MIKI: A Speech Enabled Intelligent Kiosk

Lee McCauley and Sidney D'Mello

Department of Computer Science, The University of Memphis
Memphis, TN 38152, USA
{mccauley, sdmello}@memphis.edu

Abstract. We introduce MIKI, a three-dimensional, directory assistance-type digital persona displayed on a prominently-positioned 50 inch plasma unit housed at the FedEx Institute of Technology at the University of Memphis. MIKI, which stands for Memphis Intelligent Kiosk Initiative, guides students, faculty and visitors through the Institute's maze of classrooms, labs, lecture halls and offices through graphically-rich, multidimensional, interactive, touch and voice sensitive digital content. MIKI differs from other intelligent kiosk systems by its advanced natural language understanding capabilities that provide it with the ability to answer informal verbal queries without the need for rigorous phraseology. This paper describes, in general, the design, implementation, and observations of visitor reactions to the Intelligent Kiosk.

1 Introduction

As we find ourselves at the height of the information age the need for user-friendly, naturalistic, and intuitive information systems becomes paramount. Stephandis et al. have predicted that public information systems, terminals, and information appliances will be increasingly used in a variety of domains [1]. Of particular interest in the field of intelligent information systems and virtual agents are information kiosks. These are a special variant of information appliances that are usually deployed in public locations such as transportation hubs, malls, businesses, etc. In addition to the basic issues that accompany the design of typical information systems such as information retrieval, multi-modal communication, and interface design these systems pose some novel and interesting concerns. Cassell et al., point out that kiosk systems differ from traditional systems in that they should stand out so that they are noticed by visitors, their functions should be self-evident, no user training should be required, and they should be able to recover from user errors [2].

One of the information kiosks that demonstrated a significant improvement over earlier systems is the MINNELLI system [3]. MINNELLI facilitates interactions with bank customers primarily by the use of short animated cartoons to present information on bank services. However, the MINNELLI system requires basic user training which reduces its applicability in most public sites. Another successful kiosk with a broader scope than the MINELLI system is the MACK system [2]. MACK is an embodied conversational kiosk that provides information on residents and directions to locations at a research site. It integrates multiple input sources that include speech, gesture, and pressure. The system also exhibits a degree of spatial intelligence by utilizing its awareness of its location and the layout of the building to reference physical locations

J. Gratch et al. (Eds.): IVA 2006, LNAI 4133, pp. 132 – 144, 2006.

when it provides directions [4]. The August spoken dialog system is also kiosk based and helps users find their way around Stockholm, Germany using an on-screen street map. August is designed to elicit a conversation from the user and facilitates the study of such interactions [5, 6].

This paper describes the design, implementation, and initial user reactions of one such intelligent kiosk system called MIKI: The Memphis Intelligent Kiosk Initiative. The system is deployed on a plasma screen at the FedEx Institute of Technology, a building that houses a community of interdisciplinary researchers, at the University of Memphis. As a person approaches the display, MIKI greets them, introduces itself, and offers to be of assistance. The individual can then verbally ask a question related to any of the following topics: (1) events at the FedEx Institute of Technology, (2) research groups housed at the Institute, (3) directions to rooms within the building, (4) people involved in research at the Institute. In answer to the visitor's question, the kiosk responds in a number of different ways. The response might include a verbal answer along with 3-D animations, video presentations, images, or additional audio. Along with the prototype kiosk (Figure 1), tools were created that allow for the maintenance and timely update of information presented by the kiosk.

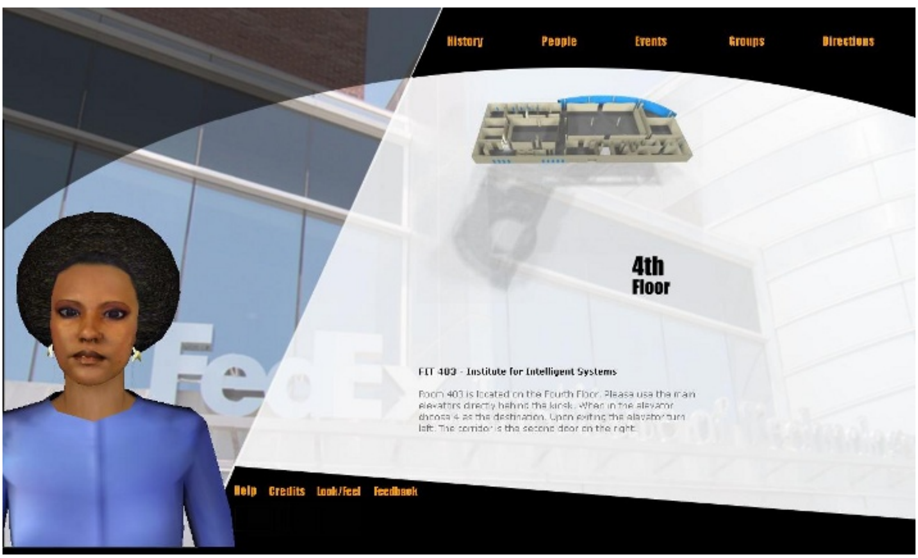

Fig. 1. MIKI: The Memphis Intelligent Kiosk Initiative

MIKI shares several similarities to the MACK system in that both systems use multiple input channels and that they both provide information on people and groups and directions to locations at a research site [2, 4]. However, the MACK system relies on rule based grammars alone for speech input. This greatly restricts the scope of questions with which a user can query the system. MIKI is equipped with a standard grammar as well as a statistical based natural language understanding mechanism that reduces the need for rigorous phraseology in the information requests a user presents to the system. We identify this facet of the Intelligent Kiosk as the paramount factor that distinguishes it from other such systems.

The intelligent kiosk, itself, is a collection of different technologies that are all integrated and work seamlessly together. Among these different technologies is video processing for face detection, a digital avatar, speaker-independent speech recognition, an advanced graphical user interface (GUI), an array microphone for noise cancellation, a database system, a dynamic question answering system, and a cutting-edge touch panel technology for large displays. We proceed by describing the primary components of the system followed by some of the technical challenges encountered. We then present some anecdotal accounts of humans interacting with the system and provide details on some usability studies that are underway.

2 Primary Components

There are several components that comprise the Intelligent Kiosk. A general layout of the major software elements is presented in Figure 2.

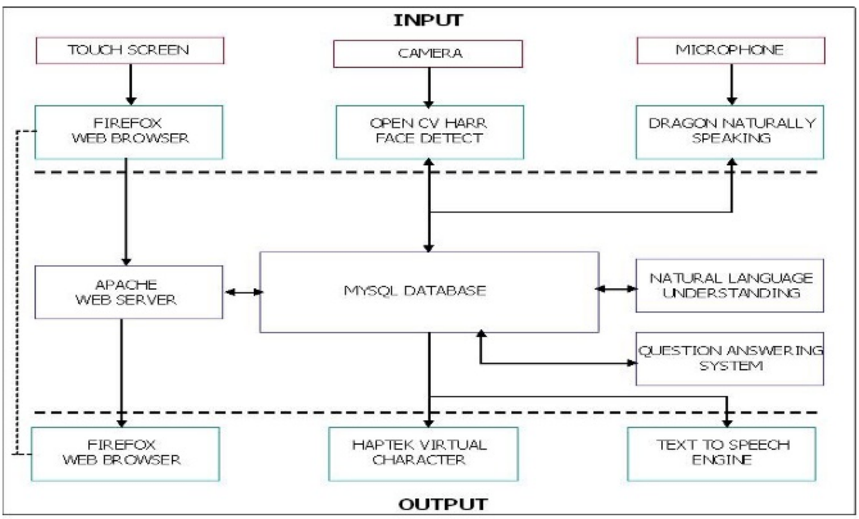

Fig. 2. Major software components of the intelligent kiosk

2.1 The Hardware

The Intelligent Kiosk resides in the lobby of the FedEx Institute of technology on a small wall facing the main entrance. The location was chosen for its optimal visibility to outside visitors. As a visitor enters the building, they see a 50 inch display surrounded by custom-made cabinet. The 50 inch Panasonic® display has been augmented with a touch panel overlay provided by Smart Technologies®. Mounted above the display is a small, FireWire web cam. Just below the display is a somewhat larger Acoustic Magic® array microphone. Both the camera and the array microphone are angled for optimal function for someone standing between 1 and 3 feet in front of the display. Inside the lower part of the cabinet are two Dell® workstations, an Ethernet hub, a KVM switch, and a wireless keyboard and mouse. The two systems are basically identical with 3.2 GHz CPUs, 128 MB AGP video cards, and 2 GB of memory. Both are running windows XP as their operating system. Figure 3 depicts the major hardware components of the Intelligent Kiosk.

Fig. 3. Major hardware components of the intelligent kiosk

2.2 Speech Recognition

For seamless verbal interaction with visitors seeking information a speaker independent speech recognition system was required. We decided to use the commercial available product Dragon NaturallySpeaking developed by Nuance® although it was not designed for speaker independence. This decision was motivated by the fact that we were in the possession of working software to interface with this speech recognition engine through CloudGarden's JSAPI implementation. Therefore, even though we looked into a few other speaker independent speech recognition engines, such as CMU Sphinx, we decided that we would stick with Dragon NaturallySpeaking version 8 primarily due to time constraints. By carefully restricting the language model for speech recognition we were able to simulate a reasonable quality of speaker independent recognition (more details provided below).

2.3 Natural Language Understanding

The natural language understanding module has one responsibility, to provide an analysis of the user's utterance in order to determine what action or actions need to be taken. When a system is employed in a limited domain such as the Intelligent Kiosk and has only a limited number of choices of what to say or do next, it need only classify a visitor's utterance rather than completely comprehend every word. MIKI uses two different NLU technologies that include simple keyword matching and a classification technique based on Latent Semantic Analysis [7, 8]. LSA is a statistical technique that measures the conceptual similarity of two text sources. LSA computes a geometric cosine (ranging from -1 to 1) that represents the conceptual similarity between the two text sources.

Classification approaches to NLU include statistical [e.g.,9, 10], information retrieval [e.g.,11, 12] and connectionist approaches [e.g.,13]. Notable among these approaches are those that are based on word co-occurrence patterns such as LSA [7, 8] and HAL [14]. LSA is an attractive approach because it can be trained on untagged texts and is a simple extension to keyword based techniques.

In general, methods such as LSA, involve the analysis of a large corpus of text in order to create a semantic representation of each word. We then use these semantic representations to categorize incoming utterances as one of the existing grammar rules. There has been research conducted on the application of this type of corpus analysis to speech recognition, although it is generally assumed that the language models produced would replace or modify those in existing speech recognition engines [15, 16]. A simplified version of the algorithm used for natural language understanding in conjunction with LSA for verbal input is presented in Figure 4. More details on the algorithm and preliminary performance analyses on semantic, feature, and gibberish instances on queries relating to appointments in the Microsoft™ Outlook's Calendar program can be found in [17] and [18].

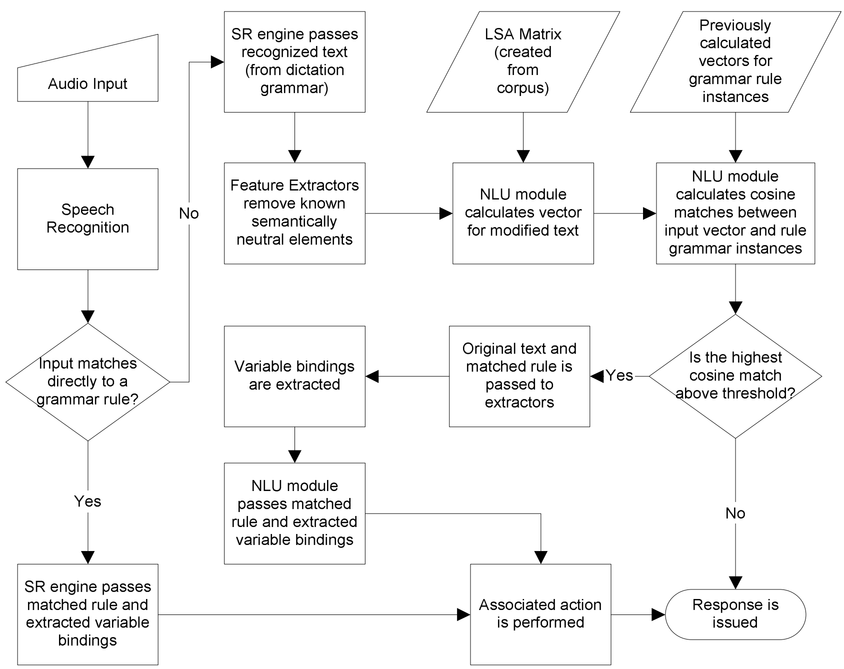

Fig. 4. Algorithm for natural language understanding

2.4 People Tracking

Rather than relying on sophisticated trackers, such as the Kalman Filter [19], a rather simple tracker that relied on face detection was used. The general object detection algorithm consisting of a cascade of boosted classifiers proposed by Viola and Jones [20] was used to detect faces on the basis of Harr-like features. SharperCV, a C# wrapper for the Intel® OpenCV library [21], was used for all video processing. This use of the face detection algorithm provides the ability to not only recognize when someone is approaching the kiosk, but also to count the number of individuals that interacted with the system as opposed to those that walked by. The tracker was based on the assumption that with a large frame rate (30 frames per second) the change in the location of a face from one frame to the next is rather minute. Therefore, a face in the current frame was assigned to the nearest face (based on Euclidian distance) among all the faces in the previous frame as long as the distance was not greater than some predefined maximum threshold.

2.5 Graphical Interface

The graphical interface is a combination of two technologies: a standard web-based front end that is the primary interface and a secondary animated character.

2.5.1 Primary Interface

The primary way that individuals can interact with the kiosk is through a web-based front end. It uses a combination of PHP, HTML, and Flash. The PHP and HTML are used to render dynamic pages based on the data stored in the database. The visual transitions between screens as well as direction animations are provided via several Flash scripts. Of particular interest are the animations for directions to various locations in the building. They were generated using the original CAD files of the building. When a user asks for directions to a given room that flash file is loaded. A small ball is illuminated where the user currently is and is then moved along the path that will get that person to their destination. As the floating ball changes floors, that floor becomes visible and the others fade away. In addition to the animation, the directions are also spoken through the avatar and provided in text below the animation.

2.5.2 The Avatar

The animated avatar was created using Haptek's® PeoplePutty software and displayed with the Haptek Player. The programmatic interface through an ActiveX control is fairly straightforward and creates a realistic animated character. The player is free for distribution and has a small footprint both in respect to memory and CPU resources. Using the PeoplePutty software, we were able to create two avatars based on the two students that also provide the voice for their characters. The result is two characters, one male and one female, that look and sound surprisingly like their human models. Despite the fact that the system has a fully functional text-to-speech (TTS) system as shown in Figure 2, it was noted that most of the vocal interaction from the agent would be known a priori. As this suggests, all of the vocal output produced by the avatar was prerecorded by the human models. These recordings were then processed using a Haptek tool that tags the recordings with mouth movements.

When these recordings are played back through the animated character, the mouth animations of the character match the spoken utterances.

2.6 Question Answering

Question answering was accomplished through a combination of mechanisms but primarily involves a naïve use of the database to disambiguate between a number of possible questions being asked. First, each possible type of response is described through an XML file. Because the vocal interaction with the kiosk is intended to be completely open-ended, it is not possible to define each and every question that might be posed to the system. Therefore, each type of response defined in the XML file is defined primarily by what screen services that response requires. For example, all questions regarding general information about an individual will be answered by displaying a detailed personnel page that is dynamically generated based on information from the database. A single element in the XML file describes how to recognize and respond to all inquiries of this type. Each response element then provides the following types of information: (1) the frame name (screen), (2) a key field in the database, (3) a template for an SQL command, (4) text features (as extracted by the speech recognition and/or NLU module) used to match this response to a question, (5) follow-up question types used to disambiguate between several possible answers, (6) a template for a textual response assuming a positive answer was located, and (7) a template for a textual response assuming a positive answer could not be located. With the current system using prerecorded vocal responses, items 6 and 7 above are not used.

The first two items, the frame name and the key field, are used to communicate with the graphical front end. The frame name informs the GUI what screen to display while the ID field contains specific information about which record to display on that screen. For example, a visitor might say, "can you tell me about John Doe?" The speech recognition/NLU system categorizes this utterance as a request for specific information about John Doe. The question answering system picks this up, matches the question and information provided to a specific response type, and updates a query tracking object with the information provided. It and then binds the variables in the SQL template for that response and runs the database query. Assuming that only one record comes back matching that query, the value in the key field is extracted and a message is sent to the GUI that includes the frame name and the value of the key field for that person's record. The GUI then handles displaying the information on the appropriate screen as well as issuing any messages for actions through the animated avatar.

A distinction should be made between this method and the widely used Artificial Intelligence Mark-up Language (AIML) [22]. Many "chatbots" have been created using AIML as the question-to-answer connection engine. This method essentially maps the text of a question to an appropriate response. What makes this system powerful is its ability to decide between multiple possible answers based on the closeness of the match. MIKI, on the other hand, uses the results of the database search to dictate the final response. As a side note, AIML has been integrated into MIKI as a way of dealing with questions that are not part of the primary database that deals only with information about the FedEx Institute of Technology.

3 Technical Challenges

The following section describes the challenges faced during the implementation of the Intelligent Kiosk. Solutions are also presented although some components are still under refinement now that the system is in place.

3.1 Component Communication

The various pieces of the kiosk communicate through the database. A table in the database, called the "messages" table, holds all information that is being transmitted between components. The table holds some basic information about each message such as an auto-incremented identifier, the sender, the recipient, the message tag (e.g., LOAD, SPEAK), text data (e.g., a recognized utterance), binary data (e.g., a Hashtable mapping features to values, *first-name=John*), and a timestamp. Any number of components can access the messages table and effectively react to the messages. All they need to know is the tag or tags in which they are interested. This is similar to a shared memory framework.

Using a database as the focus of a communication scheme between components has several advantages. For example, a simple ontology can be used to address senders and information types. This allows the system to be implemented quickly without confusion even among a dispersed group of developers. Adding new resources simply involves a name assignment and the definition of new message tags. Even so, this technique is not significantly different from a standard blackboard model. A central component polls the database on a regular basis and messages are distributed to the appropriate modules. Instead of using specific techniques, such as Galaxy Communicator [23] or other similar systems, it was decided that maximum interoperability would be gained through the use of a method that was common to all of the disparate languages. Even the web-based front end could easily make use of an industry standard database. However, this does pose a problem in that the web component has to poll the database independently. As long as the table of messages remains small, this is a minor issue. When the table becomes large, agent and interface response becomes slow. For this reason, daily maintenance is performed on the database that backs up the table of messages into an archive and erases old messages.

The communication scheme is persistent-asynchronous in nature. Message persistence is provided by the messages table in the database. The system is asynchronous because components read and write messages on their own schedules independent of other components. This communication scheme alleviates several problems associated with distributed components. The only thing that each component needs to have a connection to is the database. This is a very simple and well tested procedure that is not specific to a particular language. For instance, the graphical front end is a combination of PHP code running some Macromedia Flash displayed in a Mozilla Firefox web browser. The PHP code has no problems sending or receiving messages from other components to tell it to change screens or display a particular frame. Much of the "back-end" code is written in Java while the virtual agent control software and the tracker is written in C#. Each choice of language for a given component was made based on what core functions were most important and which language supported those functions best. Getting these very different components to talk to each other using some other method like CORBA or SOAP would have been problematic at best and would not have provided any benefit over the database solution. Finally, the database solution already incorporates speed optimizations for the transfer, storage, and logging of data. Logging of the system's internal workings is a simple matter of backing up the messages table.

The use of a database as the medium for sending and delivering messages between components is the central idea that makes the rest of the distributed framework almost trivial. Once the database was in place as the message delivery system, all the components were able to use any method they chose to access from that database. Each component was then designed as a stand-alone process. In addition to the fact that any component could then be run on any accessible computer, this also facilitated easy testing of components in pseudo-isolation. Testing code manually created messages in the database and then read the resulting messages entered by the component being tested to determine if the test passed or failed.

For the Java and C# components, a generic API to facilitate communication with the database was created to allow for easy implementation of various components. The API contained data structures and algorithms to ensure optimal transparency related to the encoding and decoding of messages to and from the database. Such a system could be said to have scale-up problems if the number of separate components were to become excessively large, but for the foreseeable future of less than one hundred separate modules this system should not have a noticeable degradation in performance.

3.2 Speaker Independent Speech Recognition

The commercially available software package Dragon NaturallySpeaking from Nuance® was used for speech recognition. Unfortunately, Dragon NaturallySpeaking was designed for non-speaker independent use. The major motivation for the use of this recognizer over more viable options such as CMU Sphinx was the existence of legacy software using Dragon and a JSAPI implementation from CloudGarden. In order to achieve an acceptable degree of speaker independence we used an untrained speaker profile and replaced the default language model with a much smaller model that consisted of common words that would likely be used in the kiosk domain. To this we added the names of individuals, groups, and some events associated with the FedEx Institute. In other words, the content of the kiosk was used as the majority of the words that the speech recognizer would handle. Additionally, the process of updating the language model was automated by periodically dumping relevant tables (e.g., groups, events, etc) from the database and utilizing the language building tool available as part of the Dragon NaturallySpeaking software suite. Therefore, as people, groups, and events are added, deleted, and modified, the language model is guaranteed to stay consistent.

This method of utilizing a very restrictive language model worked better than was originally expected, but is still unsatisfactory from a performance perspective. Studies are currently under way to determine how well this method performs compared to other methods and recognizers. It is our expectation that we will discover that this method only performs at around 60% accuracy, but that is purely speculation. Ultimately, we are sure that another method or recognition system will be used with better success.

3.3 Session Maintenance

One important issue is how and when to start, end, and maintain a session. We would like the kiosk to proactively initiate interaction whenever a person is looking at the screen. The vision component identifies and tracks faces. Based on the size of the face, an approximate distance from the kiosk can be determined. If the face is within about 4 feet of the kiosk, then it is assumed that the person is within the range of interaction. This is used to open a session if one is not currently open, or maintain a currently open session. Voice input and touching of the screen also triggers an opening or maintaining of a session. Finally, sessions are kept open for a few seconds (currently 30 seconds) even if no face is visible and no other input is received. We have found that this works pretty well. It does not close a session prematurely or keep open a session beyond a reasonable length of time.

4 The Human Component

The purpose of the Intelligent Kiosk is, of course, human interaction. Studies are currently being conducted to determine the general usability of the kiosk along with the number of visitors that choose to use the kiosk compared to the human staffed information pavilion. An unobtrusive, non-reactive, observational study that counts the number of people who approach and use the kiosk as well records details of their interactions is underway. On the basis of the schedule of events at the Institute we determined two days on which we would conduct the observations. These include a typical "slow" day when little or no outside visitation would be expected and a typical "busy" day. The qualitative studies are currently underway, therefore, what follows is anecdotal evidence based on informal observation.

4.1 Visitors' Reactions

The types of interactions that we have observed have been somewhat different than what we expected. First, many of the visitors that have heard a description of the kiosk assume that it will engage them in some form of conversation. Consequently, the first words often spoken to the kiosk are, "hello" or "hey there." Since this was not part of the kiosk's repertoire, the visitors are left not sure what to do next. Other casual users, those that just see the system for the first time and approach it, most often just stand in front of it and do not seem to be sure whether they are allowed to touch it and unaware that they can speak to it. This has revealed one of the primary limitations of the kiosk in its current form, namely that there are not enough cues provided to the casual observer as to what the kiosk is and how to interact with it. This is especially acute when the kiosk is in screen-saver mode. During these times the system is displaying a video about the FedEx Institute and does not look that much different from any of the other plasma screens in the building.

When the kiosk is not in screen-saver mode, the virtual agent has a distinct look that draws some attention. We believe that it is the very presence of this human-like face that causes some people to try to converse naturally with the kiosk. On one occasion, a visitor walking past paused long enough to look at the agent and wave. Needless to say, the kiosk did not wave back.

4.2 The Casual Hacker

One other class of visitor to the kiosk seems to think that it is there duty to show us where the system can be hacked. During the implementation of the kiosk, we did not devote any serious effort to designing security measures for the physically present miscreant. We did put in place several layers of digital security. Nevertheless, we have had several attempts by individuals standing in front of the kiosk to "break" it. The system is running on Windows XP primarily through an Internet browser. This means that we get all of the drawbacks of these two technologies even though we have some advantages. First, the keyboard is locked in a cabinet with the two computers and is not accessible without a key. Second, the only mouse input is through the touch panel meaning that there is no right-mouse-click. Third, the browser is in "kiosk mode" meaning that it is completely full screen without any menus or other parts of the screen visible.

Unfortunately, any bug in the code that causes an error may also cause the program bar or the desktop to become visible. Once this happens, there is almost nothing that a casual hacker can't do to the kiosk. We have had around five such incidents but we have not had any truly destructive hackers. So far, they have been content simply to surf the web or play solitaire. In response, we have added an additional browser window open in kiosk mode behind the kiosk's main window. This means that if the kiosk has an error that causes it to close some module, the user does not have access to the desktop. Instead, another window takes over that does not have any active areas. The numbers of incidents have significantly decreased since this method was put in place. We realize that these measures are not likely to stop a truly knowledgeable and determined hacker, but they have been sufficient up to this point.

5 Future Directions

The current version of the intelligent kiosk serves as a useful prototype for usability analyses and as a test bed for general issues involving speech recognition in noisy environments, natural language understanding from speech, and question answering from incomplete queries. The software which is an amalgamation of Java, C#, C++, HTML, and PHP is modular and extendable with minimal component interaction thus yielding very low coupling and high cohesion. Since each component executes as a separate resource (or process) components can be migrated to additional hardware on the fly. This opens up an interesting research forum for issues related to dynamic resource allocation and recovery from partial failure.

The Intelligent Kiosk is still under development from a number of different directions. The internal workings are being updated to increase the speech recognition accuracy and cosmetic changes are being made to the interface in order to increase user interaction and satisfaction with their experience. From the perspective of speech interaction, a different speech recognition engine will be installed that is specifically designed for speaker-independent recognition.

We are also, as previously stated, in the process of conducting several qualitative and quantitative tests on the kiosk. In addition to determining its general usability, we will also be testing the users' satisfaction with the experience, ease of use, and the implications of using an avatar. Along these same lines, we are interested in how people's reactions and perceptions of the kiosk change with a different gender, voice, and race of the avatar.

Finally, additional installations are being pursued. These might include retail stores, local corporate office buildings, or healthcare institutions. Based on current experience, we would expect these installations could be implemented within three months primarily due to artistic customization.

Acknowledgements

This project was funded by FedEx Institute of Technology Innovation Grants. In addition, many people have contributed to the creation of the Intelligent Kiosk. We would like to acknowledge James Markham from the Intelligent Environments Research Lab (IE-Lab) and Lucas Charles, Austin Hunter, Loel Kim, Kristin Mudd, Susan Popham, and Michael Schmidt from the Center for Multimedia Arts (CMA) at the University of Memphis. We would also like to acknowledge Scansoft for donating the Dragon NaturallySpeaking v8.0 SDK and Smart Technologies Inc. for providing us with an Actalyst touch screen panel at no cost.

References

[1] C. Stephanidis, G. Salvendy, D. Akoumianakis, N. Bevan, J. Brewer, P. L. Emiliani, A. Galetsas, S. Haataja, I. Iakovidis, J. Jacko, P. Jenkins, A. Karshmer, P. Korn, A. Marcus, H. Murphy, C. Stary, G. Vanderheiden, G. Weber, and J. Ziegler, "Toward an Information Society for All: An International R&D Agenda," *International Journal of Human-Computer Interaction*, vol. 10, pp. 107-134, 1998.

[2] J. Cassell, T. Stocky, T. Bickmore, Y. Gao, Y. Nakano, K. Ryokai, D. Tversky, C. Vaucelle, and H. Vilhjálmsson, "MACK: Media lab Autonomous Conversational Kiosk," presented at Imagina '02, Monte Carlo, 2002.

[3] P. Steiger and B. A. Suter, "MINELLI - Experiences with an Interactive Information Kiosk for Casual Users," presented at UBILAB '94, Zurich, 1994.

[4] T. Stocky and J. Cassell, "Reality: Spatial Intelligence in Intuitive User Interfaces," presented at Intelligent User Interfaces, San Francisco, CA, 2002.

[5] J. Gustafson, N. Lindberg, and M. Lundeberg, "The August spoken dialogue system," presented at Eurospeech '99, 1999.

[6] J. Gustafson, M. Lundeberg, and J. Liljencrants, "Experiences from the development of August - a multimodal spoken dialogue system," presented at IDS, 1999.

[7] S. T. Dumais, "Latent semantic indexing (LSI) and TREC-2," in *National Institute of Standards and Technology Text Retrieval Conference*, D. Harman, Ed.: NIST, 1994.

[8] T. K. Landaur and S. T. Dumais, "A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, pp. 211-240, 1997.

[9] E. Charniak, *Statistical Language Analysis*. Cambridge, MA: Cambridge University Press, 1993.

[10] A. Sanker and A. Gorin, "Adaptive language acquisition in a multi-sensory device," *IEEE Transactions on Systems, Man and Cybernetics*, 1993.

[11] A. C. Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, D. Harter, N. Person, and T. R. Group, "Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor," *Interactive Learning Environments*, vol. 8, pp. 149-169, 2000.
[12] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. C. Graesser, "Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis," in *Proceedings of Artificial Intelligence in Education '99*. Amsterdam: IOS Press, 1999, pp. 535-542.
[13] R. Miikkulainen, "Subsymbolic case-role analysis of sentences with embedded clauses," *Cognitive Science*, vol. 20, pp. 47-74, 1996.
[14] C. Burgess, K. Livesay, and K. Lund, "Explorations in Context Space: Words, Sentences, Discourse," *Discourse Processes*, vol. 25, pp. 211-257, 1998.
[15] Y. Gotoh and S. Renals, "Topic-based mixture language modeling," in *Natural Language Engineering 6*, 2000.
[16] V. Siivola, "Language modeling based on neural clustering of words," presented at IDIAP-Com 02, Martigny, Switzerland, 2000.
[17] L. McCauley, S. D'Mello, and S. Daily, "Understanding Without Formality: augmenting speech recognition to understand informal verbal commands," presented at ACM Southeast Conference (ACMSE '05), Kennesaw, GA, 2005.
[18] S. D'Mello, L. McCauley, and J. Markham, "A Mechanism for Human - Robot Interaction through Informal Voice Commands," presented at IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN), Nashville, TN, 2005.
[19] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME-Journal of Basic Engineering*, pp. 35-45, 1960.
[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," presented at CVPR, 2001.
[21] "The Intel Open Source Computer Vision Library," vol. 2006: Intel Corp., 2006.
[22] R. J. Wallace, "The Elements of AIML Style," in *ALICE A. I. Foundation*, 2003.
[23] "Galaxy Communicator," The MITRE Corporation, 2006.