House in Your Head

Team G1-FrigidWaters

System Design Specification

Cycle # 1

Date: 11/10/2014

Team Members:

Name: Samuel Bever

Name: Michael Conway

Name: Joseph Muoio

Name: Kyle Patron

Name: Kevin Zakszewski

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the data, interface, architectural, and component-level design for our project. The architectural section will go over the modules and components, their relationship, and will describe the structure of the system as a whole. The interface section contains more detail on the system's user interface, data interface, and the programming interface. Finally, in the remaining portion of this document, any other design details or relationships will be laid out.

1.2 Scope

This system is intended to be a Brain Computer Interface that will act as a bridge between data received from the Emotiv device and a home automation system. The system on the end of the Emotiv device and the home automation system are a a bit like black boxes. We are more concerned with taking the data, interpreting it correctly and triggering the correct actions in the home automation system. The functionality of the other two systems are out of our scope.

1.3 Definitions, Acronyms, and Abbreviations

EEG Can refer to:

- Electroencephalography - Recording of the brain's electrical activity
- Electroencephalogram - The device that is used to record the brain's electrical activity

Emotiv The electroencephalogram hardware device, created by Emotiv Limited, used to read the user's brain activity (EEG)

Brain Computer Interface (BCI) The class of devices that the Emotiv belongs to

Amyotrophic Lateral Sclerosis (ALS) A neurodegenerative disorder that our target users suffer from. The main characteristics of ALS that we are concerned with in the scope of this project are the limited movement and mobility to complete paralysis.

API software level functions which allow other parts of the system (or external systems) to interact and share data.

1.4 Overview

The next section is the architectural description. In that section you will find overview of components and modules as well as a description of the system structure and component relationships. In the third section, the system interfaces are laid out. This contains user interface, data interface, as well as the programming interface. The fourth section is composed of a subsections describing the detailed design of the system. The fifth section looks at the relationship of the system to other products. The sixth section details the design decisions and tradeoffs of the system. The final section shows some rough psuedocode of the main components of the system.

1.5 Requirements Traceability Matrix

Requirements Identifiers	SR105	SR110	SR115	SR120	SR205	SR210	SR215	SR220	SR225
1.0	X				X	X			
2.0									X
2.1									X
2.2									X
3.0					X		X	X	
4.0		X	X	X					

2 Architectural Description

2.1 Overview of modules / components

Figure 1: High level DFD

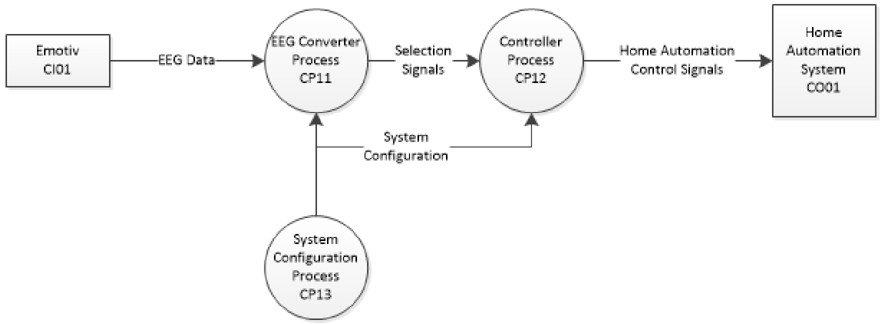

Figure 2: High level DFD with the main process broken out

Figure 1 indicates the two external systems that provide data to or accept data from our product. Our system accepts data from the Emotiv device through their driver and output controlling signals to the home automation system.

Figure 2 is a broken-out version of Figure 1. The System Configuration Process (CP13) must deal with system configuration. This involves reading it, modifying it, writing it back for later use and providing it to the other parts of the program. The EEG converter process (CP11) takes in the EEG data and is responsible for getting a clear signal out of it. The Controller process (CP12) is responsible for storing the current state of the system, transitioning states when appropriate and sending signals to the home automation system that match the state.

2.2 Structure and relationships

2.2.1 Component CI01

This component is the brain computer interface that provides an EEG signal into the system to control it.

2.2.2 Component CP01

This represents the totality of our product. It performs every action and completely controls the house given the EEG as input.

2.2.3 Component CO01

This is the home automation system. When it receives signals, it changes a physical property of the house, such as whether a light is on or off, or what channel a stereo is tuned to.

2.2.4 Component CP11

This is the process responsible for converting the large amount of data of the EEG into a more understandable signal. This will consist of only one bit at any given time.

2.2.5 Component CP12

This component translates the control signals generated in CP11 and uses them to control the house. It does this by storing the current state of the system and then transitioning when it receives the proper signals from CP11.

2.2.6 Component CP13

This is the process that is in charge of managing the system configuration. It reads in the configuration, process is, modifies it if necessary, makes it available to the rest of the system, and writes it back if modified.

3 Interface Description

3.1 User Interface

The user interface of the system is a rather simple one because most of the data we will receive and the choices the user will make are binary. There will be one type of screen where the user will cycle through the options of the objects that they can interact with in the home automation system. This is also the screen where the user makes the decision which object they want to interact with. When an object is chosen, they will be directed to the second type of screen. The second type of screen that handles the changing of the state of the object the user selects. From this screen the user can either change the state or return to the previous screen (the first type where the users cycle through the options). These two types UI of screen will make up the majority of the system. These designs will be modified or extra user interface designs will be implemented as necessary.

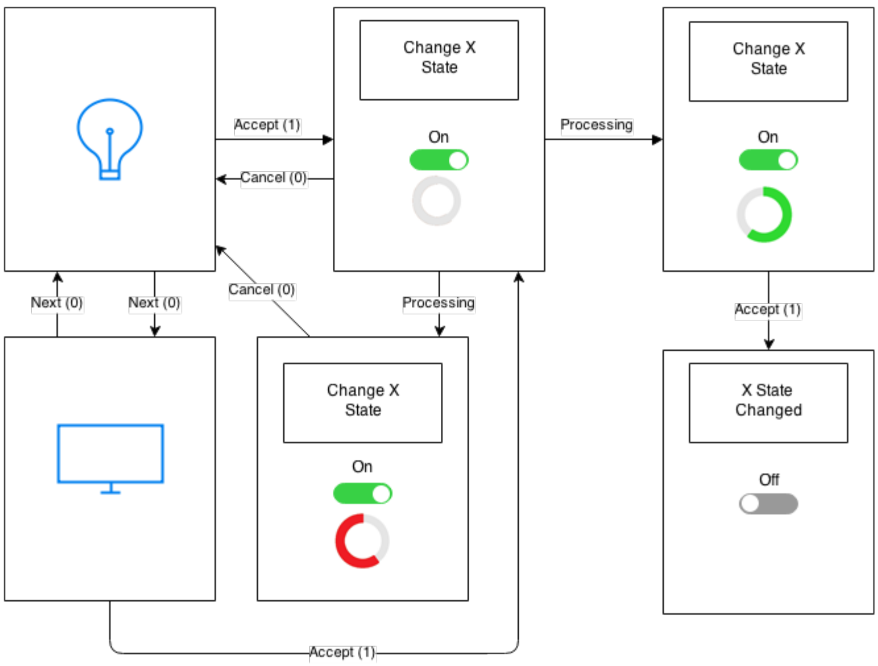

Figure 3: Sample User interface

3.2 Data Interface

The data flowing into our program will consist of a tuple of floats, representing the EEG data at any given time. There will also be a JSON blob that stores the parameters that should be used in signal extraction. The output will be in the form of calling methods of the home automation system. Figure 4 shows the internal representation of the user, house and state. This consists of a binary tree of UI structures, which may or may not reference a device to call the method of.

3.3 Programming Interface

There is no need for a programming interface because it is a generic, self-contained system.

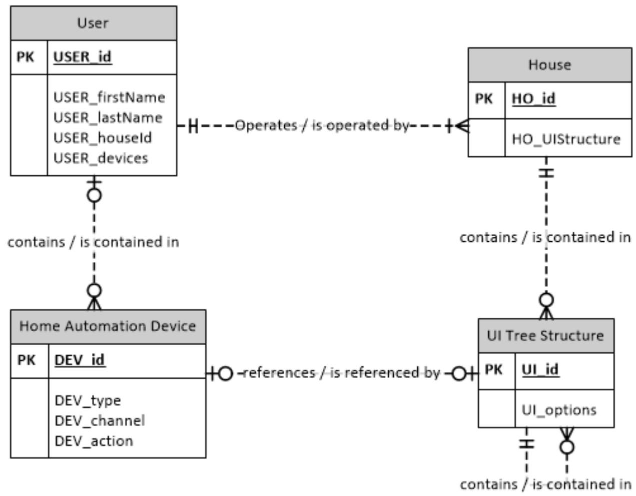

Figure 4: ERD

4 Detailed Design

4.1 Component Template Description

The design entities include the following:

Type	The type of Entity. This could be one of the following: Module, or Data Store.
Purpose	The reson the entity exists.
Function	What the entity does.
Subordinates	All entities that compose the entity.
Dependencies	The relationship of the entity to others.
Interface	How this entity interacts with other entities
Resources	External elements used by the system.
Processing	The algorithm used for the entity.
Data	View ERD for the data.

4.2 Design Entities

1.0 - EEG Interpreter

Type	Module
Purpose	Determines if the current brain scan information matches the trained data
Function	Uses statistical techniques to look at the current EEG information coming from the user's brainwaves and compares this to the data that was trained to see if it matches one of the trained classes.
Subordinates	None
Dependencies	Relies on Training datastore (2.2). There needs to be trained models to compare the stream of data to.
Interface	The EEG Interpreter will have an API that can be polled by other components.
Resources	The EEG information comes from the emotive device.
Processing	Various statistical techniques will be explored. At first, a simple algorithm that checks whether the EEG information falls within a given threshold to the training data, but more advanced techniques will be explored as well.
Data	See ERD in Section 3

2.0 - Training Module

Type	Module
Purpose	Trains a model for the current user for each state.
Function	Delegates to state trainers for each state that needs to be trained for the current user. Then stores this in the Training Datastore (2.2).
Subordinates	State Trainer (2.1). Every state that needs to be trained will be trained through a specific state trainer for that state.
Dependencies	Training Datastore (2.2). This is needed to save the trained models.
Interface	None
Resources	None
Processing	For each State Trainer (2.1) that is untrained, train the state model for that trainer. Then, save the results to the Training Datastore (2.2).
Data	See ERD in Section 3

2.1 - State Trainer

Type	Module
Purpose	Trains a model for a single state.
Function	Trains a model for a specific state. Each state trainer trains on one state, so there will be a State Trainer for the resting state, active state, and any other states deemed necessary in the future.
Subordinates	None
Dependencies	GUI Module (3.0). To train a model, interaction with the user of the system is required. This is done through a Graphic User Interface.
Interface	None
Resources	None
Processing	Interact with the user on the screen until a model is built based on the user's personal EEG.
Data	See ERD in Section 3

2.2 - Training Datastore

Type	Data store
Purpose	Holds the trained data for the current user.
Function	Holds all the trained states for the current user of the system. This includes the resting state and the active state at a minimum. Other states will be explored in the future.
Subordinates	None
Dependencies	None
Interface	The training datastore will have an API that allows read and write access to it.
Resources	May need hard disk space if it will be saved permanently.
Processing	The currently loaded user's data will be the only data retrieved. Loading from disk and saving to disk are allowed.
Data	See ERD in Section 3

3.0 - GUI Module

Type	Module
Purpose	Hands output from the system to an external monitor.
Function	All interaction with the user is through an external monitor. This system handles all output to the screen.
Subordinates	None
Dependencies	None
Interface	There will be a writer API to allow other parts of the system to output to the screen.
Resources	External Monitor
Processing	The GUI module will wait until given something to write to the screen. When it is given a request, it will write that request to the screen.
Data	See ERD in Section 3

4.0 - Home Automation Interface

Type	Module
Purpose	Interfaces with external home automation controls.
Function	Manages the system's interface with external home automation controls. This includes providing the rest of the system with information about available controls, maintaining any necessary state, and processing requests to manipulate the home via the controls.
Subordinates	None
Dependencies	None
Interface	The interface will support querying for information about available home controls and to manipulate the home via the controls.
Resources	Home automation system controls
Processing	The module will query control state as necessary and reject requests that are not valid in the current state.
Data	See ERD in Section 3

5 Reuse and Relationships to other Products

There has been previous related work done with the Emotiv device by related researchers. However, their interface was unappealing and their code messy. It may be referenced to help understand the Emotiv device, but it will not be reused directly.

There are very few commercially available BCI devices, and often there is no software available for them. The few that do exist are prohibitively expensive. Having the Emotiv available at a reasonable price will improve the quality of life for many ALS patients.

6 Design decisions and tradeoffs

The first design decision that was made was what type of interface to use. The options available were EEG sensor devices, eye movement trackers, and facial muscle detecting devices. EEG sensors seemed the best fit as they will work for all ALS patients, including those that are “locked in.” A locked-in patient may be unable to use their facial muscles or move their eyes. In addition, this means that a patient will not have to relearn how to use the system with new interfaces as their disease progresses.

Another design decision was what device should be used to best assist ALS patients. As noted in Figure 5, there already exist devices with similar functionality, but they are much more expensive. The Emotiv is $500, which is affordable when compared to its alternatives. The Emotiv is capable of detecting up to four thoughts plus a neutral state, which provides enough flexibility to design features in a number of ways.

Given this flexiblity, deciding how users should interface with the system was the next step. Previous user reviews indicated that training the Emotiv to recognize multiple thoughts was challenging, suggesting that a binary interface would be the most useful. On the other hand, an interface that uses a larger substantial branching would enable users to navigate it much more quickly. The binary interface was chosen in order to maximize ease-of-use.

The main purpose of the interface, home automation, was decided upon because it allows patients to regain capabilities that they have lost. It also serves as a proof of concept for interfacing with nontrivial systems.

7 Psuedocode for components

1.0 - EEG Interpreter

```
repeat forever:
    monitor EEG
    if EEG in a trained state:
        notify all registered listeners
        % TODO OR call all registered callbacks
```

2.0 - Training Module

Upon training request:

```
for each untrained State Trainer:
    run the trainer
save results to Training Datastore
```

Each State Trainer:

```
repeat until trained:
    display prompt
    collect data from EEG Interpreter
    update training parameters
```

3.0 - GUI Module

Upon menu display request:

```
% TODO OPTION A
display menu
for each menu option:
    register callback for corresponding state in EEG Interpreter
% TODO OPTION B
display menu
register self as EEG Interpreter listener for necessary states
upon state notification, perform action
```

4.0 - Home Automation Interface

Upon receiving a control request:

```
if request is valid:
    perform action
else:
    report error
```

Table of Contributions

	Section	Writing	Editing
1	2, 3.2	Kyle Patron	Michael Conway
2	1.5, 4	Joe Muoio	Kyle Patron
3	5, 6	Sam Bever	Kevin Zakszewski
4	7	Michael Conway	Sam Bever
5	1, 3.1	Kevin Zakszewski	Joe Muoio

I certify that:

- This paper/project/exam is entirely my own work.
- I have not quoted the words of any other person from a printed source or a website without indicating what has been quoted and providing an appropriate citation.
- I have not submitted this paper / project to satisfy the requirements of any other course.

Signature: Samuel Bever Date: 11/10/2014

Signature: Michael Conway Date: 11/10/2014

Signature: Joe Muoio Date: 11/10/2014

Signature: Kyle Patron Date: 11/10/2014

Signature: Kevin Zakszewski Date: 11/10/2014

Grading

The grade is given on the basis of quality, clarity, presentation, completeness, and writing of each section in the report. This is the grade of the group. Individual grades will be assigned at the end of the term when peer reviews are collected.