Voice Control of da Vinci

Lindsey A. Dean and H. Shawn Xu

Mentor: Anton Deguet

5/19/2011

I. Background

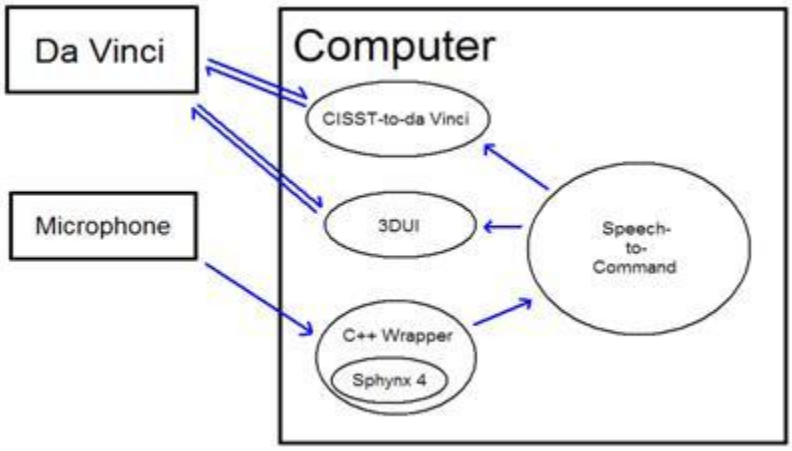
The *da Vinci*® is a tele-operated robotic surgical system. It is operated by a surgeon sitting at the surgical console using his or her hands and feet. Limited by the physical constraints of only having two hands and two feet, surgeons often must perform a myriad of different gestures during surgery in order to do some simple peripheral functions. These gestures, such as clutching in and out, mean that the surgeon is temporarily relinquishing control of the robotic arms and surgical tools. This leads to stop-start procedures and inefficiency. Furthermore, the problem is only getting worse, since each new version of the system is more complex than the last.

Our project explores voice integration with the *da Vinci*®. We believe that voice is an appropriate medium for the surgeon to control some peripheral functions during surgery without having to physically move his or her hands and/or feet. It would allow the surgeon to devote more of his or her attention to controlling the robotic arms and the surgical tools attached to them. This would serve to ultimately smooth the surgical process. Ultimately, the integration of voice control would serve to smooth the surgical process.

Voice control of surgical robots has been explored previously, most notably in the case of ComputerMotion's introduction of the AESOP robot in the nineties. The AESOP robot consisted of a laparoscopic camera fixed to a robotic arm. There were two ways to control he motion of the camera: either through a joystick or with voice commands such as "right" and "left". However, many in the medical community complained that the voice control feature caused long reaction times and limited reliability, resulting in the AESOP's discontinuation. This provides evidence that voice control is not appropriate for everything. The ultimate goal is to smooth the surgical process, so the idea is to use voice control in a way that makes the human-robot interaction more intuitive and less distracting. We kept this in mind when we were identifying which functions and behaviors of the *da Vinci*® would be best controlled by voice commands.

We first identified control of the 3D graphical user interface overlay (3DUI) part of the Computer Integrated Surgical Systems and Technology (CISST) Libraries developed here at Johns Hopkins University . The 3DUIallows the surgeon to perform many basic tasks like measuring the distance between two locations by moving the robotic arm, or adding and deleting visual markers from the display. It also allows the surgeon to bring up a previously loaded 3D model on his or her display during surgery. All of this however, requires that the surgeon use the master arms as pointers to select menu buttons. We felt that voice control would significantly improve this program, since it would allow surgeons to perform these tasks while continuously performing surgery. We made it our project goal to create a proof-of-concept demo that allows the surgeon to perform the same functions using voice commands as those available from the 3DUI..

II. Technical Approach



For any voice command to trigger an action, three things must happen. First, the program must recognize what word or phrase was said. Next, the program needs to determine the correct action to take based on what was said. And finally, the program needs to actually execute that action. Each phase of this process is implemented as a separate component in the CISST library. They are connected by matching provided and required interfaces.The reason for this is so that it

becomes it easier in the future to switch out some part for another. For example, it would be very easy to switch to a different speech recognition component, so long as that component has the same provided interfaces as the current one. Here, we evaluate each of the three phases separately, then show how we put them all together.

Speech Recognition

The speech recognition package that we used is Sphynx 4. Sphynx 4 is a package written in Java that was developed at Carnegie Mellon University. Since the rest of CISST is in C++, the component in the library is really just a wrapper around the Java program, which executes at runtime on a Java Virtual Machine (JVM), and is accessible through the Java Native Interface (JNI). This fact actually caused quite a bit of trouble because it forced us to compile all of CISST as dynamic instead of static libraries.

Command Execution

In order to execute commands, CISST must be able to communicate with the *da Vinci*®. This is done using the CISST-to-daVinci package, which makes use of Intuitive Surgical's black box API. However, for our program we are specifically concerned with reproducing the behaviors of the 3DUI, another part of CISST. Thus, we simply used the same behaviors and added or modified the provided and required interfaces as necessary.

Speech-to-Command

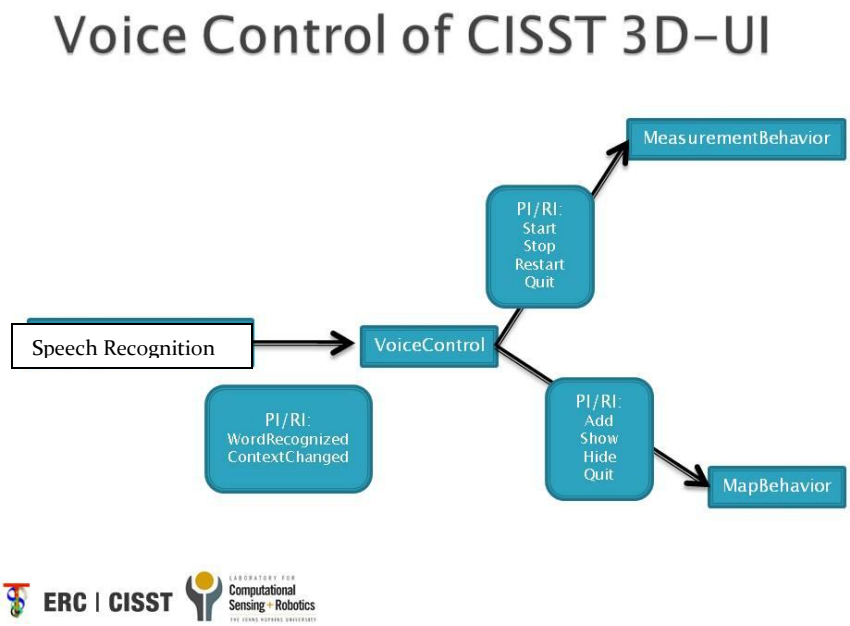
This component takes the output from the speech recognition component (a string), decides what (if any) action to take, and triggers the correct event. In other words, it waits for an event from the speech recognition component and fires an event at the desired behavioral component. Here, all the logic that determines which commands trigger which actions is defined.

Our approach to the logic was state-based. We felt this was a good way of mimicking the menus that come with a graphical user interface. Each state is defined as a context. Within each context, a certain list of words that the program is listening for is defined. This is called the grammar. Each word in a grammar can do one of three things: it can trigger a specific action, it can change the current context, or both. The voice user interface starts out in a passive state. This allows the surgeon to proceed without the use of voice control if he did not wish to engage it. In this state, the program listens for the keywords that trigger the active listening state. and when it hears a keyword, it transitions. In the actively listening state, the user can enter different modes (contexts) in which he or she can trigger different commands (defined in the grammars of those contexts). Or, he or she can exit back to the passive context.

Currently, voice control is implemented as a behavior component in the 3DUI behaviors package.

Putting It All Together

The main program itself is very straightforward. First, a speech-recognition and 3DUI component, containing specific speech-to-command behavior are instantiated. The required and provided interfaces are connected as follows:



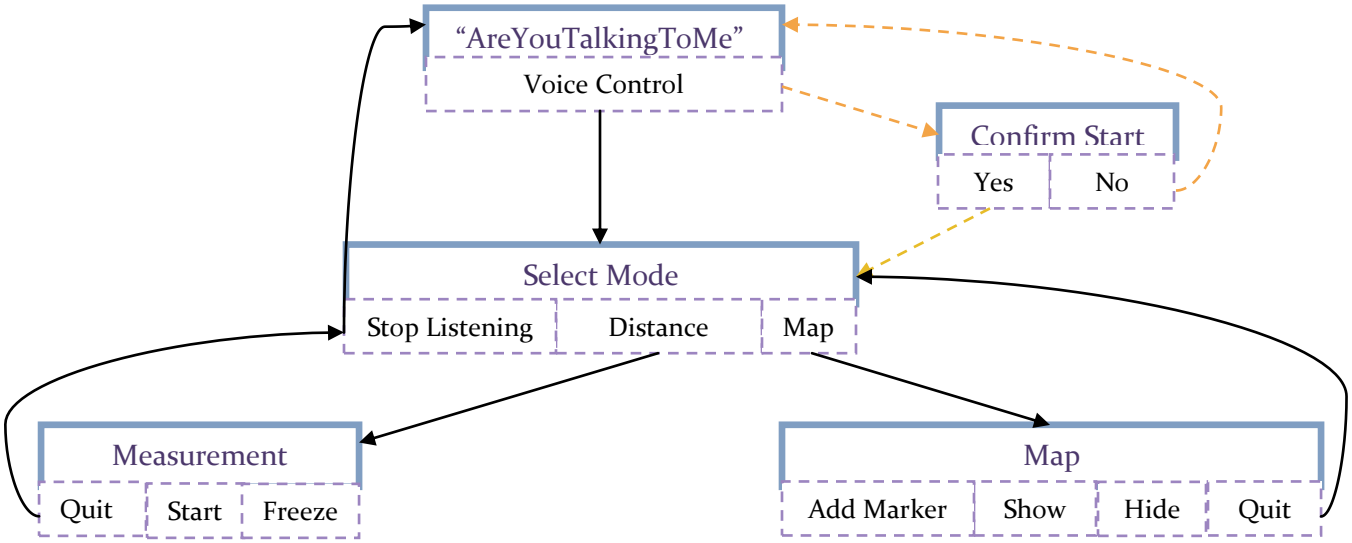
The corresponding provided and required interfaces are connected as shown above. Finally, the components are added to the CISST multithreading component manager.

The simplicity of the main program shows why we chose the software architecture we did. As long as the provided and required interfaces match, components may be swapped out and replaced. It would not be difficult to adapt our program for another speech recognition package, or similar robotic surgical system (so long as the desired behaviors already exist as a component).

III. Results and Discussion

To test the viability of our technical approach we identified two menu options from the 3DUI: measuring distance and creating a 3Dimension map of markers to incorporate into a proof-of-concept demonstration. We felt these behaviors demonstrated the most practical use of voice integration. During our exploration phase on the daVinci in the beginning we noticed how cumbersome it was to use the master controller as a mouse in order to start an application. Additionally the clutch must be held down constantly while the user aims the finger grippers at

the icon of their choice -- pinch and release. However, now the user can enter the voice control interface and speak the commands “distance” and “map” to trigger the same events that previously required a great deal of physical exertion. The entire logic of our voice control demonstration is as follows:



In the preceding figure, the words in the blue boxes represent the contexts and below them in the dashed lines are the commands listened for while in the corresponding context. To facilitate development we included a widget that displays the possible words while in each context. We believe that visually cuing the surgeon to speak is more intuitive and less distracting than having the machine recite the options for each menu.

To enter the voice control interface the user must first recite the command word which we have set to “voice control,” and once this word has been recognized it causes the 3DUI to run in the background. From this step we currently have a confirmation context which as can be seen connected in red as in the future it is not necessary. Our original choice to have it was to try to decrease the probability of accidentally entering the interface, however in practice we found that it did not actually decrease the amount of times the interface was accidentally entered and

therefore only decreases efficiency. Once the user is in select mode the command options are "distance", "map" and "stop listening." The latter of these simply returns the speech recognition back to its idle state.

"Distance" opens up the measurement feature in the 3DUI which causes a green number indicating the distance travelled to appear on the screen. Currently, whenever the measurement state is entered through voice a non-zero is displayed. Therefore it is necessary to say "start" in order to reset the value. To help the user with this glitch until it has been solved we implemented an alternative command of "reset" which serves the same function as "start" however since the meanings can be interpreted very vaguely we wanted to include all commands a user may want to say. In order to stop the measurement from changing the user must say "freeze." The reason we chose this particular word was that we knew "stop" could cause too many words to be misrecognized since "start" was the first word we implemented. Finally "quit" simply returns the user back to "Select Mode." We found this set of vocabulary to be particularly robust and able to handle a wide variety of voices and accents.

In "map" mode, the surgeon can add, show, and hide overlay markers to the visual display in 3D space.

Our demonstration confirms that voice is an appropriate interface for controlling peripheral functionality on surgical robots. However, due to the limitations of speech recognition in terms of accuracy and reliability, the use of voice control for complex physical functions such as the movement of surgical equipment could potentially be dangerous and is therefore inappropriate.

Through our work this semester we successfully developed a proof of concept for the potential of integrating voice control. We have met all of our expected goals and were able to recover from unanticipated technical difficulties with compiling the framework and the always fickle Sphinx4.

Management Summary

Together we accomplished the following milestones which comprise our expected deliverables.

All aspects of this project were shared equally and completed as a pair.

Milestone	Status	Date Planned	Date Accomplished
1. Overcome logistical dependencies: NDA, Mock OR access, JHED accounts	DONE	2/27/11	
2. Ready for Software Architecting: all necessary libraries on computer and functional walk through of how current system is working.	DONE	3/12/11	3/17/11
3. Approved Document of software framework: create object oriented class design with all relevant necessary/required interfaces.	DONE	3/12/11	3/23/11
4. Working demo of voice control on daVinci robot –INTUITIVE DEMONSTRATION	DONE	4/17/11	4/21/11
5. Incremental improvement of first voice demo: meeting with mentor in between and discuss strategies to improve existing demo	Future Work	4/20/11	4/23/11
6. Improve logic of CISST 3D-UI to increase ease of use for implementing other types of interfaces	Future Work	Future	---

Future Work

One functionality we really wanted to implement but did not have time for was a voice command that allows the camera to find the tools when they move out of the surgeon's visual frame. This required learning about elements of the CISST libraries we did not have time for. Additionally the CISST library functions could be simplified and made more efficient so that other types of interfaces could be easily plugged into the framework.

What We Learned

- Always be ready with backup plans for delayed or unresolved dependencies (
- It's impossible to foresee every obstacle, so plan accordingly
- Having a good mentor makes your project a lot less stressful taught us a lot about project management and the stages of software development

References

A. Kapoor, A. Deguet, and P. Kazanzides, "Software components and frameworks for medical robot control," in Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, 2006, pp. 3813-3818. "Sphinx-4." CMU Sphinx - Speech Recognition Toolkit. Web. http://cmusphinx.sourceforge.net/sphinx4/javadoc/index.html.

Liu, Peter X., A.D. C. Chan, and R. Chen. "Voice Based Robot Control." International Conference on Information Acquisition (ICIA) (2005): 543.
Web. http://ieeexplore.ieee.org.proxy3.library.jhu.edu/stamp/stamp.jsp?tp=&arnumber=1635148.

Patel, Siddtharth. "A Cognitive Architecture Approach to Robot Voice Control and Respons." Web. <http://support.csis.pace.edu/CSISWeb/docs/ MSThesis/PatelSiddtharth.pdf>. (2008)

Schuller, Bjorn, Gerhard Rigoll, SalmanCan, and Hubertus Feussner. "Emotion Sensitive Speech Control for Human-Robot Interaction in Minimal Invasive Surgery."Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication(2008): 453-58. Print.

Schuller, Bjorn, SalmanCan, Hubertus Feussner, Martin Wollmer, DejanArisc, and BenediktHornler. "SPEECH CONTROL IN SURGERY: A FIELD ANALYSIS AND STRATEGIES."Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on (2009): 1214-217. Print.

Sevinc, Gorkem. INTEGRATION AND EVALUATION OF INTERACTIVE SPEECH CONTROL IN ROBOTIC SURGERY. Thesis. Johns Hopkins University, 2010. Print.