

iPad Mobile Surgical Console

600.446 Computer Integrated Surgery II

Project Proposal

Written February 17, 2011

Mentors: Balazs Vagvolgy
Russell Taylor

Team Members: Hanlin Wan
Jonathan Satria

Project Summary

The main goal of this project is to develop a simple yet resourceful application for the Apple iPad that allows remote configuration of surgical robot components. When fully implemented, the application will serve the role as a mobile, centralized, ergonomically designed tablet with which the surgical team can dynamically change settings of the robot. The targeted system to test this device is the EyeRobot.

Background and Significance

An operating room that utilizes a robotic surgical assistant often requires several computer workstations, each controlling different devices or aspects of the robot-human interface. This decentralized approach places both ergonomic and efficiency constraints to the surgeon and his or her team. Practically speaking, individuals are required to approach the correct component workstations, which often placed several meters away from the machine if not in another room, and input the correct settings. This certainly poses disadvantages, which a central command module could dramatically improve upon. The current input method of keyboard and mouse also poses some disadvantages – besides being difficult to sterilize and clean, it is certainly not appropriate for the fast-paced operating room, especially when touchscreens have become much more widely available.

Through the iPad, our application will provide an easy to use interface with which to control several component modules. As a portable device, it will also allow easy access of the robot components to several members of the surgical team.

Technical Approach

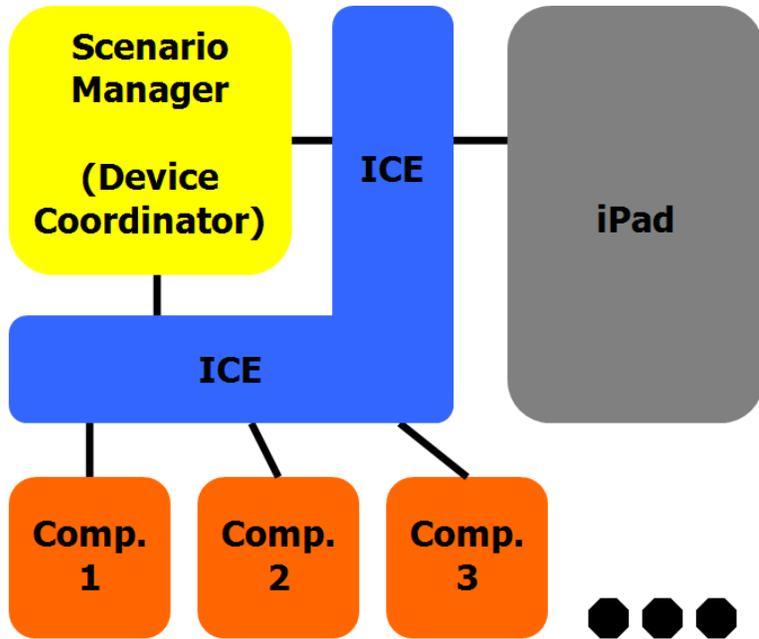
In this project we will attempt to use several tools to build a functioning iPad Mobile Surgical Console. These include:

- **Cisst Library:** This library is the backbone of the project. All communications to and from the various components uses this library.
- **CMake:** CMake allows the compilation of the source code on any platform. While it doesn't officially support iOS, online sources have stated that this is possible, albeit difficult.
- **iOS:** Development of the iPad application relies on the iOS framework. Apple's iOS Development Kit will be used for the design and implementation of the GUI.
- **ICE:** The Internet Communications Environment manages the connections between multiple computers. This simplifies the programming required to get computers to communicate with each other.
- **Scenario Manager:** This tool controls the interactions between various components. On startup, it finds all connected components and forms links between them.

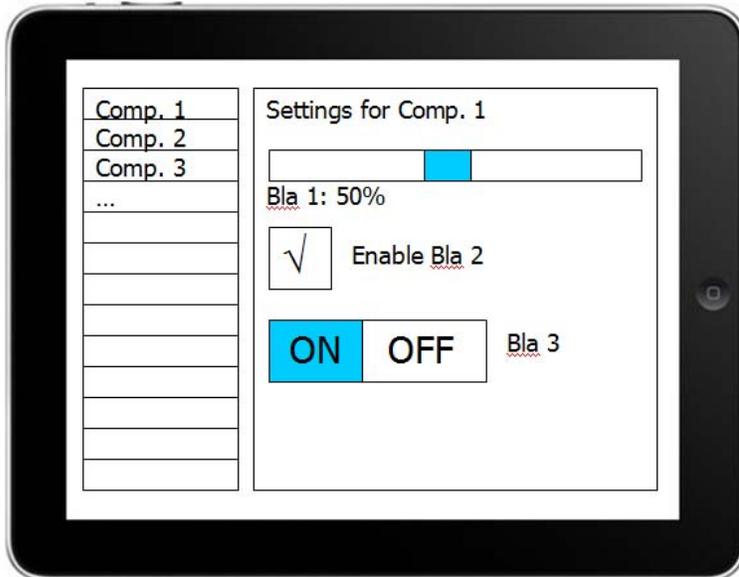
Although there are actually several approaches in completing this project, we first hope to be able to compile a subset of the cisst library on the iPad using CMake. This would allow easy integration with the ICE protocol, since cisst already provides communications with this technology. We may also need to develop wrapper functions to make valid function calls to the components from the iPad device, and this will require working with the Scenario Manager. We consider this to be the backend and brunt of the work required in this project.

Once these technologies have been successfully integrated into the iPad, GUI development for the actual application will begin. This may take several weeks, but we aim to produce a framework through which future components can be easily added to the GUI by late-March. Once a preliminary application has been developed, testing and integration within the OR can begin.

To illustrate the final schema, a diagram of the different components has been included on the following page.



A preliminary GUI for the iPad has been cartooned below, where component devices can be accessed easily through the panel on the left, and system configuration can easily be navigated in the large panel on the right. We hope to obtain a GUI much more aesthetically pleasing than this, however.



Deliverables

Minimum (develop iPad as dummy console)

- Install VNC
- Develop GUI for Scenario Manager

Expected:

- Compile and build cisst library on iPad using CMake
- Build GUI iPad application

Maximum:

- Clinical runs - Mock OR
- Revisions based on user experience

Milestones

1. Core Components
 - a. Install the core components – CMake, CISST, ICE – onto the iOS
 - b. Documentation for the interface between these
2. GUI Component
 - a. Become familiar with iOS GUI development
 - b. Build GUI to control the various components
 - c. Code the interfaces between the GUI and the components
 - d. Documentation for GUI design and interfaces
3. Testing
 - a. Test device with EyeRobot
 - b. Revise GUI based on user feedback

Dependencies

Dependencies already resolved:

- two Apple iPads at \$499 each
- one Apple Developer Subscription at \$99
- overhead costs related to obtaining the right development environments (namely, Mac machines)

Dependencies that still need to be resolved:

- one more Apple Developer Subscription (hopefully will be resolved by 2/25 but not urgent since we already have one to work with)

Management Plan

- Weekly meetings with Balazs
- Task Leaders: instead of having one individual complete an entire module of the project, we have decided to appoint a leader
 - o Jonathan: task leader for backend cisst integration
 - o Hanlin: task leader for GUI/frontend
- Mutual collaboration in terms of implementation

Timeline

As discussed before, there are three major milestones to this project: successful integration of the CISST library and its associated technologies (yellow), development of the application GUI (green), and testing and revising (red). We aim to complete the systems integration by Spring Break 2011 and begin developing the GUI shortly thereafter. Final testing and revisions are scheduled for late April and early May.

Task	February		March				April				May		
	1	2	3	4	5	6	7	8	9	10	11	12	
Install Cmake	Yellow	Yellow			Spring Break								
Compile CISST	Yellow	Yellow	Yellow	Yellow									
Install ICE			Yellow	Yellow									
iOS Interface Documentation			Yellow	Yellow									
Familiarize with iOS Development			Green	Green									
Build GUI for Components							Green	Green	Green				
GUI/Components Interface								Green	Green	Green	Green		
GUI Documentation								Green	Green	Green	Green	Green	Green
GUI Revisions											Red	Red	Red
Mock OR Testing											Red	Red	Red
Project Documentation	Cyan	Cyan	Cyan	Cyan			Cyan						

References:

1. Apple, iOS Reference Library,
<http://developer.apple.com/library/ios/navigation/>
2. A. Deguet, R. Kumar, R. Taylor, and P. Kazanzides, "The cisst libraries for computer assisted intervention systems," in MICCAI Workshop on Systems and Arch. for Computer Assisted Interventions, Midas Journal, Sep 2008.
3. M. Henning, M. Spruiell, Distributed Programming with Ice,
<http://www.zeroc.com/doc/ice-3.4.1-IceTouch/manual/>
4. M.Y. Jung, G. Sevinc, A. Deguet, R. Kumar, R. Taylor, "Surgical Assistant Workstation (SAW) Communication Interfaces for Teleoperation.
5. P. Kazanzides, A. Deguet, A. Kapoor, O. Sadowsky, A. LaMora, R. Taylor, "Development of open source software for computer-assisted intervention systems," In MICCAI Workshop on Open-Source Software, Oct 2005.