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?
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*
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
Project Goals

• 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
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

• 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
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);`

• Preferred:
  – Dynamically create object variable
    • Ex: `pos = daVinci.PSM1.GetPositionCartesian();`
cisstMultiTask

• 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

• 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 build 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

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

* #progress
* Complete
References

- https://trac.lcsr.jhu.edu/cisst
- https://trac.lcsr.jhu.edu/cisst/wiki/cisstMultiTaskTutorial
- http://www.cmake.org/cmake/resources/resources.html
Thank you

Questions?