A Voice Operated Tour Planning System for Autonomous Mobile Robots

Charles V. Smith III
John C. Licato
Michael V. Doran
School of Computer and Information Sciences

and

Thomas G. Thomas Jr.
Department of Electrical and Computer Engineering
University of South Alabama
Mobile, AL 36608, USA

ABSTRACT

Control systems driven by voice recognition software have been implemented before but lacked the context driven approach to generate relevant responses and actions. A partially voice activated control system for mobile robotics is presented that allows an autonomous robot to interact with people and the environment in a meaningful way, while dynamically creating customized tours. Many existing control systems also require substantial training for voice application. The system proposed requires little to no training and is adaptable to chaotic environments. The traversable area is mapped once and from that map a fully customized route is generated to the user's specifications. All interactions with the mobile robot are done through the interface, either by using a touch screen or through speech. The system is designed to handle all special cases and even error processing in a germane manner, primarily using speech.

Keywords: Autonomous Mobile Robot, Speech Recognition, Voice Operated Interface, Robot Navigation, Human Robot Interaction and GPS Mobile Robot.

1. INTRODUCTION

Veronica JagBot is a robotic tour guide which interacts with human beings by asking and answering questions while directing people on a tour of a semi-known environment. She is a capable of understanding and synthesizing human speech for communication. Her responses vary based on context and location, using a combination of pre-written statements and dynamically generated speech.

By maintaining an easy-to-use interface requiring minimal training, we hope to create a system that can be both used and modified by a large portion of the public. This can open up interest in the field of robotics, and contribute to a wider public acceptance of both speech interfaces and robots in general.

Veronica is divided into three main systems: the interface system, the navigation system, and the real-time control system. The interface interacts with the user, which in our case is a person touring campus facilities. This paper will be focusing on Veronica's upper-level software, namely the interface and navigation systems. The navigation system determines the actions necessary to complete a series of operations and determines the appropriate route. The real-time control system (a CompactRio from NI) accesses all the low level hardware (motors, sensors, arms) and controls the movements of the robot. The majority of communication is routed through the navigation system. Thus if the interface system needs to take an action it must delegate the task to the navigation system, which ultimately decides what is sent to the real-time control system. This is displayed graphically in Fig. 1.

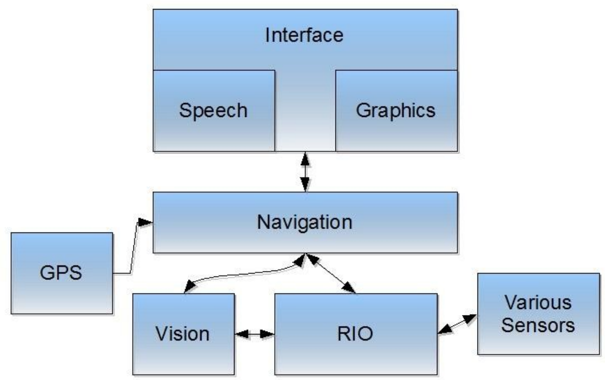

Fig. 1: Organization of Veronica's subsystems.

2. THE INTERFACE

The interface system, which receives input from and outputs content to the user, can be broken down into two subsystems: the speech system and the graphics system. This grouping is made necessary by the common information shared by both systems. For example, when the navigation system determines that Veronica has arrived at the next point on the current route, it sends a message to the interface, which then reroutes this message to the speech system and graphics system. The speech system then speaks the appropriate text for that point, and the graphics system updates the GUI.

As the highest level system of the set, the interface is limited to the commands made available to it by the navigation program. In default mode, it can not directly order Veronica to move a certain amount of feet in a certain direction. Instead the interface sends a high-level command to the navigation system to move to a specific map point. In return, it receives only high-level information from the navigation system, such as point indexes. The interface receives GPS information from the navigation system in order to update the display and to make map building more convenient. Although this abstraction of commands makes interface programming simpler, it also limits the ability of the user to control Veronica should something go wrong with the navigation system or route. For example, Veronica might come across a large obstruction that she cannot circumnavigate. For this reason, a special "command mode" is added, in which the user can manually navigate Veronica using a radio controller.

The visual portion of the interface is referred to as the graphics system. Modeled on popular map applications such as Google Maps and MapQuest, the graphics system allows for the convenient display of Veronica's current location on a map. The visual interface was designed for use with a touch screen: Buttons were made larger so that they would be easy to find and hit. This display is updated regularly with the GPS information from the navigation system. It also serves as visual confirmation of some features of the speech system, such as confirming choices a user makes while they are choosing a route. In this way, it is used not only to complement the speech system, but as a backup as well.

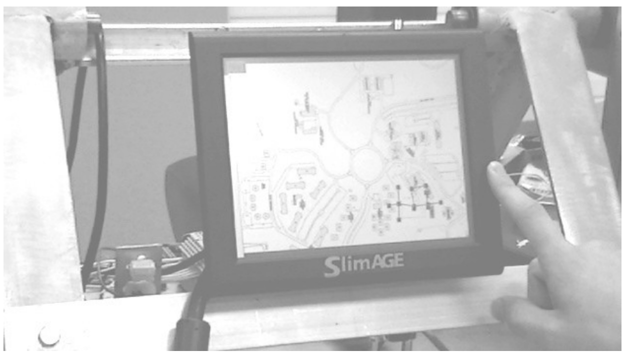

Fig. 2: Touch screen used to display Veronica's interface.

3. THE SPEECH SYSTEM

The speech system is an extension of a system created by Clay Davidson's master's thesis [1][2], which uses Microsoft's SAPI (Speech API) and C#. It was observed early on that the speech system had trouble distinguishing between statements directed at it and statements that should be ignored. Human beings in a group may determine they are being spoken to by using a combination of context, visual cues (if the speaker is facing towards them), and audio cues (such as hearing one's name, or the volume of a certain voice relative to background noise). Since Veronica can not currently pick up on visual clues, every statement directed towards Veronica is preceded with her name.

The interface system attempts to encapsulate as many of the complexities of the speech system as possible. After recognizing speech, the speech system returns several arguments that the interface then uses to determine what commands will be sent to the navigation system. The speech system, being a subsystem of the interface, does not communicate directly with any other systems. This modularity is maintained so that the speech system might be easily used in other applications, such as a route advisor on the JagTran (the system of buses that transport students around campus).

Another feature important to the speech system's modularity is its ability to adapt to context. Context is recognized by having a combination of static, pre-written grammar files and some dynamically generated grammars active at any given time. The static grammars are written to recognize questions that are likely to be asked at the robot's current location. For example, answering "where is the nearest bathroom?" would depend on Veronica's current location. thus this question's answer is redefined in the grammar for that individual point (called the "point grammar"). Questions with answers that change based on location but change less frequently than at each point are placed into a different category of grammars. Our system utilizes three categories of static grammars that can be active at each point:

General Grammars: "When were you created?", "Where are we heading?", “What is your name?”
Area Grammars: "Who is the dean of this college?"
Point Grammars: "Where is the nearest bathroom?", "What is that painting behind you?"

Dynamic vs Static Grammars

Dynamic grammars, generated either during run-time or boot time, are created using the libraries included in Microsoft's .NET 3.5 framework. The route creation grammar is an ideal example of a dynamic grammar in use. When a user asks Veronica to create a new route, a grammar is generated to recognize all of the points on the map that are *referable*. Referable points are points that can be referred to by name, such as the "jaguar statue" or "Dr. X's office". Because of the fact that these points and their names can be changed at run-time, the grammar to recognize a particular point must be generated dynamically.

Veronica's original system was composed of purely static grammars, the XML of which had to be manually written. Since this leads to a system that would be difficult to modify, it became a priority to create a more object-oriented grammar system that generated this XML automatically based on properties of objects in the map. This approach has several key advantages:
Easier to modify: It would take considerably less effort and specialized knowledge for a user to modify the map (for example, adding a bathroom at a certain location), and tying the grammars to the map in this way allow changes to the map to be made without having to manually modify all relevant grammars.
Easier to understand conceptually: This makes the system more accessible, which was one of our original goals.
Similar phraseologies are automatically applied to similar questions: Phraseologies refer to the different ways the same question can be spoken (for example: "Where is X," "What is the location of X," etc).
Speech-referable items can be weighted: By identifying all the items that can be referred to by the speech system, we can assign weights to them based on the likelihood that they will be asked about by the user. These weights can be adjusted over time based on the current context (relative location, interests of the crowd) and historical information. Using this approach, we may be able to improve the recognition accuracy of the speech system.

However, there is at least one disadvantage of switching to such a system worth mentioning. Increased vocabulary complexity, while allowing for a wider range of commands, also allows for a larger chance that Veronica might incorrectly recognize a statement. As we will describe later in this paper, this is indeed a problem that we began to experience. Although this is not inherently a problem of object-oriented dynamic grammars, it can arise indirectly as a result of automatically-generated phraseologies.

Describing Directions

Generating an answer to the question "where is X?" requires a description of a physical location, and making this description both understandable and sufficiently descriptive offers a unique challenge. In an effort to make the speech system appear more conversational, we began by analyzing two common ways that humans give directions to each other. Using the example of a bathroom located in a nearby library in a hypothetical campus:

Area-Based: These are directions which tend to use a hierarchy of areas, usually from broadest to most specific. For example: "The bathroom is in the library, on the first floor, on the eastern side of the building." Area-based descriptions are static and can trivially be generated from map information, but they are rarely descriptive enough for people unfamiliar with the area. In the example given, the speaker assumes that the asker knows the location of the library.

Path-Based: The second category of directions is based on the topological path the asker would take to get to the destination, and can be generated from the map using a shortest-path algorithm, in the same way that popular automobile GPS units give step-by-step directions. Because the specific path depends on the starting location, these are always dynamic. Perhaps the strongest weakness with a purely path-based description is that it can get rather lengthy. For example, the previous example might be: "Face south, go straight for 5 feet, go through the door, turn left, go across the street, go straight for 50 feet until you see the library, enter the back door, etc..."

Given the inherent weaknesses of area-based and path-based descriptions, it is no wonder that humans often give some hybrid of these two approaches. However, programmatically generating a hybrid description from map and path information remains somewhat unnatural-sounding, and before the final text is generated we apply the following tricks:

Remove insignificant distance and turn information: Things like "go forward for five feet" can be safely removed in most cases as they do not add much to the description.

Use reference points to shorten the path: Since we cannot assume what the user knows, Veronica starts by asking the user if they know of any points near the destination. If the user says yes, Veronica's description can start from that reference point.

Avoid using units that are not intuitive: Because the average human being does not walk around carrying a compass and odometer, it makes little sense to rely on measurements based on feet and directions. Instead of starting by saying "face east" Veronica might physically turn to face the starting direction, or give a starting direction relative to her current one.

Use the route to save time: As a way of avoiding directions entirely, we gave the speech system the ability to check the current route to see if the point being requested was ahead in the tour. If so, Veronica would respond by mentioning the object requested was on the current route, and that she would point it out when they arrived.

The resulting system would generate directions sounding much more intuitive and practical, as would be expected from a human tour guide. This was tested at the 2009 Computing, Communications and Control Technologies conference in Orlando, Florida, where we quickly created a map of the hotel the night before our presentation and had the speech system give directions to the bathroom and other locations from inside the room in which we were presenting.

Historical Awareness of Responses

Upon recognizing which question (or category of questions) was asked and which answer the question is mapped to, the appropriate response in most cases is to simply read from a static response file (defined in W3 Standard [8]). The approaches described in this paper also generate dynamic responses based on context, which can be further improved in the future. This can be seen when a question, or alternate phraseologies of it, are asked multiple times within a short period of time. In true static fashion, Veronica previously read the same scripted response to each question every time it is asked, even if asked multiple times in a row. In order to make her seem more contextually aware, the speech system was modified to allow her to choose her words based on several factors, including how many times this question has been asked in recent history. After repetition of an inquiry reaches a certain threshold she will prefix her answer with, "As I said earlier," or "Like I said already." In the future, this might even take into account something like the temperature, so that if it is particularly hot out Veronica might conclude her answer with, "but I'm getting tired of answering this same question. Let's move on with the tour already, it's hot!"

In addition to the entertainment value of this feature, teaching the speech system to be aware of the history of its responses may also benefit recognition rates. The speech grammars allow for us to adjust the probability that the speech system will match a certain rule to the speech it hears. We can take advantage of this by decreasing the probability associated with a rule that was just recently matched, as it is conceivably less likely that a user would repeat a question that had just been answered.

4. THE NAVIGATION SYSTEM

Directly communicating with the interface is the navigation system, which consists of GPS input, hardware control integration and interface integration. This is a lower-level software system handling all the path planning and high level decision making. All physical points are represented as nodes on a graph, each one containing identifying features such as a point name, index number, and GPS coordinates. Higher level information such as area data and common object categories are also contained in the nodes, but are largely ignored by the navigation system and used exclusively by the interface. The nodes are connected by directed edges containing angle and distance information.

The initial design classified all points into one of four types: decision, location, way, or speaking. Decision points were located at intersections where the robot would stop, speak, and allow tour participants to modify the route. Location points were used strictly for navigation, serving as points at which the robot would not stop or speak. At speaking points, Veronica would stop, speak, and continue the tour when directed to by the user. Finally, at way points Veronica would not stop, but might mention something in passing. For example, she may randomly comment on the presence of a soda machine or water fountain. However this method had disadvantages since modifying the type of a point changed route behavior. The point system coupled navigational data, speech data, and behavior instructions into each point, making dynamic behavior more difficult. Thus, this design had to be modified.

To rectify this problem in the next iteration, we analyzed Veronica's possible actions when reaching a point. Three factors combine to form a number of possible behaviors: Whether she stops moving at the point, speaks when reaching the point, and if she stops, whether she waits for a command from the user before continuing the tour. Because some combinations of these three factors are not possible, five responses remain:

1) Veronica stops at the point, talks, and waits for the crowd before continuing (similar to speaking point behavior)
2) Veronica stops at the point, talks, and continues (used on turns)
3) Veronica stops at the point, does not talk and continues (used on turns)
4) Veronica does not stop at the point, talks and continues (way point behavior)
5) Veronica does not stop at the point, does not talk and continues (location point behavior)

Determining which of these responses are appropriate is not always straightforward, and in many cases there are several acceptable responses. Whether or not she speaks at a point, for example, depends on what knowledge is present about the point and the interests of the user. Nevertheless, this approach allows that decision to be made independently of navigation and movement-related decisions.

Route planning is handled in this system as well. Route planning involves finding the shortest routes to several points of interest while also traveling the shortest path. This resembles the classic NP-Complete problem known as the traveling salesperson, whose complexity is not desirable on a real-time system. The solution currently implemented is simple. After designing a map, the distances from each decision point to every other decision point are calculated and that information is saved in the map. This information can be calculated offline. A route can then be found relatively easily by simply retrieving known paths between decision points. However, this solution must be temporary, as it will no longer be feasible for much larger maps.

The navigation system also handles connections to lower level systems and communicates with real-time hardware to obtain sensor information and control the robot. Calls are made to the lower level and information is obtained from the real-time control system. The navigation system will also be used to process vision and make corrections to Veronica's heading in real time.

5. HARDWARE-SUPPORTED CAPABILITIES

Veronica JagBot stands at just under six feet tall and has the ability to move in all directions using a six wheel differential drive system (see Fig. 3). Two motors are used: one for the left drive wheels and one for the right. Two Hall-effect sensors are used to count the revolutions of each motor gear. The Hall-effect sensors allow for fine tuned motor control and provide information on distance traveled [5]. There are two bumper sensors for generating a tactile interrupt in case the robot hits an object that was not otherwise detected.

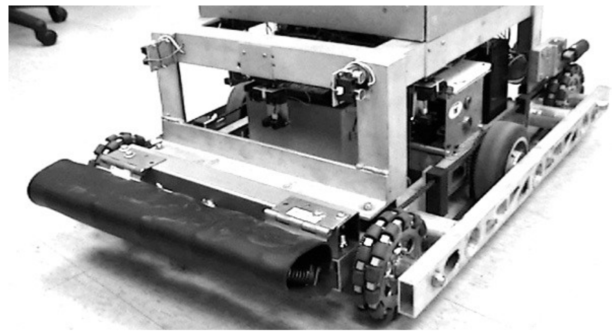

Fig. 3: Veronica JagBot's base.

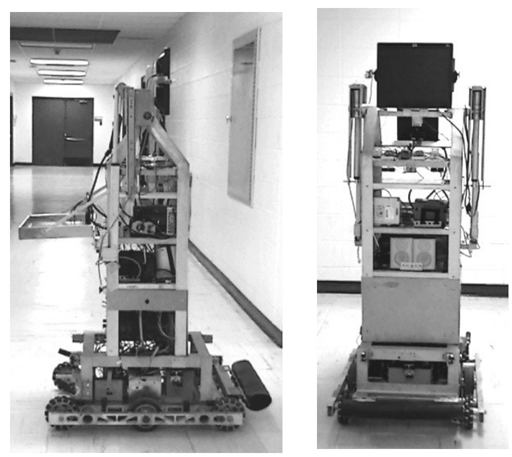

Fig. 4: Veronica Jagbot as of March 2010.

Other sensors available to Veronica include infrared and sonar. The sensors are used to obtain distance information from the robot's position relative to surrounding objects [7]. Video cameras are also used to collect data and can possibly be used to identify objects by recognizing common features of known objects. All of these sensors are collectively used for obstacle avoidance and fine-tuning of movement.

Originally, Veronica's primary method for determining location was to utilize data from the GPS system. However, measurements revealed that our GPS unit was not accurate enough to be used as a primary tool, especially when next to a building or nearby a large obstruction like a tree. Additionally, GPS is consistently unavailable when inside of buildings. More precision is required in order to stay on sidewalks and other pathways, and therefore GPS has been reduced in importance, used instead as secondary verification of location. The specific location within that area is primarily determined using a combination of vision, sonar, infrared, and distance readings.

On the other hand, odometry gives a fairly accurate determination of position relative to a starting point, since the wheels tend to hold good traction [4]. The compass is accurate to within a few degrees in wide open areas, providing a reference point while turning. However, when next to certain buildings on campus compass readings have varied significantly, suggesting interference from ferrous or magnetic objects.

The interface and navigation systems are integrated using a TCP/IP connection, enabling the programs to either operate on the same machine or on different machines over a network. One machine will be able to handle the demands of voice processing (through the interface) while the other machine will perform high-level logic. Veronica has a local area network on-board which is used for facilitating this communication.

In her current state, Veronica's hardware is capable of executing commands to move forward, backward, make turns, and stop within a predefined range of forgiveness. These abilities will improve as research continues in the areas of sonar, infrared, and vision [6].

6. NORMAL OPERATIONS

When first starting up, Veronica attempts to use GPS and sensor data to determine her location. If that location can not be determined, Veronica asks the user to tell her the point at which she is currently located. The user then tells Veronica to begin the route creator, which is the voice module that simply accepts points of interest. When all the points are added the user confirms the selections (Fig. 4), and the navigation system organizes the points and begins the route (Fig. 5). The tour then begins by simply telling Veronica to continue.

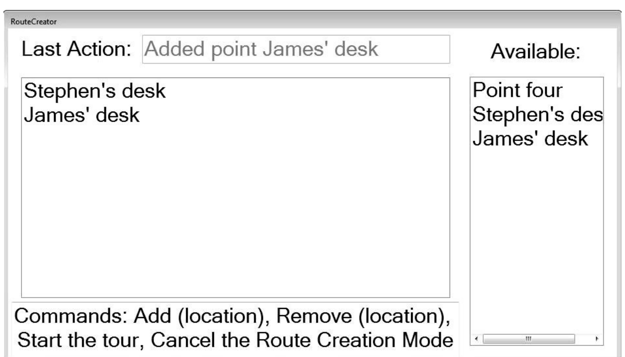

Fig. 4: Visual interface displaying a route confirmation.

Veronica then travels to the point, and when the navigation system determines that the point has been reached, this confirmation is sent to the interface. The speech system then starts a new conversation based on that location. At the end of the tour, the user can restart the route generator and instantly begin another route. In most cases, this will be a route taking the users back to their starting location, but it is possible to stack routes.

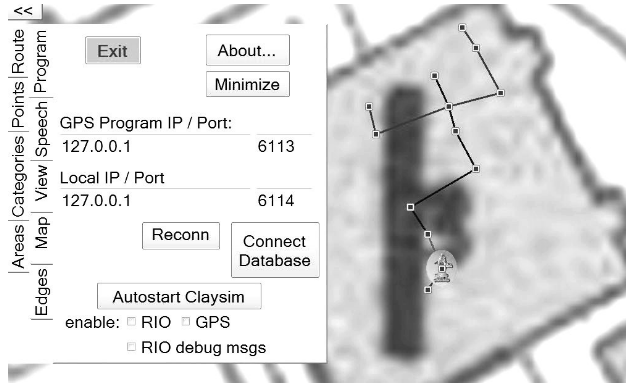

Fig. 5: Visual interface displaying a created route.

Anywhere along the route, the path can be changed due to either recovery from error (if for example a path is unexpectedly blocked) or a change of mind of the user. In fact, Veronica encourages this dynamic changing of the route. When Veronica reaches a decision point, she suggests nearby points of interest not on the route and if the user agrees, they are automatically added to the route. Veronica currently decides which points to suggest based on two factors: proximity, and whether or not the point was already visited. This framework allows for different factors to affect this suggestion in the future, such as information about the user's interests and historical user interest in that point. In this way, Veronica can even further customize the tour into one that the specific user (or tour group) will enjoy.

Error recovery, however, is a critical operation of the interface. A lot can go wrong when a six-foot robot guides a crowd of tourists around an outdoor environment, as we have learned many times throughout this project. These can be broken down into two categories: software/user errors and hardware errors.

Software / User Errors

Since a stated goal of our project is a system with high usability with little training, there is considerable room for user error, and the interface thus needs to be extremely robust. Perhaps the most common source of errors will be incorrect recognition of user speech and commands. All speech data is logged, so immediately after Veronica gives an incorrect response the command "Veronica, that was wrong" allows for future recalibration of the system. This can be achieved by examining the context in which the incorrect recognitions are made and developing patterns to prevent them in the future. The speech system also requests confirmation at times, and in the case of route creation displays a visual confirmation and allows for correction before the route is finalized.

Veronica depends heavily on map information, so incorrect map information can lead to serious error. The front and back bump sensors shut down Veronica's motors immediately, preventing any damage in case of collision. Route information can also be modified at run time in case of last-minute changes, such as an edge that is no longer traversable due to an obstacle. In this case, because computer vision is not currently utilized, Veronica must manually be driven to the next point on the tour, where she will automatically resume. It should be noted here that autonomous obstacle avoidance through vision is an interesting future direction of this project.

Hardware Errors

Catastrophic hardware errors that lead to a loss of control of the hardware system by the upper level software are outside of the realm of control of the software, and can only be handled by manual shutdown (most notably a big red switch on the back of Veronica that instantly shuts down the entire system). However, because of the fact that the interface runs on laptops using their own battery power, total hardware failures almost always leave the upper level software running. Should the navigation system detect that it has lost connection with the lower level hardware, it simply waits until the connection can be resumed. This has proven useful during events where Veronica had to be radio controlled, either because finer precision was needed to navigate through narrow doors, or because her path was unexpectedly obstructed. Software recovery then attempts to bring Veronica into a well-defined state.

Inaccuracies with the compass or GPS readings can cause Veronica to think she is at the wrong point, or point towards the wrong direction when turning. For the purposes of these tests, a delay was introduced in between her turning and forward movement stages, to allow for manual correction. Additionally, routes were chosen that tend to be in wide-open areas, to improve reception and reduce any possible interference.

7. SYSTEM TESTING

Interface Testing

To demonstrate the feasibility of the interface and navigation systems, a simulation mode was constructed without the lower level control system. This system was tested on a single laptop using the GPS, interface, and navigation systems. Without information from lower level hardware, GPS alone was used to determine location. The example tour used only a few points loaded into a map covering a small outdoor area. In place of the wheels moving Veronica from point to point, we simply placed the laptops on a cart and had Veronica instruct us to manually walk to the given point.

The results were promising. Veronica was able to provide correct routes, and there was virtually no delay in the transition between active grammars. Although we were outside and using the built-in laptop microphone, the speech system was able to recognize spoken commands, which was consistent with Davidson's thesis [1].

The interface and navigation systems were also tested by putting the laptops in a car, essentially repeating the previous test over a wider area (the southern half of the University of South Alabama campus. Part of this test can be seen at http://www.youtube.com/watch?v=AMSi6UKx0Xo). The upper-level systems successfully provided a tour, proving sufficient accuracy of the GPS system and demonstrating that the basic map and route creation features were ready to be tested with the lower-level hardware.

Full Integration Testing

A fully integrated test of the system came after the success of the interface and navigation test. Points were chosen to avoid obstacles like steep hills and crowds. A map containing the selected points was created, containing speech information and static grammars unique to each point. Our goals for this series of tests were to have Veronica decide the best route to take for the user requests, safely navigate point-to-point, and narrate at each speaking point.

The real-time control system was then connected to the navigation system, which ran simultaneously on the same machine as the interface. The map was selected by the user through the interface and sent to the navigation system. After several points were selected using the route creator, Veronica successfully and consistently determined the best route including these points, and turned in the direction to travel. Using odometry to notify Veronica if she has traveled the correct amount, Veronica was able to successfully traverse a route and return to her original point.

Some issues became apparent with these initial tests. When reaching a point and preparing to traverse the next edge on a route, Veronica would rely solely on compass data. This data was inconsistent inside of the building in which we performed the indoor tests due to some sort of magnetic interference, and as a result Veronica would take anywhere from thirty seconds to a minute to line herself up with the correct compass value. Even when the alignment was complete, the inconsistency of the original map information would require us to manually correct Veronica's heading. Additionally, without any sort of verification along the way to ensure Veronica was going straight (no machine vision was implemented at this stage) she had trouble staying on the path without being in danger of hitting objects on the side. Unfortunately, this problem only became worse when Veronica was taken outside into a less controlled environment, where she would be required to stay on sidewalks.

Nevertheless, the upper level route creation and tour management system were largely successful. In a limited outdoor tour directly outside of the engineering building (which can be seen at: http://www.youtube.com/watch?v=BFIP0y5kWoo), Veronica was able to describe her location to the users at several points, and to generate directions to objects on the map, some of which were added mid-route.

Next, a larger tour was organized covering about 1/4 mile and containing 20 points. This tour would bring Veronica through heavily populated student areas, requiring her to traverse through hallways, over sidewalks, switch from outdoor to indoor settings, finally concluding in front of the University President's office. Veronica was manually controlled once during this tour (for her safety), at a car crossing which led to a particularly narrow and steep ramp. After this, however, her location was manually updated and the tour resumed at the correct location.

8. RESULTS

We will now examine some stated goals of this system and describe the results of the tests in achieving them:

High Usability

In order to gauge the usability of a system we must first identify the needs of the user(s). The user needs to interact with a tour guide robot in order to obtain information about an otherwise unknown area. The guide should avoid using any colloquialism that would confuse the tourist. The system interface needs to be designed with the casual user in mind.

Buttons need to be clearly labeled and some features may require tips to encourage correct user input. Veronica's interface implements all of these ideas.

The voice system also is usable with very little training. Because of the wide variety of alternate phraseologies that are accepted by the grammars, specific commands need not be memorized in detail. This is an important feature of the system and contributes greatly to its overall usability.

Increased Public Acceptance of Robotics and Speech Interfaces

The minimal training required by the speech system serves an additional purpose in that it also has been shown to increase public acceptance of robots and speech interfaces in general [1]. The success of the tests mentioned in this paper demonstrated the potential for widespread public acceptance of Veronica as a tour guide. Although still aware that she was a robot, users were interacting with her and receiving responses naturally. She gathered notable attention from the students passing by, who seemed to express a genuine interest in the project and excitement looking forward to a campus-wide tour.

Versatility

In addition to its use as a tour guide robot, there are many diverse applications of this system. We have demonstrated, in our in-car test, that the system could be used to provide information to users who are driving around campus. This means individuals touring an area on their own or for a bus full of tourists might be benefited by use of this interface. We have also demonstrated that the system could be used for individuals on foot in the walking tour. Such a walking tour could conceivably be adapted for use on a smart phone. Another university-related application is for new students getting acclimated with campus. This could be used to keep freshmen informed about events and services on campus and could serve as a public relations tool. The system could be adapted to robots used for facility tours or even remote tours of the campus.

9. FUTURE DEVELOPMENTS

There are a number of features that are in progress that would make interactions with Veronica richer and more captivating:

Generalizing common grammar structures

As mentioned earlier, there are often many different ways to ask the same thing. "Where is the bathroom" can be asked a certain number of ways, but their general phraseologies are similar to those of other questions that ask for an object's location. An informal template was developed to take advantage of these similarities, so that whenever a new rule (a question-answer set) was added to a grammar, the most common variants of the question would be easy to list. For example, a question asking where the location of an object X is would include the following forms:

Where is X?
Where can I find X?
Can you tell me where X is?
What are the directions to X?

The grammars associate each group of phraseologies with a single response code, which is then processed to dynamically generate a meaningful answer. It is hoped that in the future, the speech system can be modified to instead recognize the form of the question and the subject of the question separately. This would give us the advantage of a cleaner, more object-oriented approach. The speech system would then return two pieces of information to the interface: The name of the object, and the name of the property it is requesting. "Where is the bathroom?" would be broken down as a request for the "where" property of the object "bathroom." Such a feature opens the door to speech grammars that can be generated automatically based on map data and location context, which might be applied to a wide variety of other areas that use speech recognition.

This approach, however, requires that the speech system be trained to recognize individual words, rather than recognizing on a sentence-by-sentence basis. The accuracy of this approach will need to be tested using the same conditions as the original thesis to determine if there are any improvements or decline in recognition rates as compared to Davidson's thesis [1].

Upper Level Data and User Interests

Work is currently being done on examining exactly what upper-level data can be stored in the map to deliver a more enriching experience for users. Ideally this would be information that can be directly translated into an appropriate grammar, so that it can be recognized and referred to by the speech system automatically. A possible future area of research is setting up a website through which students can input information about clubs, offices, and possibly even events, that can then be processed and imported into Veronica's map data.

This map object metadata can be used to create more customized tours for the users. Interests of the users can be extrapolated from the types of questions they ask, or which points they choose to visit on the tour, and more meaningful suggestions of places to visit can be made. Furthermore, the speech system can choose which facts to mention about points visited on the tour, creating a tour guide system in which no two tours are the same.

Finer Control

Better control of the robot is important to Veronica's successful completion. Because she is so large, she has had trouble moving through doors and staying on sidewalks. By tightening up some of the control algorithms and using sensor data to reduce the probability of error, Veronica will appear graceful and intelligent rather than big and clumsy. Investigating and upgrading hardware as well as incorporating new types of senor data may also be necessary.

Machine Vision

Our experiments with GPS and the compasses have shown that they are not reliable enough for the fine-tuned navigation Veronica requires. This has made it clear that vision is much more important than originally assumed, and so an increased emphasis has been placed on getting Veronica to “see” her surroundings and act on what she sees. This can be done by actually collecting data on the tour and manually going through the data later identifying objects (like doors or sidewalks) and training the vision system to better recognize such things. This will be an ongoing process, currently in development, and Veronica should continue to improve and evolve as she learns to identifying shapes, lines, and eventually complex objects [3].

10. CONCLUSIONS

We have attempted to create a control system for tour guide robots with an emphasis on usability and easy accessibility. The interface and navigation systems can aid a person who is unfamiliar in an area by giving the person clear paths to take, and offering suggestions of locations of which the user might not otherwise have heard. The result is a virtually seamless integration of navigation, object awareness, and speech, providing value-added information to a completely customized tour.

The work done on the speech system also offers several general applications. Our emphasis on making sure that the speech interface requires little to no training gives this robot a broader appeal. It also offers a template that can be followed to enrich robot-human interaction in areas from customer service to assisting the visually impaired. The potential ability of our speech and map system to dynamically generate speech grammars is also an application that opens up customization of speech systems to those who are less tech-savvy. Casual software users might one day be able to use our system to configure custom tours of any area, which we will in fact test as we take Veronica to high schools in the area.

Veronica is capable of traversing any number of routes in series as long as she can safely reach points within a sufficiently large margin of error. Since GPS alone failed to deliver an acceptable level of consistency, an increased emphasis is placed on Veronica's hardware system. After compensating for this loss of accuracy, the walking tour results should be reproducible on the fully autonomous system, giving any user a dynamically tailored tour without ever making any changes to the code implementation.

11. AKNOWLEDGEMENTS

This work was supported by NSF grant IIS-0744070. The authors would also like to acknowledge the work performed by Clay Davidson, Aaron Munoz, Jenny Plott, and Matthew Rudnick. The authors also acknowledge the initial efforts of our colleague and co-investigator Dr. W. Eugene Simmons who unfortunately passed away before the completion of this work.

12. REFERENCES

[1] C.C. Davidson, **Investigating a Voice Enabled Robotic Interface for Noisy Environments**. Thesis U of South Alabama, 2008.

[2] C.C. Davidson, M.V. Doran, W.E. Simmons, and T.G. Thomas, **Veronica JagBot – A Robotic Campus Tour Guide**, WorldComp'08, 2008 International Conference on Artificial Intelligence IC-AI'08, Las Vegas, NV, June 2008.

[3] D.G. Lowe, **Object recognition from local scale-invariant features**. In International Conference on Computer Vision, Corfu Greece, pp. 1150–1157, 1999.

[4] F. Chenavier and J. Crowley, **Position Estimation for a Mobile Robot Using Vision and Odometry**, Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, May 12-14, pp. 2588-2593, 1992.

[5] J. Borenstein, H. R. Everett, and L. Feng. "**Where am I?" sensors and methods for mobile robot positioning**, Technical report, University of Michigan, 1996.

[6] J. Tard´ os, J. Neira, P. Newman, and J. Leonard, **Robust mapping and localization in indoor environments using sonar data**, Int. J. Robot. Res., vol. 21, no. 4, pp. 311–330, Apr. 2002.

[7] Kleeman, L., and Kuc, R. **Mobile robot sonar for target localization and classification**, Int. J. Robotics Research 14(4):295–318, 1995.

[8] **Speech Recognition Grammar Specification Version 1.0**, 16 Mar 2004. World Wide Web Consortium. 1 Jul 2008. <http://www.w3.org/TR/speech-grammar/>.