delivering situational dependent words derived from historical usage contexts.

1 INTRODUCTION

Internet and wireless communications have made it possible for context-aware, human-oriented and proactive systems such as SECE to sense everything and assist everything for our daily work and life (Boyaci, 2012). However, many speech pathologists' patients have not yet benefited to the same extent as other people. For them, carrying out conversations becomes impaired due to illness, aging or injury. Alternative and Augmentative Communications (AAC) devices were created to aid both speech pathologists and their patients. The function of AAC devices is to allow creation of conversation for expressing wants and needs, transfer of information, and social etiquette (Robertson, 2004). Traditionally, AAC devices construct conversation with lexical items available in the device's lexicon which is a predefined core vocabulary developed by Speech Pathologists for general conversations. This core is often lacking words that are specific to a user or a conversation context. In some cases the core vocabulary has been found to contain only 33% of the words for a desired conversation (Patel, 2007).

The goal of the project reported in this paper is to provide a software framework that assists programmers to enhance AAC devices by applying context awareness technologies. Based on the situational contexts including the location and the categories of the conversation, the enhanced lexicon system can filter, sort the words internally and highlight the words in display. The objective is to lessen the burden of device-aided communication by proactively delivering situational dependent words derived from historical usage contexts.

The rest of this paper consists of the following sections. Section 2 describes the AAC devices for speech pathology. Section 3 discusses how the context awareness technology can enhance AAC devices. Section 4 describes the design of the context-aware AAC and the development framework. The implementation and prototyping are reported in Section 5. Section 6 concludes.

2 BACKGROUND: AAC DEVICES

Speech impediments stem from various congenital and degenerative causes. The American Speech and Hearing Association's (ASHA) defines AAC systems as "an integrated group of components, including the symbols, aids, strategies and techniques used by individuals to enhance communication" (Hill, 2010). For those who are unable to express themselves through speech, AAC devices are an assistive technology to aid and enhance communication capabilities.

Loup A., Blue L. and Tu S..
Enhancing Alternative and Augmentative Communications Devices with Context Awareness Computing.
DOI: 10.5220/0004082405000505
In *Proceedings of the 7th International Conference on Software Paradigm Trends* (ICSOFT-2012), pages 500-505
ISBN: 978-989-8565-19-8

There are two main types of AAC, unaided and aided. We will focus on the aided AAC which refers to approaches that rely on additional peripherals that render representations of what the user wants to convey. Aided devices also include digital devices that playback recorded or synthetically created speech. To date, most effective means of language representation in aided AAC devices has been accomplished by presenting the user with alphabet-based symbols. Access to individual words, through spontaneous novel utterance generation, has been proven to increase participation in casual conversation and to promote natural language development (Hill, 2010). Pre-stored messages or phrases rarely meet the needs of conversing in the natural environment and often fail to give the user adequate conversational ability (Patel, 2007).

The choice of vocabulary available to the user is a critical aspect to the success of AAC usefulness. There are two main divisions of vocabulary, core and extended (Robertson, 2004 & Hill, 2010). The core vocabulary is the few hundred words that speech pathology research has deemed critical to create general conversation and the majority of social interactions. Extended vocabularies are those words which are used to describe specific items and are used infrequently. Together, these two categories provide a solid foundation for improvements to AAC.

3 AAC DEVICES NEED CONTEXT AWARENESS

The current AAC devices provide a closed and rigid lexicon. Due to cost considerations, the patients themselves or family members rather than speech pathologists are responsible for maintaining the device. While these users are allowed to adjust the words presenting on a device, they often lack the expertise for systematically choosing words. There is a strong need for fine-grained, intelligent personalization for AAC devices.

Users of AAC devices are often bound to a limited number of conversational contexts. Meanings expressed through conversations are highly dependent upon the context in which it is created. Some existing AAC devices divide words displayed into categories of semantic frames that the user chooses for the desired conversations (Robertson, 2004). We have pushed this idea further in our enhanced AAC device prototypes. First, the category choice is easily made from a panel which can be called out by one button press. Second, upon a choice of the categories the whole screen displays the words for the chosen category. Third, new categories can be created by the patient or the family members.

Our AAC devices constantly collect the word usage statistics. The user’s vocabulary size can be estimated by the number of different words used by the user and the new words added to the lexicon by the user. It is reasonable to shrink the display list for the user with a small vocabulary. The statistics about the usage of each word in different categories are used to form the collections for each category.

While conversation category is chosen by the users, the enhanced AAC device can predict possible category change by sense the location change of the device, assuming the user is carrying the device. When the user moves from one place to another place, he/she would likely engaged in a different conversation. Therefore the AAC device prompts the category change panel to the user. The device records the user’s category choice into the location-category pattern table. In general, conversations of multiple categories can happen at a location. In the category-choice panel, the buttons of the most frequently chosen categories at the current location will be highlighted. If the usage records show an adequately strong one-to-one bound between a location and a category, the AAC device can automatically switch to the category.

For a set of user-selected locations, the enhanced AAC device records the frequency of the words used at these locations. As a result, each location may have a subset of words for some categories. For example, the kitchen area will bound to the food category and having a subset of words about hot meals in the food category. The daily diet of the user would determine the frequently used words.

When the AAC device senses the user’s movement, a threshold of the moving distance is used to trigger category choice promotion. The threshold value for each user selected location is given an initial value. For instance, the threshold for a park area would be 200 meters; the threshold for home could be ten meters only.

Other statistics, such as the user’s speed of word selection and his speed of page flipping can also be used to adjust the AAC device’s performance. These two measures are often closely correlated and indicate the user’s overall communication pace. For fast users, more words can be displayed in the device.

Finally, the sentence completion speed is the measurement of the user performance. The enhanced

AAC device will constantly monitor this measurement for each conversation category. As words in a layout can be organized in various fashions decided by the designer, the methods of organizing words for layouts vary greatly. Using user's sentence completion speed, the effectiveness of any adjustments or enhancements to the device can be quantitatively measured. Any deceleration will indicate an adverse modification.

4 DESIGN

In this section, we will first illustrate the user interface of our AAC device. Then we will describe the architecture of the context-aware AAC system as well as its key components. As we discuss the decision options of this system, we advocate the importance of the flexibility in design. Finally, we specify a framework that will facilitate developers with flexibility in reusing both proven design and tested components for future AAC development by deploying techniques based on context awareness.

4.1 AAC User Interface

Figure 1 illustrates the upper portion of the touch-screen user interface of our prototype of the context aware AAC device. Nearly the entire screen is the word display area. Common among tablet devices, the user can scroll up and down in the word display area by scrubbing the screen. The user makes word selection by touching it. The selected words will be displayed in the top text line in the order of selection. The AAC device will speak the words in the top text line when the user presses the "say" button on the right.

The button at the upper right corner allows the user to switch modes such as main (as shown in Figure 1), library manager (to import/export categories) and configuration (to set up the home location and create interested locations). The drop-down menu next to it (displaying "core" in Figure 1) is for the user to select conversation categories.

The word display area is divided into two parts. The three columns on left make the "stop words" area. Stop words consists of articles, pronouns, interjections, and other function words that serve to create grammatical structures but are irrelevant to conversational category (Porter, 2001). The remaining area (the seven columns on right) is the main word display area. Only the words displayed in this main area will vary according to the conversation category and the current context. However, the word selections in both areas will be affected by the general context information such as the user's vocabulary size.

4.2 AAC System Architecture

Our enhanced AAC system is compartmentalized into components: *Lexicon*, *Operation Logger*, *Word Provider*, *Phrase Analyzer*, and *Lexicon Database* as shown in Figure 2. In this system, the *Lexicon* takes the central role and provides services that collect, organize and maintain words available to the user by leveraging the capabilities of the other components. The *Lexicon Database* stores the collected data and supports queries. The *Operation Logger* monitors and records the user's operation speed and sentence completion speed. The operation speed is the average time interval between two word selections. Note, the time interval from pressing the "say" button to the first word selection is excluded in measurements because it is often the user's listening period. The sentence completion speed is the average duration of each spoken phrase or sentence, which is estimated by measuring the duration from the first word selection after the "say" button was pressed to the next press on the "say" button.

Context awareness is achieved by leveraging multiple contexts into a singular situational context, the *LexicalContext*. Simple context data such as geographic location is combined with inferred context information consisting of the actual words spoken and the categories currently used (Park, 2005). Historical contextual information has a significant role in personalization. This is collected by the *PhraseAnalyzer* that filters, analyzes and records the word usage along with the context information into the lexicon database.

4.3 Lexicon Management

The Lexicon component is in charge of presenting the word set relevant to the user's current situational context. The Lexicon obtains words by querying every available WordProvider component using the current location and category. When a change of category or location happens, the lexicon immediately sends a query to every available WordProvider component with the changed parameters. The lexicon combines and sorts the WordProviderResult collections for display to the user as each WordProvider component returns results.

Figure 1: The graphical user interface of the AAC prototype.

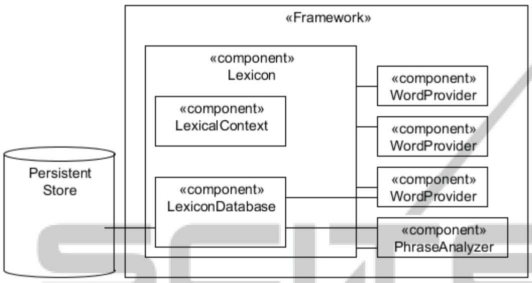

Figure 2: The AAC Architecture.

The lexicon is also responsible for dispatching the tasks of recording word usage and context information. Every time when the user completes a phrase or sentence, the Lexicon passes the phrase to the *PhraseAnalyser* component. The Phrase-Analyser component breaks down the phrase into individual words. First, any stop words are removed. Next, word objects are created for any words new to the current word set. New words may not be base words. Identifying the base of a word is carried out with a stemming algorithm, the Porter-Stemming algorithm, which is implemented and available in the Lucene Snowball API. If the base word does not exist in the Lexicon, a Word object for the base word is created. Any new Word objects created by the *PhraseAnalyser* are added to the lexicon database through the persistence layer. Finally, a record about the usage context is created for each word and category pair along with the current geographic location. These records are then stored in the lexicon database. Since one word can have multiple conjugations, the base word's ID is used in place of the word ID. If a record corresponding to the word-category-location combination already exists in the database, the usage counter is incremented. Word usage is kept because the lexicon tries to highlight the words most relevant to the user's current context. Technically, relevance refers to how often a word is used in a context.

Initially the stop word set is created from a predefined, common list used in all natural language processing. To promote personalization, words can be added to the stop word set. For example, if the word usage statistics show that a (non-stop) word is unanimously used in all categories and locations repeatedly, it can be added to the stop word set and to be displayed all the time.

4.4 Word Providers

Word Providers are used to populate the lexicon with words and to discover meaningful lexical items. They dictate which words go to the lexicon at a given time. Word Providers deliver words to the lexicon as collections of *WordProviderResult* objects. Each such object contains the word and a relevance score representing a weighted importance of this word to the chosen context.

In our AAC architecture, WordProvider components can be replaced. We have implemented two WordProvider components, namely the *LocationWordProvider* component and the *LocalWordProvider* component. The *LocationWordProvider* returns relevant word results based on the current geographical location of the user. This is achieved through the usage history data. The importance of a word to a particular geographic location is determined by using term frequency normalization. This relevance is computed by term frequency times inverse document frequency (*tf***idf*) (Manning, 2008). For purposes of this provider, term frequency (*tf*) is the normalized frequency in which a word has been used in a situational context and is computed as $\mathrm{tf} = \sqrt{(\frac{\mathrm{ct_t}}{\mathrm{ct_c}})}$ where $\mathrm{ct_c}$ is the number of times a word has been used in all contexts and $\mathrm{ct_t}$ is the total number of times a word has been used in the current context. Document frequency (*df*) is the number of categories, per geographic location, associated with the word divided by the total number of categories (n). The Inverse Document Frequency (*idf*) is then computed as $\log\left(\frac{n}{df+1}\right) + 1$. *LocationWordProvider* uses the result of *tf***idf* for each term as the relevance score for the *WordProviderResult* of the term. Computing the relevance using IDF ensures that a lower importance is given to words that occur greatly throughout all situational contexts. A higher score is given to words that are used more often in a particular situation. The more a word is used in a specific situation the higher the relevance score becomes.

Words stored locally are also accessed through a word provider. The *LocalWordProvider* returns words that are most relevant based on the predetermined categories stored in the word_category table. These relationships serve as a starting point until the usage history is capable of supplying valid results independently.

4.5 AAC Development Framework

The framework provides two basic implementations of the word provider class, the basic *WordProvider* and the cache based *CachedWordProvider*. These two classes are meant to be extended in order to create customized Word Providers. For any provider that extends *WordProvider*, the *getWords* method is called every time the lexicon requests words. Conversely, for any word provider that extends the *CachedProvider* the *getWords* method is only called when a new context passed to the provider or if the results of a previous request have expired.

WordProvider component can be replaced. Each word provider can have a different criterion to grade the relevance score. For example, the correlation between locations and words was used in the LocationWordProvider but the correlation between conversational categories and the words was used in the LocalWordProvider. We have made the AAC development framework that supplies an API to support the creation of customized word providers. The framework allows the AAC developers to customize the methods in which word relevance is computed. The framework is intended to be a foundation to create applications on mobile devices and is designed to be lightweight with a low resource footprint.

To use this API, the programmer needs to register to the lexicon through the *addProvider* and *removeProvider* methods. Custom word providers can extend the *WordProvider* class and implement the *get* method to return the *WordProviderResult* collection. Every registered provider will be invoked by the lexicon in response to a change of context, and the context is passed to all registered word providers. The combination of the capabilities of the enhanced AAC and the flexibility provided by the framework will assist AAC developers to deploy context awareness technologies to this type of forgotten application.

5 IMPLEMENTATION AND PROTOTYPE

Implementation of the framework is done in Java. All components are compatible with the Standard Java 6.0 run-time environment. The framework was developed with the Android platform in mind but did not use any Android specific functionality.

A SQLite database and a Java data access layer (ORMLite 4.33) are used in the persistence layer. Access to the database from the components of the framework is handled through the *LexiconDatabase* object. This object is a wrapper for an external persistent data store that contains methods to obtain the Data Access Objects (DAO) for words, categories, word categories and word usages. Each DAO manages the persistence and retrieval of the respective type.

A practical AAC system has been built using our framework. This prototype runs on a touch screen tablet running the Android operating system. The main focus of the interface is to utilize the screen space to the full extent by displaying the most important elements to the user: words, and to simplify the controls for speech creation. Using the built-in WordProvider components, two sources are connected to create word providers. A base lexicon is established by importing libraries that contain the desired categories.

In order to build a useful profile large amount of usage data are needed. We collected and parsed various movie and comic book reviews into the lexicon through the *PhraseAnalyzer*. The *LexicalContexts* with predefined categories and geographical locations were created for this test. After the data were imported, the *PhraseAnalyzer* succeeded in both adding new words and creating word usage records. Words were requested from the *LocationWordProvider* using the same context passed to the *PhraseAnalyzer*, the top ten results are shown in Table 1.

Table 1: WordProviderResult: word, relevant score.

Word	Relevance Score
movie	0.83827
out	0.07824
character	0.07662
way	0.074813
world	0.069434
grey	0.068503
up	0.063997
film	0.062720
Action	0.062194
new	0.062194

Filtering the results by relevance scores was able to determine that the word "movie" was most relevance to the user in this contest. Sorting by frequency alone is not adequate in determining meaningful relevance. The *LocationWordProvider* calculates the relevance of a word to a specific context. Based on frequency alone "movie" was the twenty first ranked value.

6 RELATED WORK

Context Awareness Computing (CAC) has attracted significant efforts from academic research and electronic industry (Mehra, 2012; Yonezawa, 2011). The CAC programming language, such as SECE. connects previously isolated services into user personalized composite services (Boyaci, 2012).

A project about applying CAC to health care was reported in (Kjeldskov, 2007). The goal was to maintain a distributed electronic patient record across multiple ubiquitous devices both mobile and stationed throughout large hospitals.

The work closely related to this paper includes those on user-centric systems such as the pervasive environments in (Kalatzix, 2008; Preuveneers, 2007). The framework proposed in (Park, 2005) provided us with a reference model of complex CAC system and a number of design options.

7 CONCLUSIONS

Due to the complexity of natural language, no one best method will determine the conversational relevance of words in any situational contexts. Separating the Word Provider component from the Lexicon gives the flexibility to achieve a level of personalization that many current AAC devices do not provide. Applying varying methods of relevance computation and word discovery with a predefined organization of words allows conversations that may have previously been unavailable to the user.

The AAC device prototype needs thorough user evaluation. Future work on the framework will include further decoupling of the components that make up the Lexicon and adding means for more customization. Future work will also introduce device discovery capabilities to allow data sharing between users. Meanwhile, the AAC devices will be able to automatically sense people grouping context for further customizing available vocabulary.

ACKNOWLEDGEMENTS

Shengru Tu's work was partially supported by NSF grant CCF-0939108.

REFERENCES

Boyaci, O., Martinez, V. B. and Schulzrinne, H. (2012): Bridging Communications and the Physical World. IEEE Internet Computing, March 2012, 35-43.

Hill, K. (2010): Advances in Augmentative and alternative communication as quality of life technology. Physical Medicine and Rehabilitation Clinics of North America, 21, 43-58.

Kalatzix, N., Roussaki, I. and Pits, C. (2008): User-centric Inference based on History of Context Data in Pervasive Environments, 3rd International Workshop on Services Integration in Pervasive Environments (SIPE'08), 25-30.

Kjeldskov, J. and Skov, M. B. (2007): Exploring context-awareness for ubiquitous computing in the healthcare domain. Personal and Ubiquitous Computing archive, 11(7), Springer-Verlag London, UK.

Manning, C. D., Raghavan, P. and Schtze, H. (2008): Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

Mehra, P. (2012): Context-Aware Computing – Beyond Search and Location-Based Services. IEEE Internet Computing, March 2012, 12-16.

Park, H. and Lee, J. (2005): A Framework of Context-awareness for Ubiquitous Computing Middlewares. 4th Annual ACIS International Conference on Computer and Information Science (ICIS 2005). IEEE Computer Society, Jeju Island, South Korea, 369-374.

Patel, R. & Radhakrishnan, R. (2007): Enhancing Access to Situational Vocabulary by Leveraging Geographic Context. Assistive Technology Outcomes and Benefits, 4(1), 99-114.

Porter, M. F. (2001): Snowball: A language for stemming algorithms, http://www.cs.odu.edu/~jbollen/IR04/readings/readings5.pdf

Preuveneers, D. and Berbers, Y. (2007): Towards Context-aware and Resource-DrivenSelf-adaptation for Mobile Handheld Applications. 2007 ACM symposium on Applied computing, Seoul, Korea, 1165-1170.

Robertson, S. (2004): Understanding Inverse Document Frequency:On theoretical arguments for IDF. Journal of Documentation, 60(5), 503–520.

Yonezawa, T. and Dargle, W. (2011): CASEMANS Workshop Summary. 13th International Conference on Ubiquitous Computing, Beijing, China, 623-624.