

『SILENCE』 커피메이커 시스템

Team 4

200611450 강세용

200611458 김영승

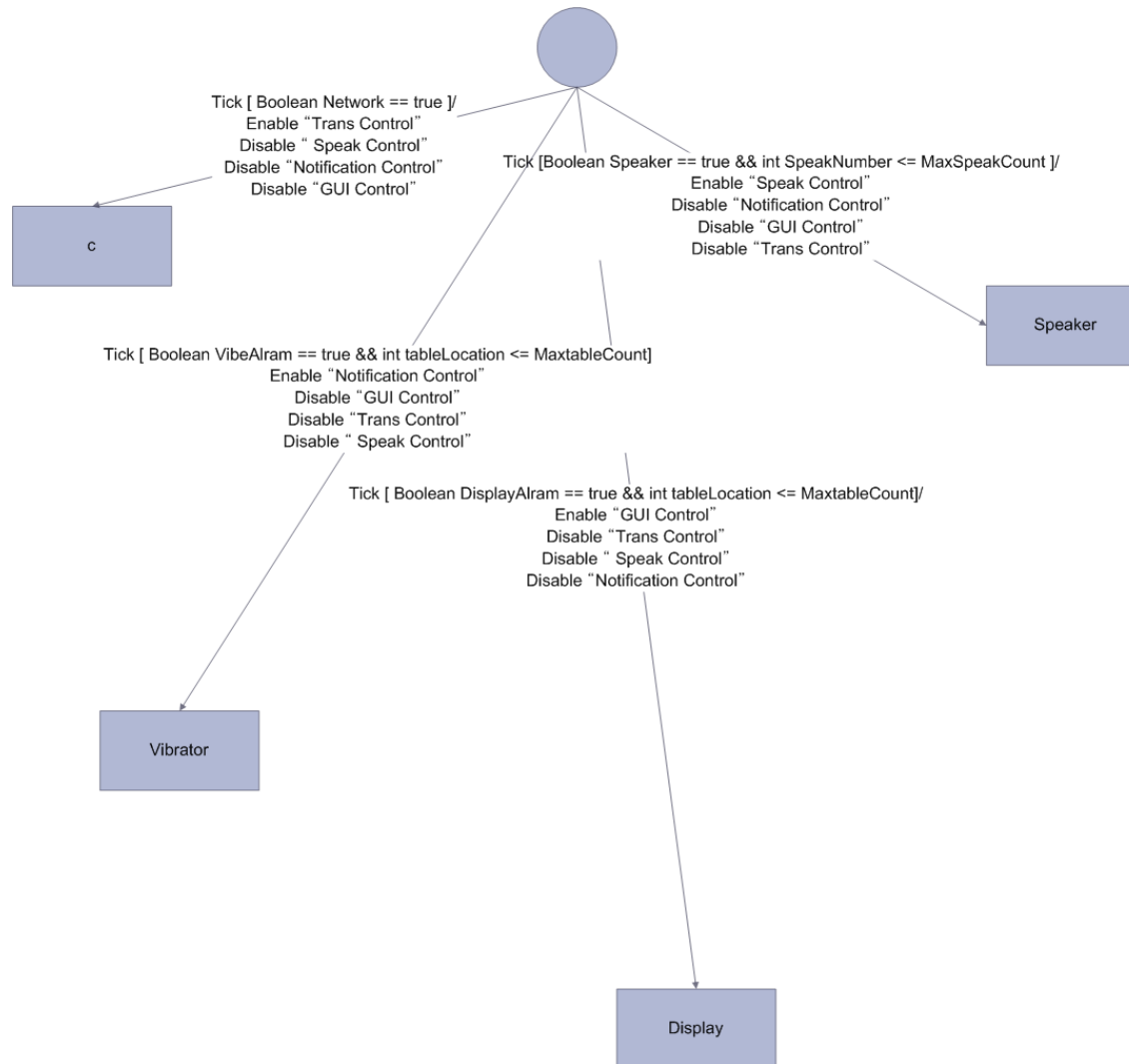
200611518 조민경

Index.

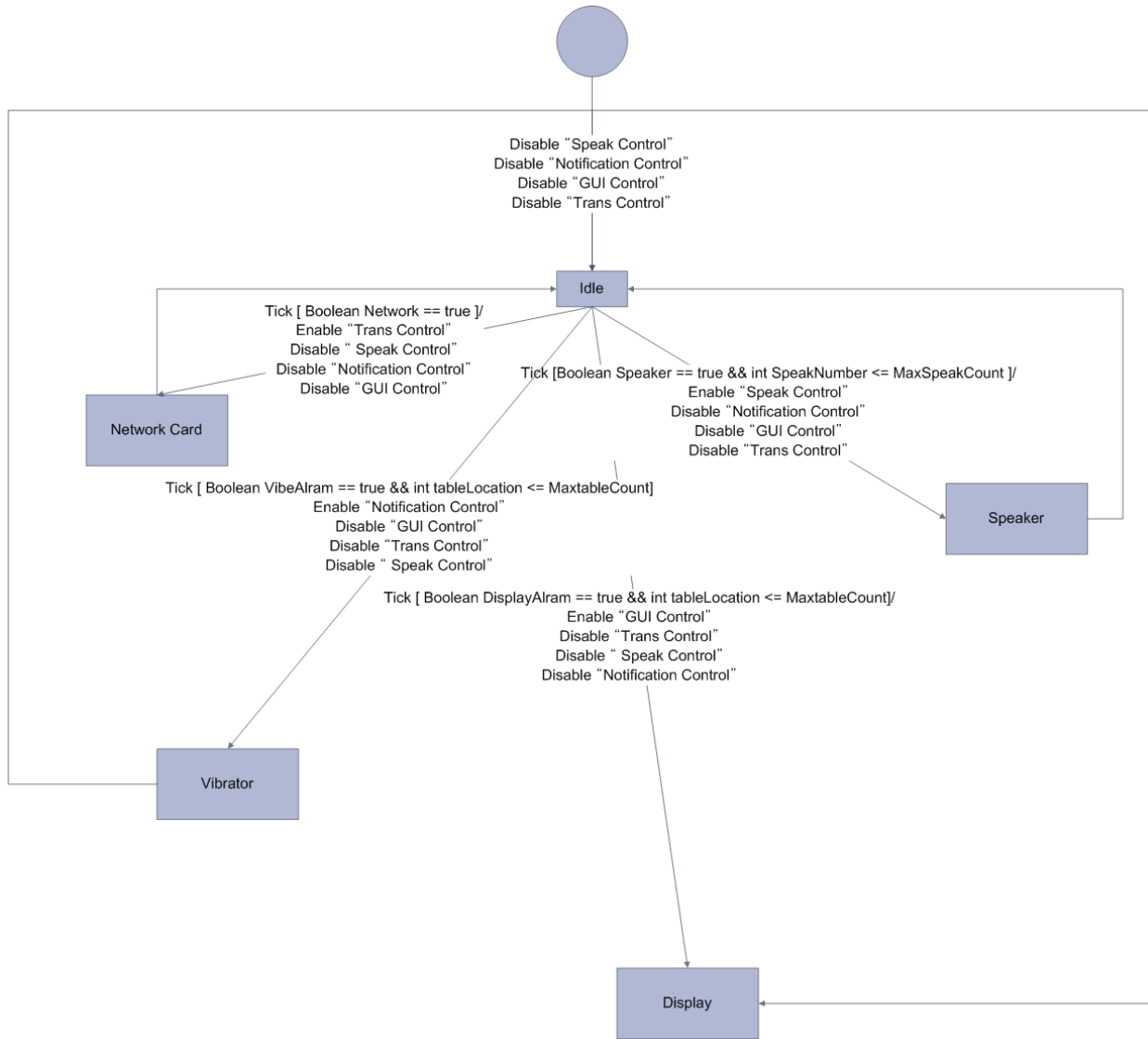
- SA에서의 수정사항
- Structured Charts
- Pseudo Codes

SA에서의 수정사항

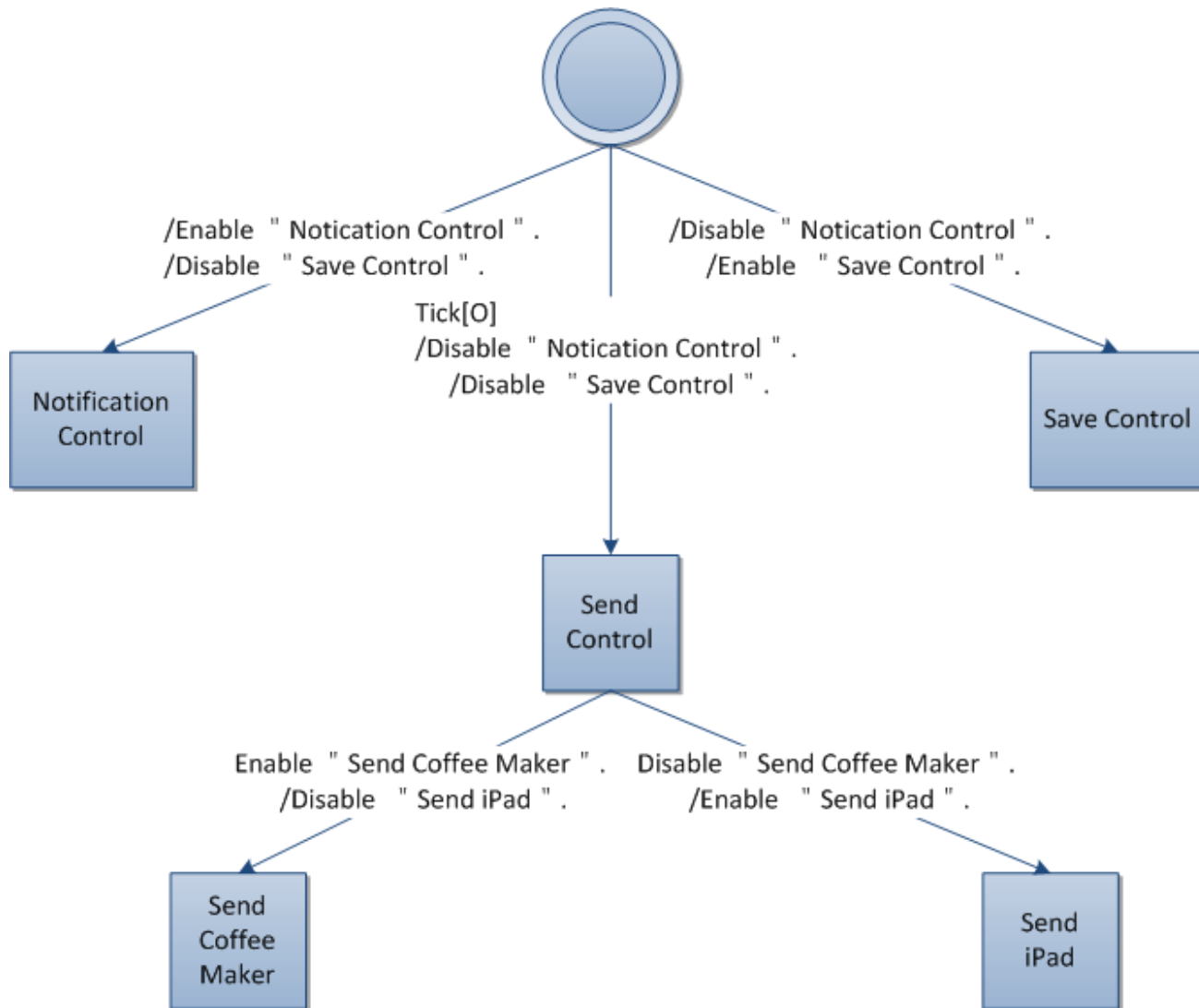
FSM - iPad



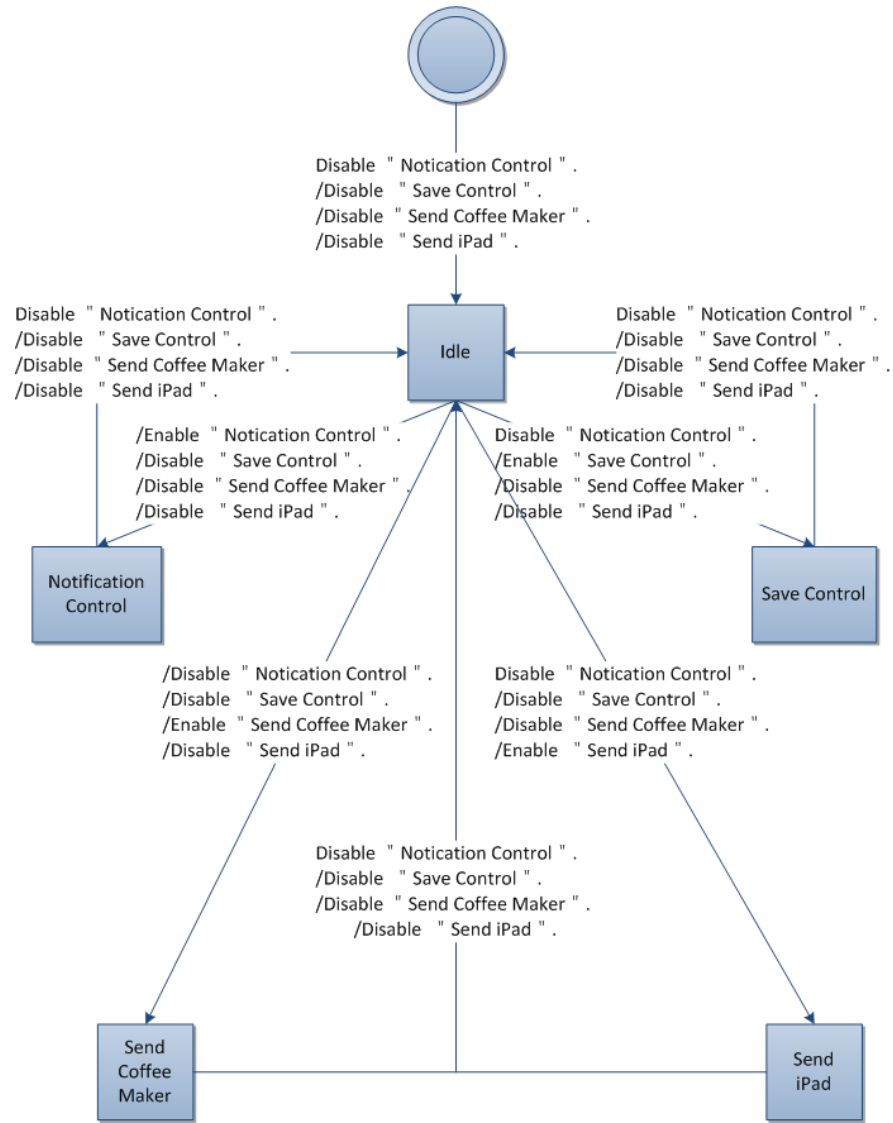
FSM - iPad



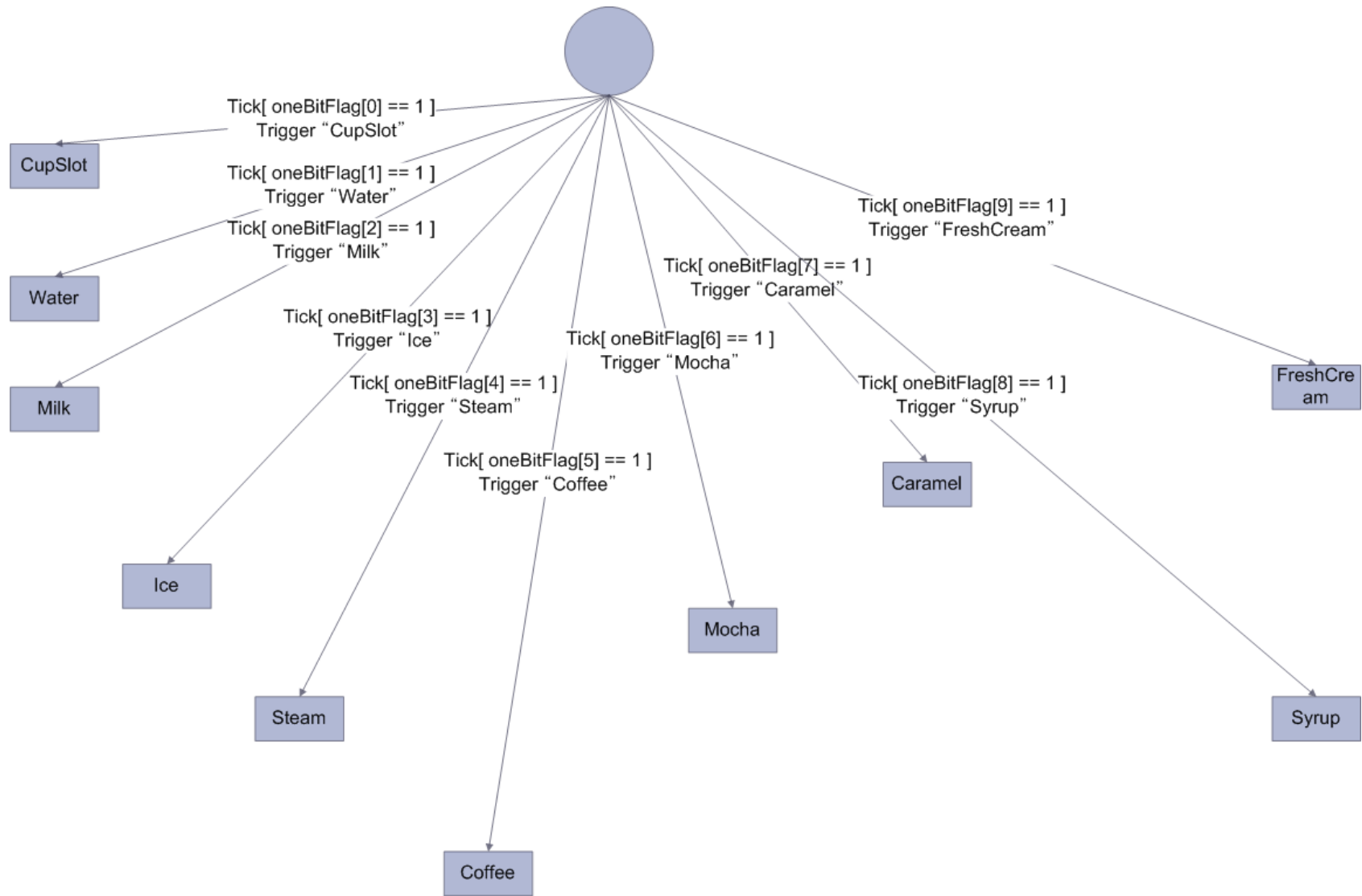
FSM - Server



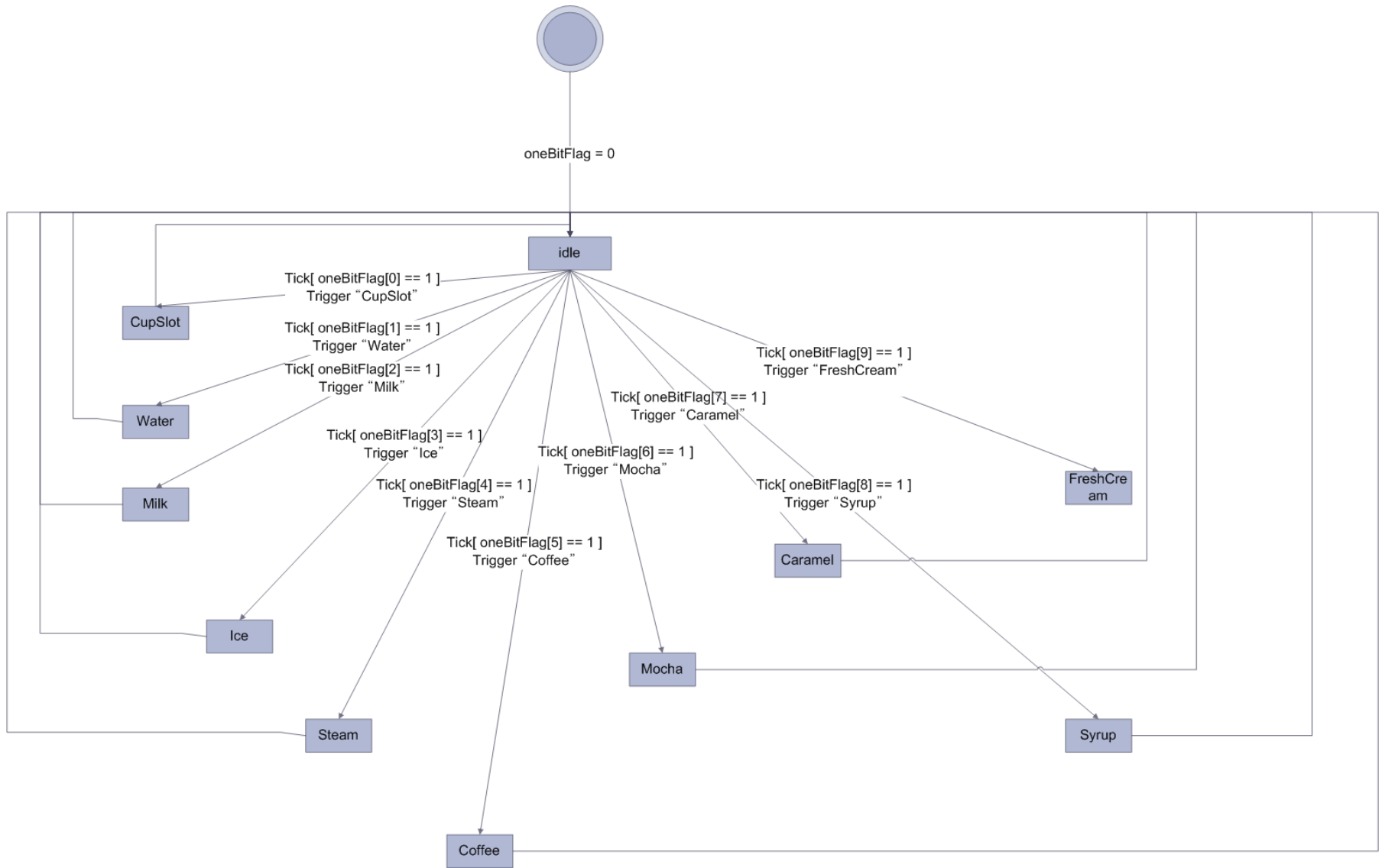
FSM - Server



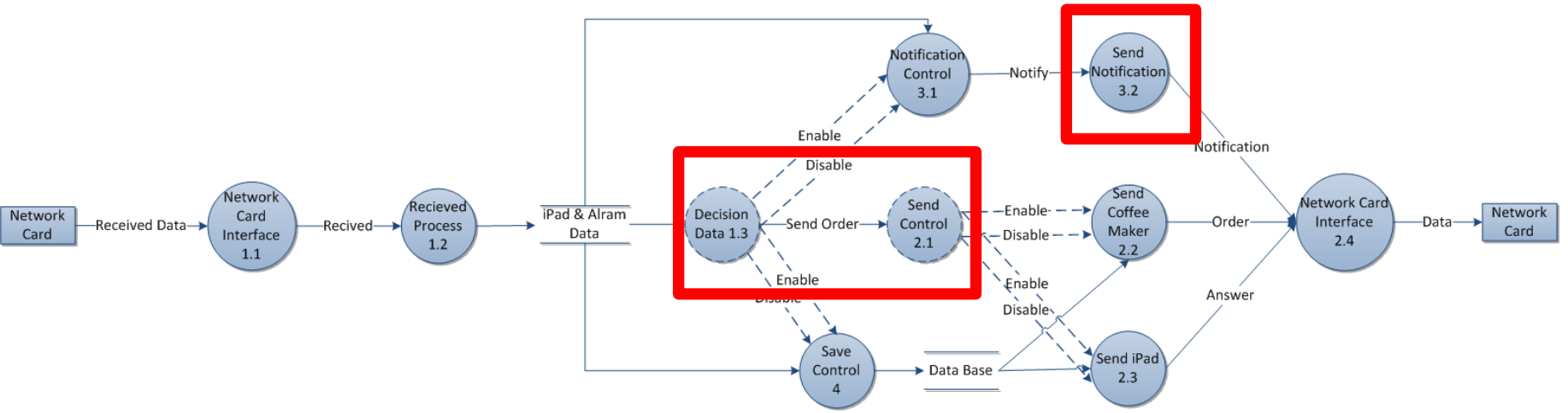
FSM - CoffeeMaker



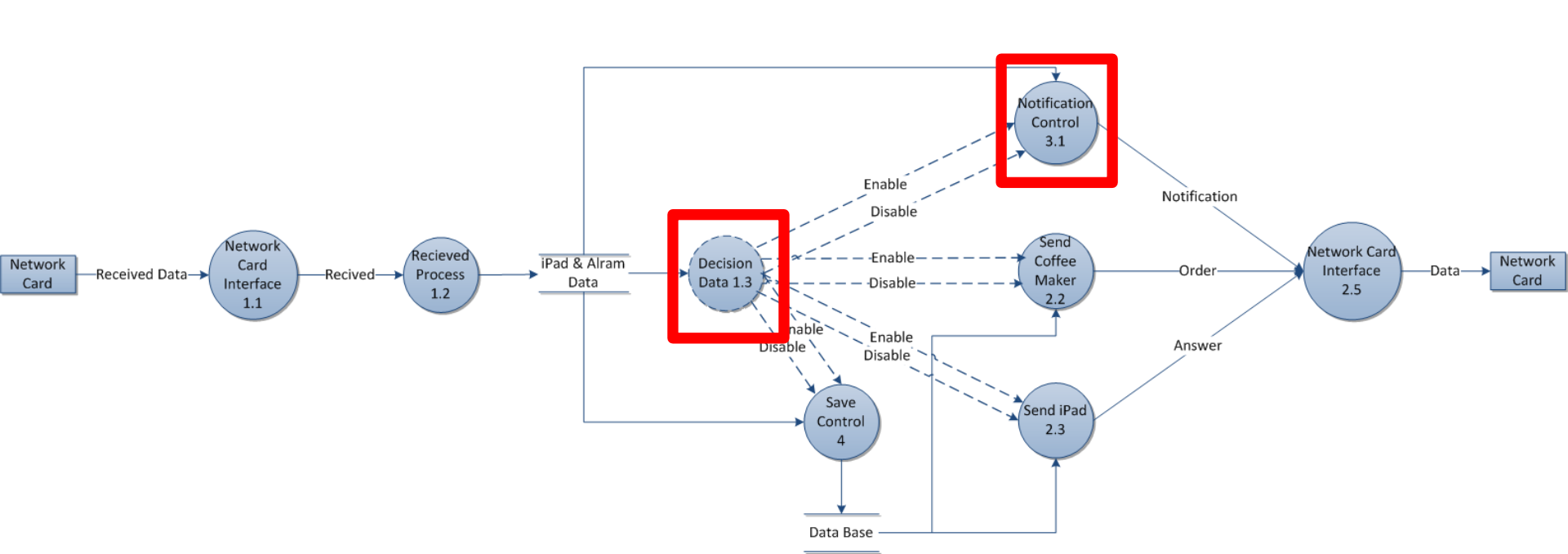
FSM - CoffeeMaker



DFD - Server

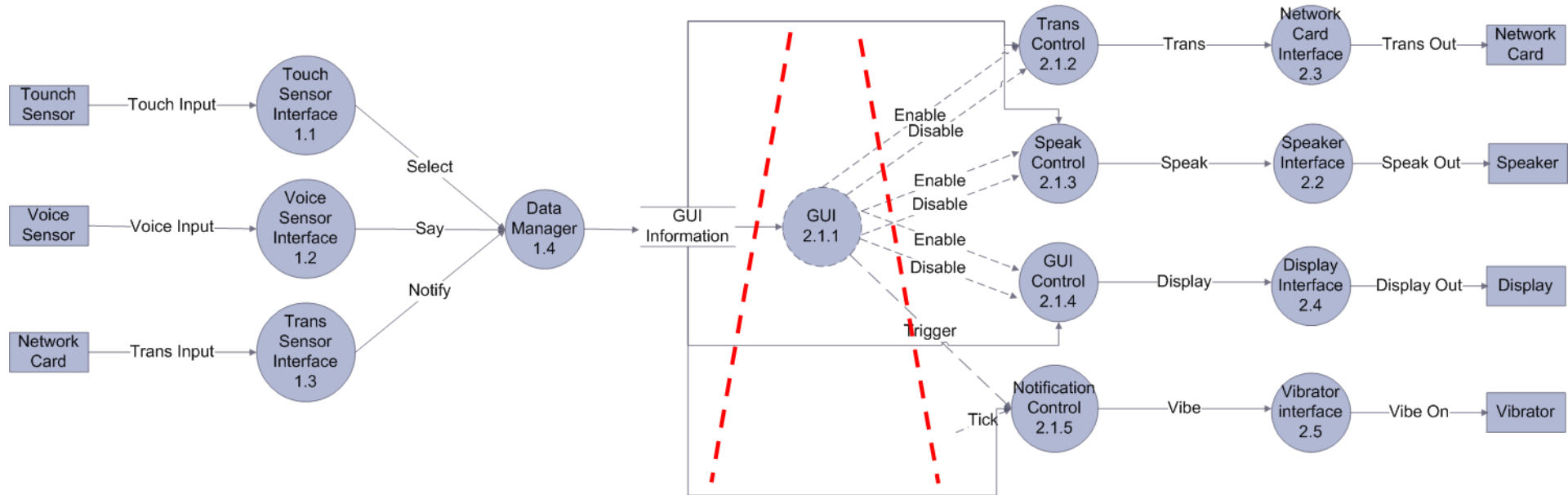


DFD - Server

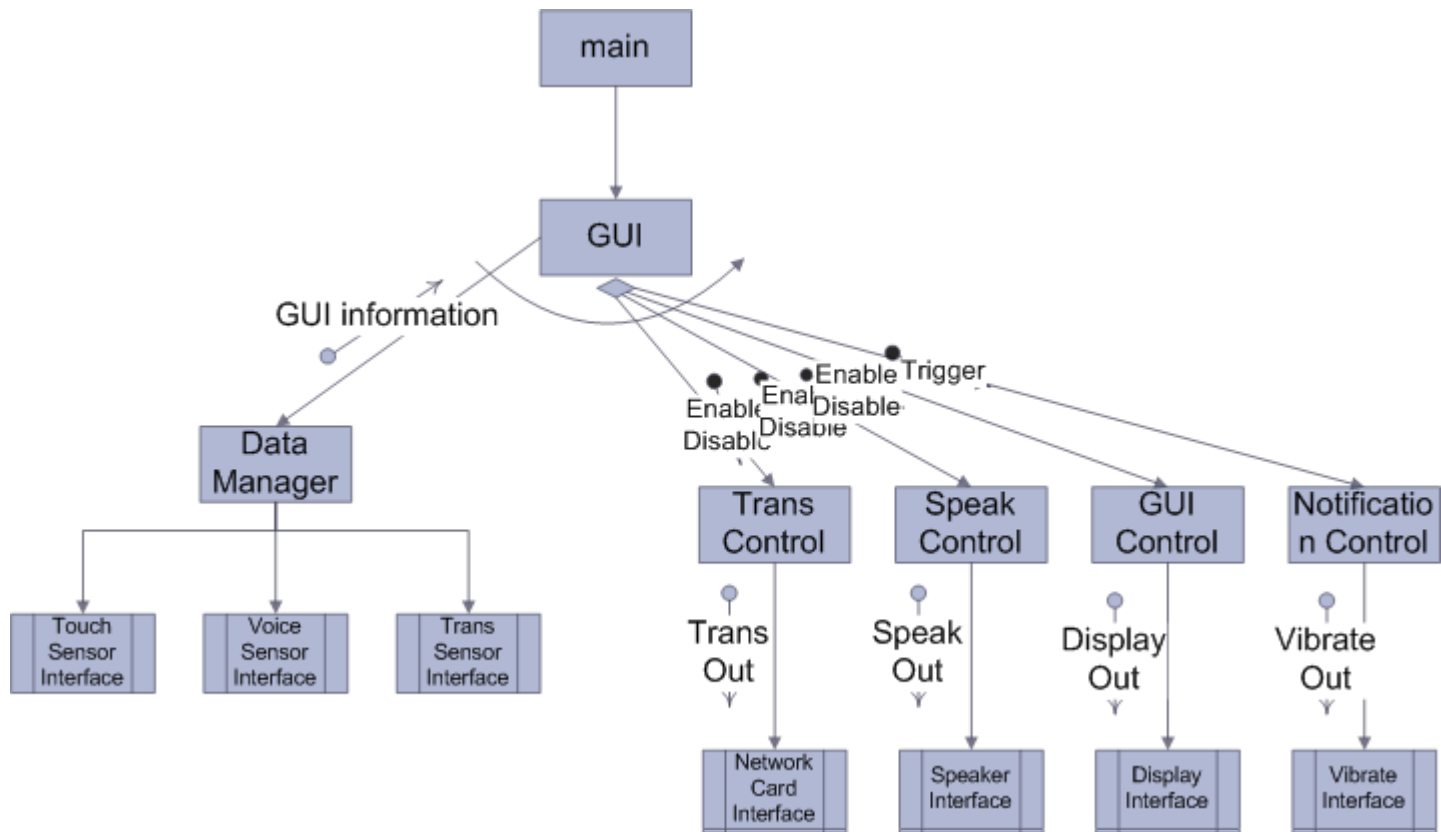


STRUCTURED CHARTS

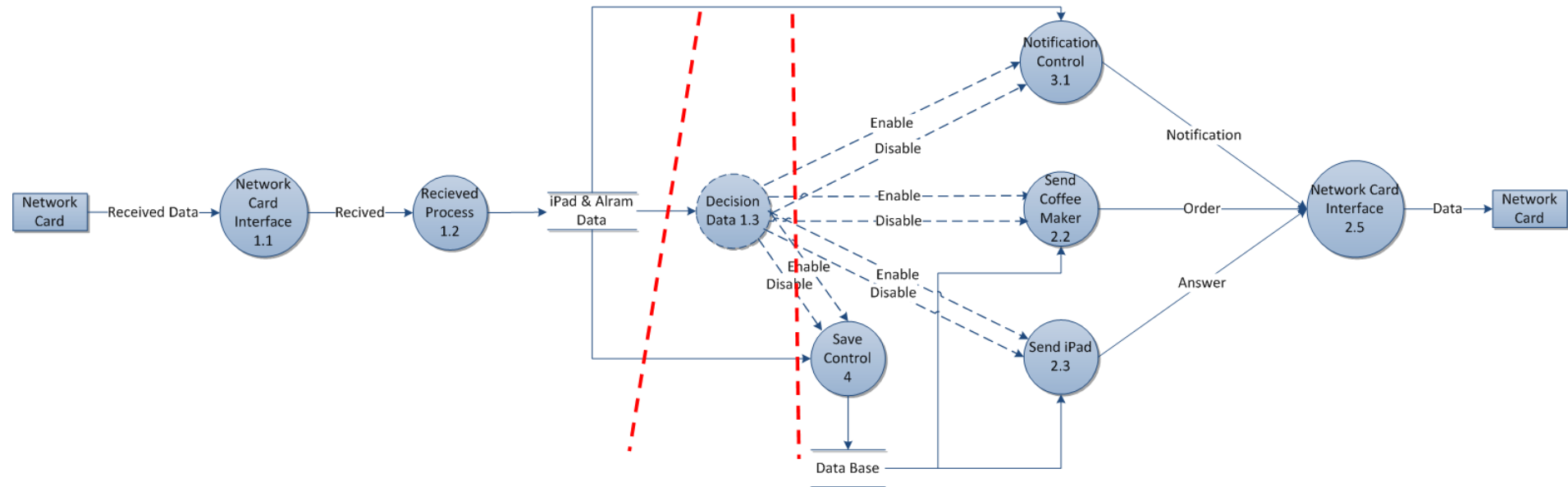
DFD - iPad



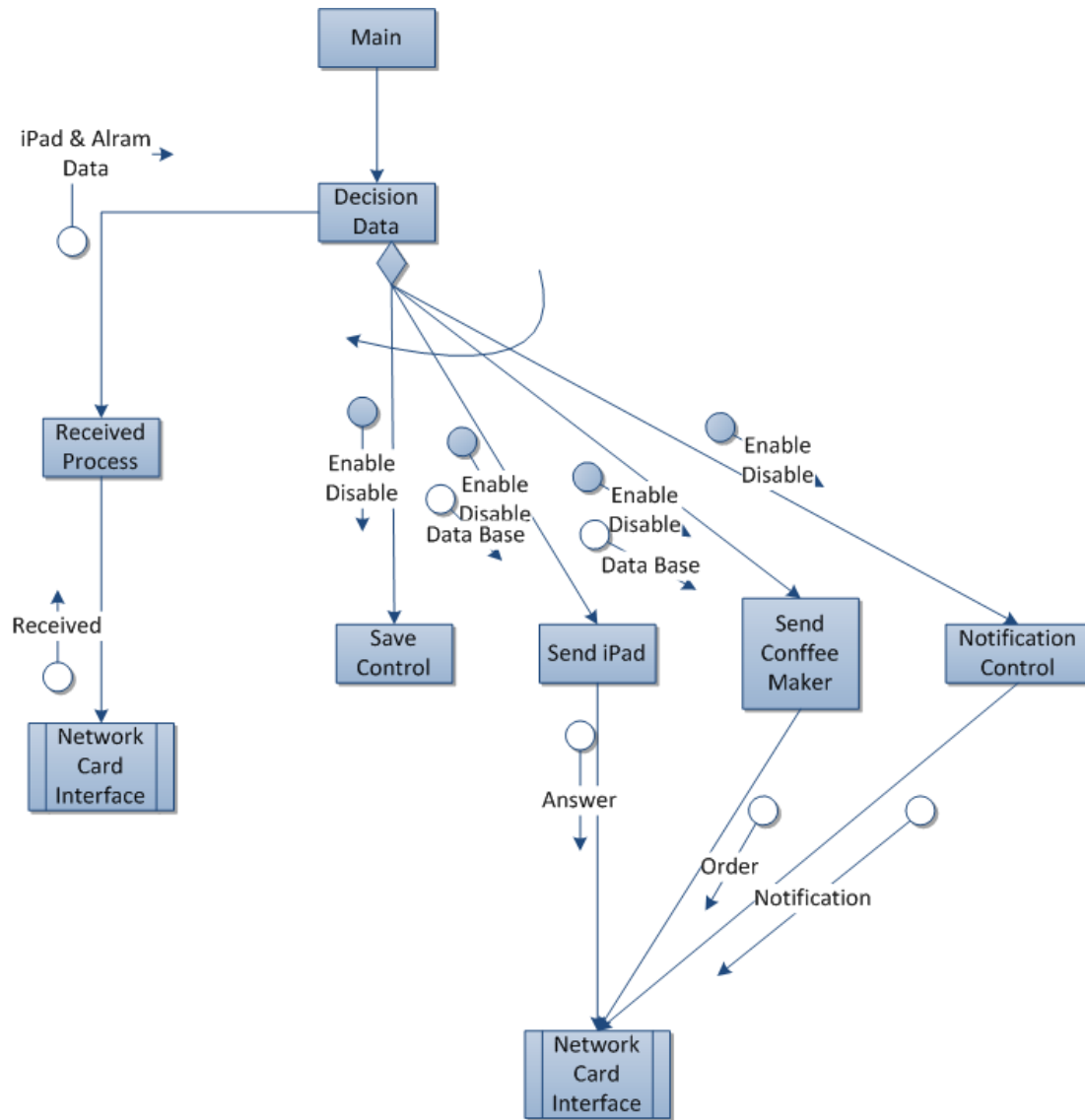
Structured Charts - iPad



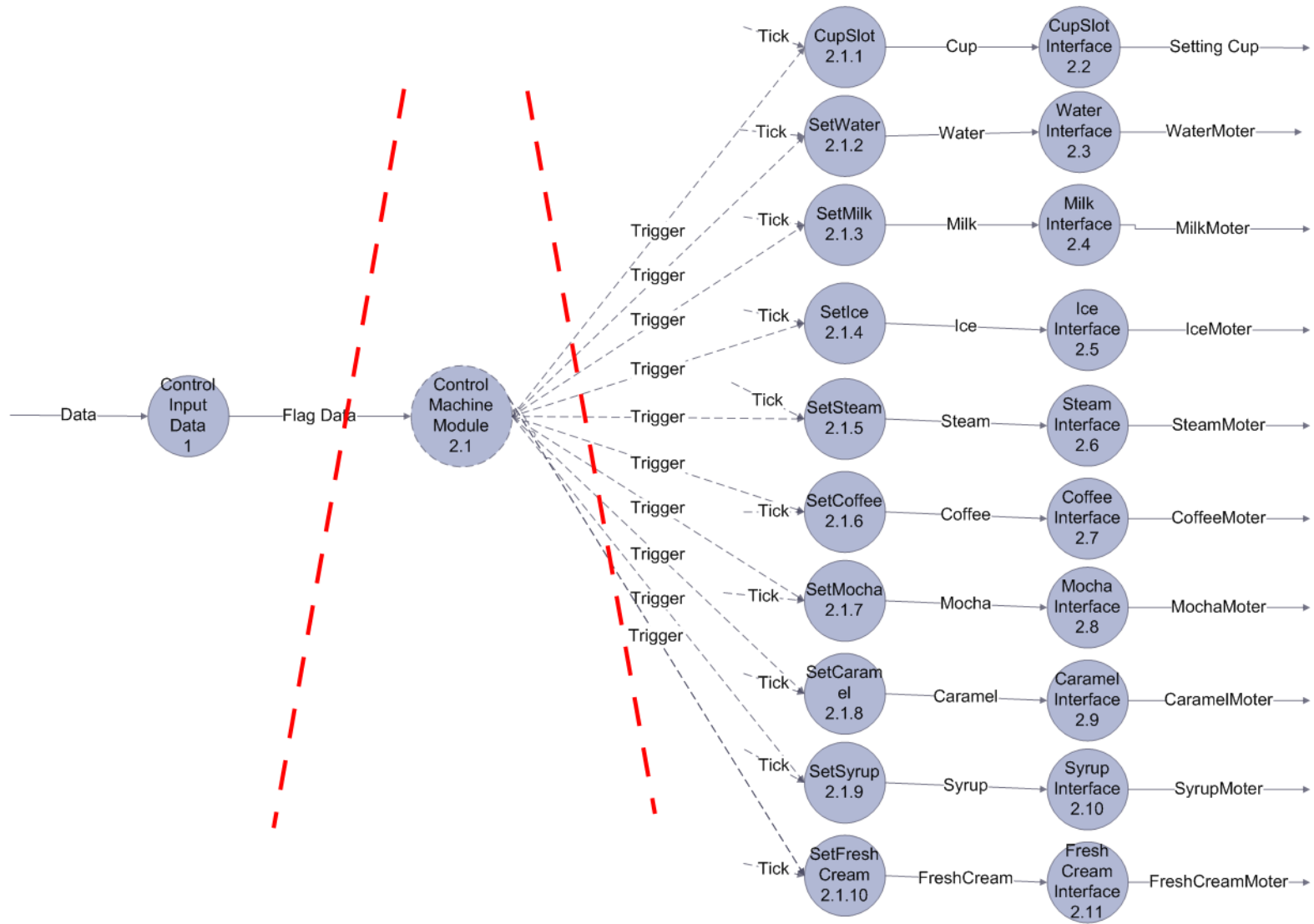
DFD - Server



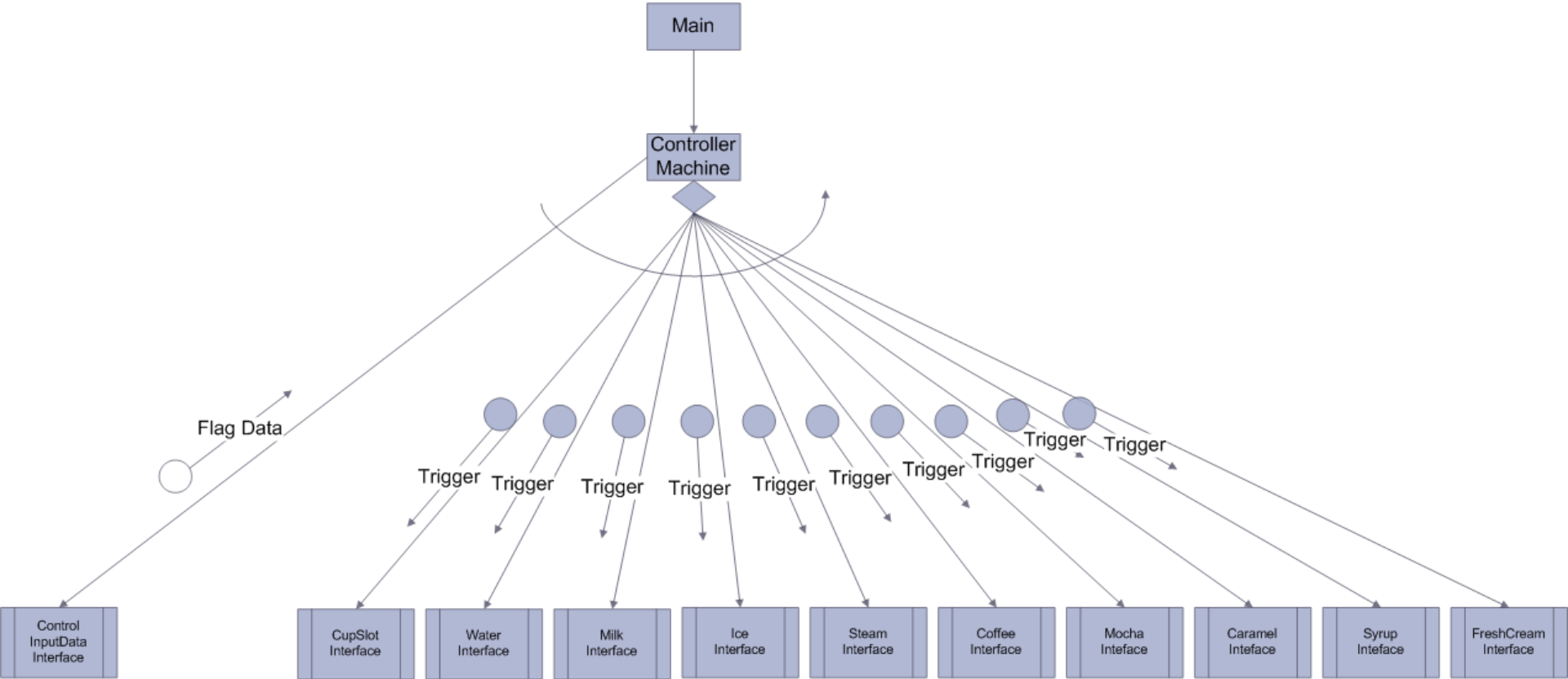
Structured Charts - Server



DFD – CoffeeMaker



Structured Charts – CoffeeMaker



PSEUDO CODES

Pseudo Code - iPad

```
void main(){
    int display, speak, network;
    int x = 0, y = 0;
    int voiceNumber = 0;
    Datastruct DS = null;
    DataStruct tempDS = null;

    while(){
        //Initialize
        DS = NULL;
        x = 0; y = 0; voiceNumber = 0; // Initialize Information
        tempDS = null; // Initialize Network Information

        //detected input information
        display = TouchSensor( int &x, int &y ); // Touch Information
        speak = VoiceSensor(int &voiceNumber); // Voice Information
        network = TransSensor( DataStruct &tempDS); // Network Information

        if(display || speak || network){
            // Input information Integration
            DataManager( x, y , voiceNumber, tempDS , &DS); }
        if( DS ){
            GUI( DS ); }
    }
}
```

```
void DataManager( int x , int y, int voiceNumber, DataStruct tempDS, DataStruct &DS){
    // Setting all information
}

void GUI( DataStruct DS){
    if( DS.trasControl ){
        /*send to NetworkCard interface*/
    }else if( DS.speakControl ){
        /*send to speaker interface*/
    }else if( DS.GUIControl){
        /*send to display interface*/
    }

    if(notificationControl){
        /*send to vibrate inteface*/
    }
}
```

Pseudo Code - Server

```
void main(){
    while(){
        Datastruct DS = null;
        DS = ReceivedProcess(); // Receive Data from network card
        if( DS ){
            DecisionData(DS); } // determine the data what controller selected function
        }
    }
}
```

```
Void DecisionData( DataStruct DS){
    if( DS.saveControl ){
        /*Save Control*/
    }else if( DS.sendiPad ){
        /*Setting Address iPad*/
        /*Send NetworkCard*/
    }else if( DS.sendCoffeMaker ){
        /*Setting Address CoffeeMaker*/
        /*Send NetworkCard*/
    }else if( DS.notificationControl){
        /*Setting AddressNotificationControl*/
        /*Send NetworkCard*/
    }
}
}
```

Pseudo Code – CoffeeMaker

```
void main(){  
  
    while(){  
        Datastruct DS = null;  
  
        DS = ControlInputData();  
  
        if( DS ){  
            // Machine control what to do  
            ControlMachine(DS);  
        }  
    }  
}
```

```
void ControlMachine(DS){  
  
    if( DS.oneBitFlag[0] ==1){  
        /*Cupslot Interface*/  
    }else if( DS.oneBitFlag[1] ==1){  
        /*Water Interface*/  
    }else if( DS.oneBitFlag[2] ==1){  
        /*Milk Interface*/  
    }else if( DS.oneBitFlag[3] ==1){  
        /*ice Interface*/  
    }else if( DS.oneBitFlag[4] ==1){  
        /*Steam Interface*/  
    }else if( DS.oneBitFlag[5] ==1){  
        /*Coffee Interface*/  
    }else if( DS.oneBitFlag[6] ==1){  
        /*Mocha Interface*/  
    }else if( DS.oneBitFlag[7] ==1){  
        /*Caramel Interface*/  
    }else if( DS.oneBitFlag[8] ==1){  
        /*Syrup Interface*/  
    }else if( DS.oneBitFlag[9] ==1){  
        /*FreshCream Interface*/  
    }  
}
```