SIS Project Report

Hanzhong Zheng

August 8, 2018

1 BrainWave Component

The BrainWave Component is a new component that I implemented in the SIS system. The BrainWave component is developed in Android studio and implemented by Java programming language. The BrainWave component is designed form Android platform. Figure 1. shows the user interface of the BrainWave component. The role of BrainWave component includes collect user's data, process the data, and send the data to the uploader.

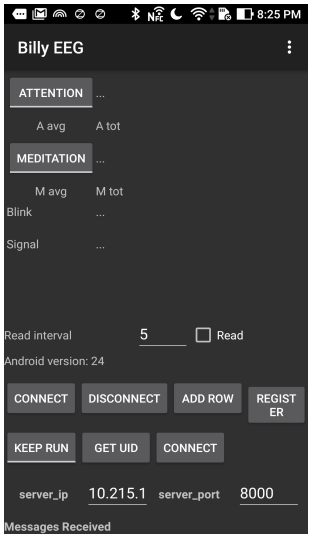

Figure 1: A BrainWave Component in SIS System

1. Collecting the Data: the brainwave data is collected by the sensor called Mindwave, which is a brainwave detection headset. The BrainWave component establishes a connection with Mindwave headset through Bluetooth. Then, the brainwave data is collected by the Mindwave headset and transmitted to the BrainWave component through the Bluetooth connection.

2. Processing the Data: After the BrainWave component received the raw brain wave data, it uses EEG algorithms developed by NeuroSky to process the raw brain wave data. The outputs of the algorithms are attention level and mediation level. For attention level, it is calculated by the Attention Meter algorithm that can indicate the intensity of mental "focus" or "attention". The value ranges from 0 to 100. The attention level increases when a user focuses on a single though or external object, and decreases when distracted. For mediation level, it is calculated by the Mediation Meter algorithm that can indicate the level of mental "calmness" or "relaxation". the value ranges from 0 to 100, and increases when users relax the mind and decreases when they are uneasy or stressed.

3. Data Transmission: the BrainWave component sends the mediation level and attention level information to the uploader component, which is responsible for strong the brainwave information to the Chronobot database through PHP. A simple brain wave record includes fields such as "row ID", "User ID", "date time", "source", "value", "type", and "originator". The "source" indicates which component sends this record. The "type" is the the attention level and the "value" is the mediation level. The "originator" indicates which phone collects this record.

2 Downloader

The Downloader is another new component that I implemented and added to the SIS system. Figure 2 shows the Downloader component. The button "DATA" and "SENDDATA" implement the function of acquiring user's brainwave data from Chronobot database and send the data to the other components in the SIS system.

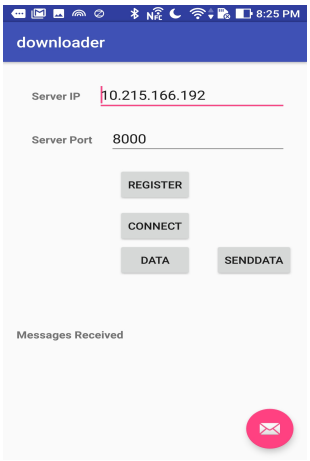

Figure 2: The Downloader component in the SIS system

The Downloader component plays the opposite role as the Uploader component. To be more specific, the Uploader component stores the user's information into the Chronobot database, while the Downloader acquires the information from the Chronobot database. In a practice system, the Downloader plays a role of end-user, or administrator. The major role of the Downloader is to obtain the current user information from the Chronobot database. To be more detailed, in the Chronobot database, a table relation called, *msgTOsis* holds user information and unprocessed sensor reading data, indicated by the flag field. Then, the Downloader is able to extract the user id information for BrainWave Component based on the flag and the component name. It looks the latest request from the table where the flag field is 0 and the source is BrainWave and extract the user id. After that, the Downloader sets the flag field to 1 to indicator that this request has been processed.

After the Downloader the extracted the user id information, it will immediately sends to Uploader and then, the Uploader will add the user id information to the brain wave data and stores the final record into the Chronobot database.

3 InputProcessor

I extended some extra funationality to InputProcessor component that allows the abnormal event analysis can done at this component. Figure 3 demonstrates the user interface of the InputProcessor. The name of the button indicates their functionalities. The "Download Data" button downloads all the user's data. and the "Event Analysis" button calls the algorithm of identifying the abnormal event. The current abnormal event analysis is able to obtain all the data regarding to the current user. However, there are still some limitations about using all the data regrading to the current user. For example, if the data is across a very long period of time, then the previous data may influence the abnormal event analysis. I propose to only download the recent data regrading to the current user and apply these data to the abnormal event analysis.

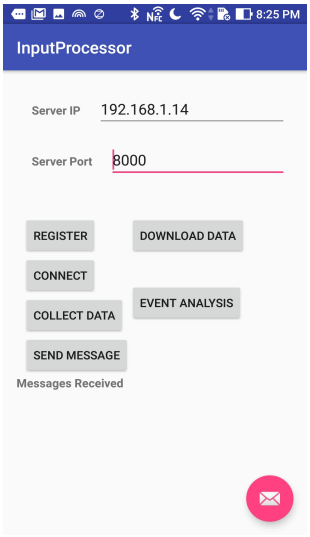

Figure 3: The InputProcessor component in SIS System.

The event analysis functionality can be divided into 3 progresses:

1. Downloading Data: the InputProcessor firstly downloads all the records for current user, including both the attention level and mediation level.

2. Calculating abnormal event: the abnormal event calculation is based on the implementation of the following algorithm:

   (a) Firstly we define: for any attribute χ,

   $$d_{max(\chi_{ij})} = max_{1 \leq i,j \leq m} \|v_i[\chi] - v_j[\chi]\| \tag{1}$$

   , where m is the total number of records. $d_{max(\chi_{ij})}$ denotes the maximum o the distance between the values of any two tuple v_i, v_j in the attribute χ, where m is the number of the tuples during $\Delta T(\eta)$, and

   $$d(v_i[\chi], v_j[\chi]) = \frac{|v_i[\chi] - v_j[\chi]|}{d_{max(\chi_{i,j})}} \tag{2}$$

   is a distance function.

   (b) According to the above distance function, for the attribute *type*, for simplicity, we denote it as TP, have

   $$d(v_i[TP], v_j[TP]) = \frac{|v_i[TP] - v_j[TP]|}{d_{max(TP_{ij})}} \tag{3}$$

   $$d_{max(TP_{ij})} = max_{1 \leq i,j \leq m} |v_i[TP] - v_j[TP]| \tag{4}$$

   , and for the attribute *value*, for simplicity we denote it as VL, we also have

   $$d(v_i[VL], v_j[VL]) = \frac{|v_i[VL] - v_j[VL]|}{d_{max(VL_{ij})}} \tag{5}$$

   $$d_{max(VL_{ij})} = max_{1 \leq i,j \leq m} |v_i[VL] - v_j[VL]| \tag{6}$$

   (c) After we calculated the distance value for all tuples, we need to find two parameter $\gamma^{'}$ and $\gamma^{''}$. If we can find a pair of tuples that satisfies the condition that $d(v_i[TP], v_j[TP]) \leq \gamma^{'}$ and $d(v_i[VL], v_j[VL]) > \gamma^{''}$, or $d(v_i[TP], v_j[TP]) \geq \gamma^{'}$ and $d(v_i[VL], v_j[VL]) < \gamma^{''}$, then we can say $v_i and v_j$ constitute a dependency violation event during this $\delta T(\eta)$.

   (d) For another parameter ϵ and under the condition of the parameters $\gamma^{'}$ and $\gamma^{''}$,

   $$\psi(\Delta T(\eta), TP < VL) \leq \epsilon \tag{7}$$

   , however,

   $$\psi(\Delta T(\eta), TP(v_i), VL(v_i)) > \epsilon \tag{8}$$

, then we can say v_i constitute an abnormal event. Similarly, if

$$\psi(\Delta T(\eta), TP(v_j), VL(v_j)) > \epsilon \tag{9}$$

, we can say v_j constitute an abnormal event during this $\Delta T(\eta)$

3. Displaying the abnormal event: Once the InputProcessor identifies all the abnormal events, it will display the information (E.g. Time. user information)in the screen so that the user can visualize them.

4 Uploader

I also extended the functionality of the Uploader component in the original SIS system. The Uploader is the bridge between the mobile SIS system and the Chronobot database. For Uploader, it adds the user information to the brainwave data so that we can distinguish the brainwave data among multiple users.

After it finishes inserting the user information into each brainwave record, then it uses the PHP to establish the connection with the Chronobot database and stores the brainwave data into the "record" relational table.

5 SISServer

SISServer is the service provider in the SIS system. For all other components of the SIS system, in order to perform their functionality, they must complete the actions: "Register" and "Connect" with the SIS Server. The service provided by the SISServer ensures the communication between different components inside the system. Besides, the actions in other components also require them to "register" and "connect" with the SISServer. For example, if the BrainWave component wants to send the brainwave information to the Uploader, it checks whether it has registered and connected with the SISServer first. Or, if the Downloader tries to acquire the current user information, it also checks whether it has registered and connected with the SISServer.

6 Summary

In conclusion, for this SIS system, I created two new components: BrainWave and Downloader, whose major responsibility is to collect the brainwave data (attention and mediation level) and acquire the current user information. I also extended the functionalities of two existing components: InputProcessor and Uploader. The new functionality of InputProcessor is to perform the abnormal event analysis in local machine. For Uploader, it can add the user information to the sensor data, directly connect with the Chronobot database and stores the record into the relational table.

However, there are still several problems inside the current SIS system. 1. communication latency: the communication between all the components in the SIS system is done through the socket communication, which has the big issue of the transmission latency. For example, the sampling time for brainwave data is irregular and there are some duplicate records at same time stamp. We propose to "windownize" the brainwave data by dividing the time series brainwave data into several windows and use the summary statistics to represent the records inside that window. Or, we also can manually preform data preprocessing procedure to clean the brainwave data. For an instance, we can manually eliminate the duplicate values at the same time stamp.