Modular Expandable Programming Architecture for Interactive Web-Based EIT Data Exploration

Hervé Gagnon¹, Jeff Dawson¹, Andy Adler¹
¹Carleton University, Ottawa, Canada, adler@sce.carleton.ca

Abstract: We designed a modular and expandable programming architecture (PA) to favor code reuse and contribution. The PA is coded in Python, HTML, CSS and JavaScript languages while minimizing use of third-party solutions. Modules can be interconnected to provide custom interactive web-based solutions for EIT data exploration.

1 Introduction

EIDORS is a PA widely used to simulate EIT data and provide images and clinically-relevant information from simulated or real EIT data [1]. Ibex is a freely available software for EIT image analysis [2]. Most EIT equipment manufacturers also provide their own version of software for EIT data processing and exploration. However, no software tools exist to provide interactive web-based data processing and exploration for EIT while providing an open environment that can be fully customized to a user need. Such a PA was developed using an approach similar to the modular expandable multi-threaded architecture described in [3].

2 Methods

The PA is designed in Python, Javascript, HTML and CSS building on the strength of each language and minimizing requirements to third-party libraries. Each module provides an arbitrary number of inputs, outputs and timers that are defined in the Python language. The architecture is event-driven: whenever new data is received at an input or a timer expires, some processing occurs to generate data that are transmitted to the relevant outputs. The python language is also used to connect the output from one module to the input of another module through channels than can be buffered or not. The HTML, CSS and Javascript languages are used to provide the web-based graphical interface for the python modules. The graphical user interface is therefore simultaneously accessible from any standard web browser.

3 Conclusion

Several modules have been successfully realized to read, process and display EIT data using the described PA as shown in Figure 1. EIT images can be displayed and analyzed at real-time speed. The user can select different types of data processing, browse through the acquired data at his/her own rythm and explore the result of frame-by-frame, breath-by-breath or time-interval EIT data analysis.

References


Figure 1: Example of an application for EIT data exploration designed with the described programming architecture.