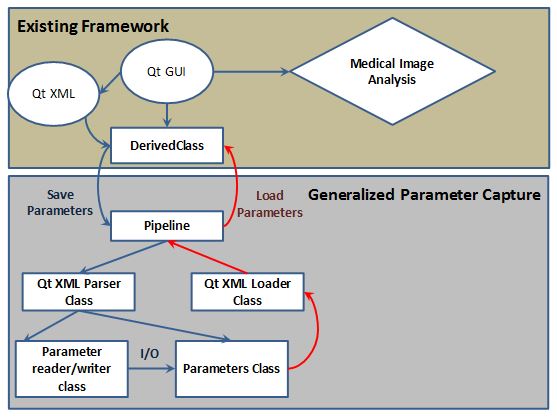
**Initial Design Document**

**Architecture Diagram**



Modules

The existing framework consists of a Qt GUI (used for Medical Image Analysis) that is described by a Qt XML Class. When the GUI needs to be saved or existing parameters need to be loaded, the DerivedClass calls the Pipeline Class, which is where Generalized Parameter Capture begins. It makes use of two classes, Qt XML Parser Class and Qt Loader Class for parsing the XML document and loading stored parameters respectively. The Qt XML Parser Class parses the Qt XML and generates a Parameters Class that stores a data representation of the GUI and a Parameter Reader/Writer Class that allows parameter values to be edited. When existing parameters need to be loaded into the GUI, the Parameters Class is sent to the Qt XML Loader Class, which in turn sends it to the Pipeline Class where it can be accessed by DerivedClass.

Data

The primary file that will be used is the XML file (which has the extension .ui in Qt). Our system will parse the XML file and generate two other files, one that acts as a data representation (Parameters) and one that allows the data to be edited (Parameters I/O). The Parameters I/O file will be editing values in the Parameters file.

Design Decisions

The primary dependency for our application is the Qt XML file, which needs to be fed in for parsing to take place. We have chosen to separate the saving and loading aspects of our application into different modules, named Qt XML Parser Class and Qt XML Loader Class respectively. In this way, each component can be modified and used independently of the others.

The functional spec goes into further detail on each module and details of implementation.

Link to the code repository: https://github.com/QTGUIFolks/QTGUI