ERC | CISST

MATLAB interface for *cisst* libraries

Group 16
Zachary Zhou
Anton Deguetz

Outline

- Introduction
  - Background, Motivation
- Goals
- Technical Approach
- Project Management
  - Deliverables/Milestones
  - Timeline
  - Dependencies

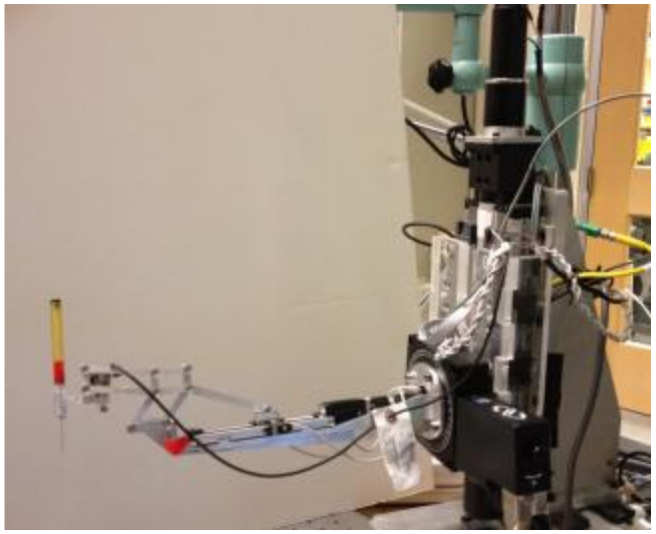
Background

- What is *cisst*?
  - "The *cisst* package is a collection of libraries designed to ease the development of computer assisted intervention systems. The Surgical Assistant Workstation (SAW) is a platform that combines robotics, stereo vision, and intraoperative imaging (e.g., ultrasound) to enhance a surgeon's capabilities. The SAW package therefore consists of implemented components (e.g., interfaces to many of the devices used for computer-integrated surgery) as well as reusable applications."

https://trac.lcsr.jhu.edu/cisst

What is *cisst* used for?

ERC | CISST



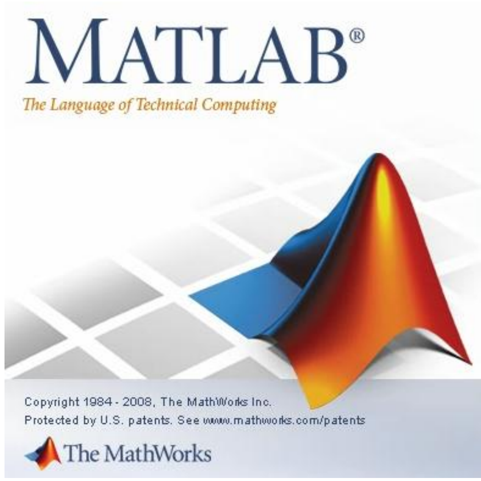
Why would we want to change *cisst*?

- Written in C/C++
  - Not everyone is proficient in C
  - Takes time to set up the cisst libraries
  - Requires some understanding of data types/structure
    - Ex: cisstVector

ERC | CISST

Why MATLAB

- User friendly
- No need to explicitly declare data types
- Good support for numerical methods
- Simple matrix manipulation
- Command console to try out code


MATLAB®
The Language of Technical Computing
Copyright 1984 - 2008, The MathWorks Inc.
Protected by U.S. patents. See www.mathworks.com/patents
The MathWorks

Project Goals

ERC | CISST

- MATLAB wrapper for *cisst* libraries
  - Be able to create cisst objects and manipulate them through MATLAB
- Utilize CMake to create plug-in library
- Handle data manipulation between C/MATLAB

ERC | CISST

Technical Approach

- Traditional methods:
  - Hard code from C to MATLAB
    - Tedious
    - Need to reflect changes to *cisst* SVN
  - Code generator
    - Potentially buggy
    - Needs to be updated

MEX files

ERC | CISST

- MATLAB includes the capability to call C methods via MEX files
- Requires recompiling C source code with the MEX compiler to generate a MEX file
  - Can be automated via CMake
- How will we know which methods to call?

cisst specifics

- All objects in the *cisst* library have a function which will return all functions in string form
- Use this function to send the names of all C methods to MATLAB

ERC | CISST

Approach

- Compile *cisst* C source code -> MEX files
- Obtain list of functions
- Dynamically generate MATLAB classes to handle *cisst* interface
- Handle sending of data between C/MATLAB

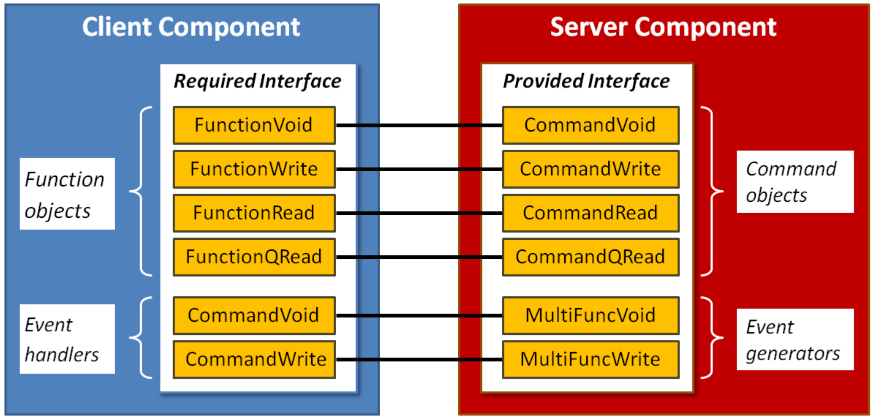
Expected usage

- Basic case:
  - Send string names through a generic function to call C methods
    - Ex: pos = cisstMatlab.Execute("daVinci", "PSM1", GetPositionCartesian");
- Prefered:
  - Dynamically create object variable
    - Ex: pos = daVinci.PSM1.GetPositionCartesian();

cisstMultiTask

ERC | CISST

- Component based framework
  - Need to provide support for required/provided interface
  - Handle function objects
- Potentially allow MATLAB to handle Events



Dependencies

- Regular contact with Anton
  - Resolve by: 2/20/2012
  - Status: Resolved
- Access/set-up to cisst packages and Cmake
  - Resolve by: 2/22/2012
  - Status: Resolved

Deliverables

ERC | CISST

- Minimum:
  - Be able to load a single component without configuration file onto MATLAB
  - Get dynamic loading to work
  - Write basic data conversion methods for native types
- Expected:
  - Utilize CMake to built MATLAB plug-in library
  - Create MATLAB object on the fly with string names
  - Populate MATLAB with component interfaces, names, and commands
  - Conversion methods for vectors and matrices
  - Proper documentation of completed portions
- Maximum
  - Conversion methods for composite types (cisstDataGenerator)
  - Test on multiple machines from MATLAB
  - Try running MATLAB wrapper from command-line
  - Extensive documentation/readme

Milestones

ERC | CISST

- Explore C/MATLAB interfaces
  - Complete by: March 1st
  - Status: in progress
- Dynamic loading working on cisst
  - Complete by: April 6th
- Data Conversion
  - Complete by: April 6th
- Use CMake to build plugin library
  - Complete by: May 1st
- Composite objects and populate MATLABinterface with interface names/components
  - Complete by: May 10th
- Documentation:
  - Complete by: May 10th

ERC | CISST

Timeline

Deliverables	20-Feb	1-Mar	9-Mar	16-Mar	23-Mar	2-Apr	6-Apr	13-Apr	20-Apr	27-Apr	4-May	10-May
Read/understand cisst library	🟨	🟦										
Explore MATLAB/C interfaces	🟨	🟨	🟦									
Call a C method from MATLAB	🟨	🟦										
Call MATLAB from C	🟨	🟦										
Pass Variables between C/MATLAB	🟨	🟨	🟦									
Dynamically create cisst objects				🟨	🟨	🟨	🟦					
Load single component on MATLAB				🟨	🟨	🟨	🟨	🟨	🟦			
Conversion of Basic Data Types				🟨	🟨	🟨	🟦					
Conversion of user defined types (cisstDataGenerator)				🟨	🟨	🟨	🟨	🟨	🟨	🟦		
Software Documentation			🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟦		
Final Report						🟨	🟨	🟨	🟨	🟨	🟨	🟦

🟨	In progress
🟦	Complete

ERC | CISST

References

- https://trac.lcsr.jhu.edu/cisst
- https://trac.lcsr.jhu.edu/cisst/wiki/cisstMultiTaskTutorial
- http://www.mathworks.com/support/tech-notes/1600/1605.html
- http://www.cmake.org/cmake/resources/resources.html

Thank you

Questions?