

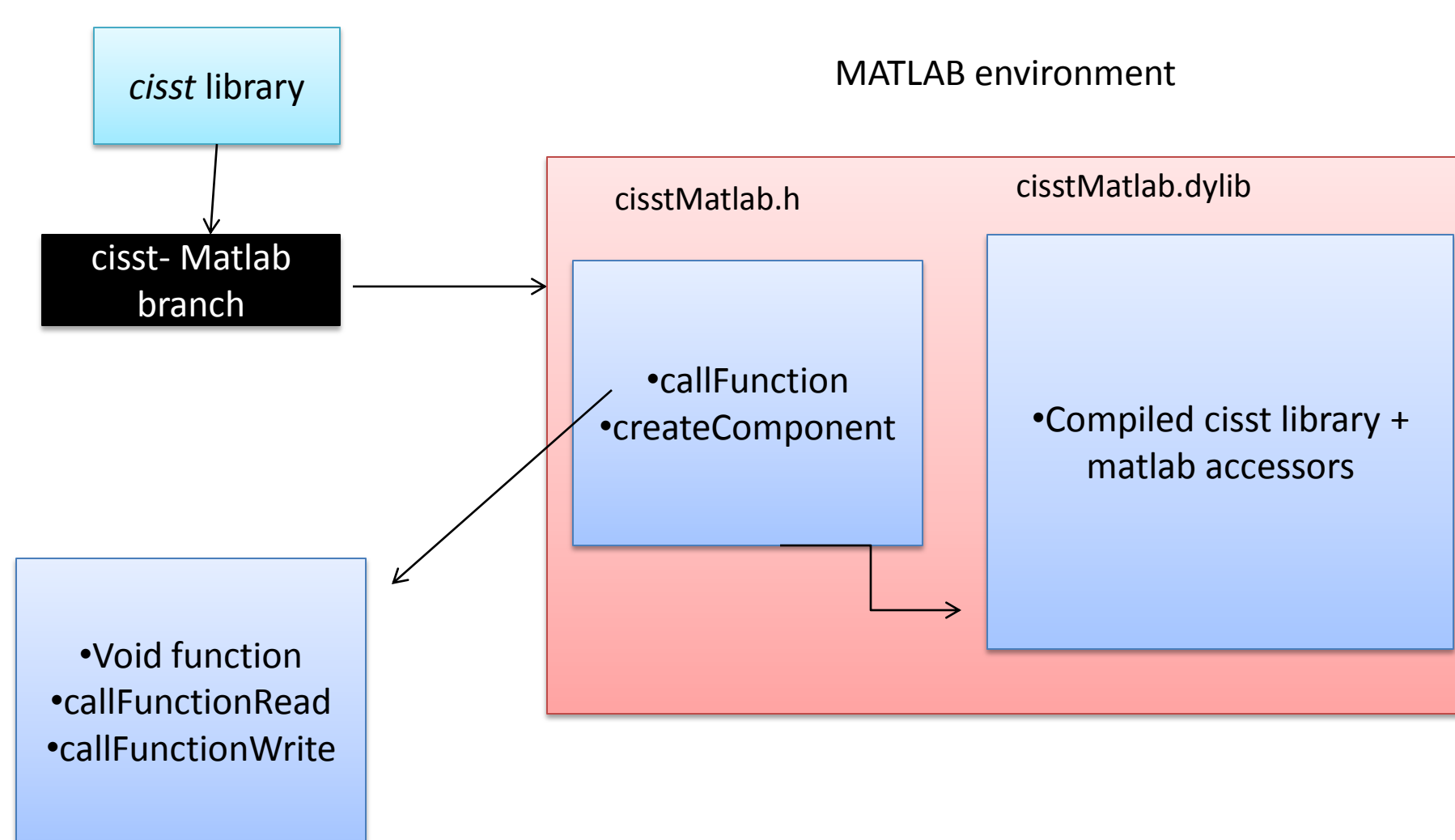
MATLAB interface for the cisst libraries

Zachary Zhou, under the instruction of Anton Deguet

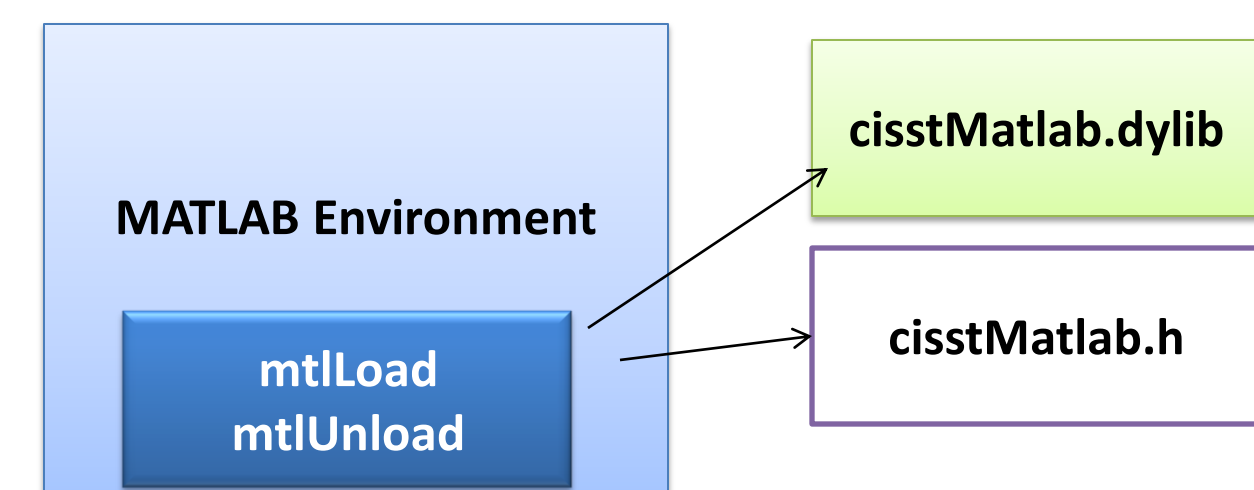
Introduction

- cisst is used to control many surgical devices
- MATLAB is a popular alternative language which is easier to learn
- There exist many routines in MATLAB to reduce data collected by surgical machines
- Desirable to create a wrapper of the cisst libraries

Approach



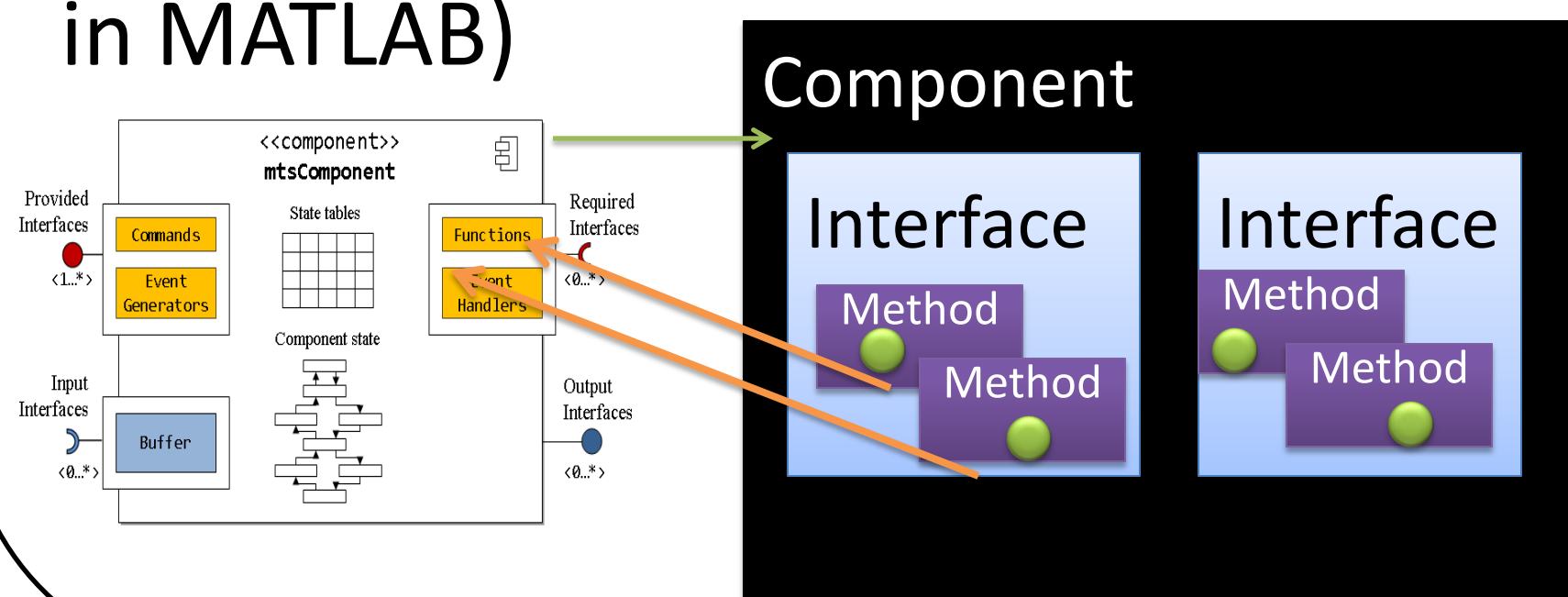
General



- Compile cisst libraries + c wrapper functions into *cisstMatlab.dylib*
- Copy *cisstMatlab.h* containing wrapper functions into working director
- Utilize small matlab script (mtlLoad/mtlUnload) to load library

Object Wrapper

- Each component stored as a dynamic prop
- Each interface is stored as sub-probs under the component
- Methods are stored as props under their interface (function pointer is stored in MATLAB)



Method wrapping

- Methods are accessed by calling the desired function under an interface:
`Result=ComponentA.interface1.DoSomething();`
- The prop is set to utilize MATLAB's *calllib* function
- The wrapper takes the function pointer and assists passing data to/from MATLAB depending on if it is a void function, accessor, or modifier.

Results

- cisstMatlab library and header file compiled via CMake
- DynamicProp created in MATLAB which corresponds to a running component in C
- Call to methods of the component via sub-props on the MATLAB environment

Future work

- Fix several pieces that are hard coded
- Provide support for additional events/components

Acknowledgements

- Anton Deguet for support and help on the project

References:

- <https://trac.lcsr.jhu.edu/cisst>
- www.cs.sfu.ca/~hamarneh/ecopy/medical_showcase2005a.pdf