

SD Part

Sweet heart

The ultimate coffee machine you've dreamed of

200412301 권용휘

200412359 최원석

200511337 양지승

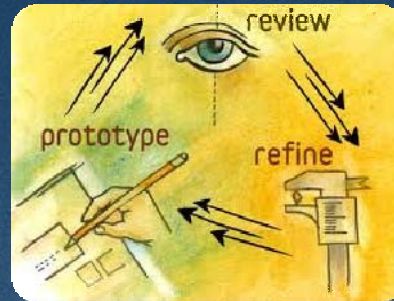
200611517 정훈섭

Agenda

- **What we've done during the SD process**
- **Remote Controller**
 - Transform Analysis
 - Structured Chart(basic)
 - Structured Chart(advance)
 - Code Generation
- **Sweet heart**
 - Transform Analysis
 - Structured Chart(basic)
 - Structured Chart(advance)
 - Code Generation
- **Web server**
 - Transform Analysis
 - Structured Chart(basic)
 - Structured Chart(advance)
 - Code Generation

What we've done

- Refined FSMs
- Extended some states in the existing FSMs

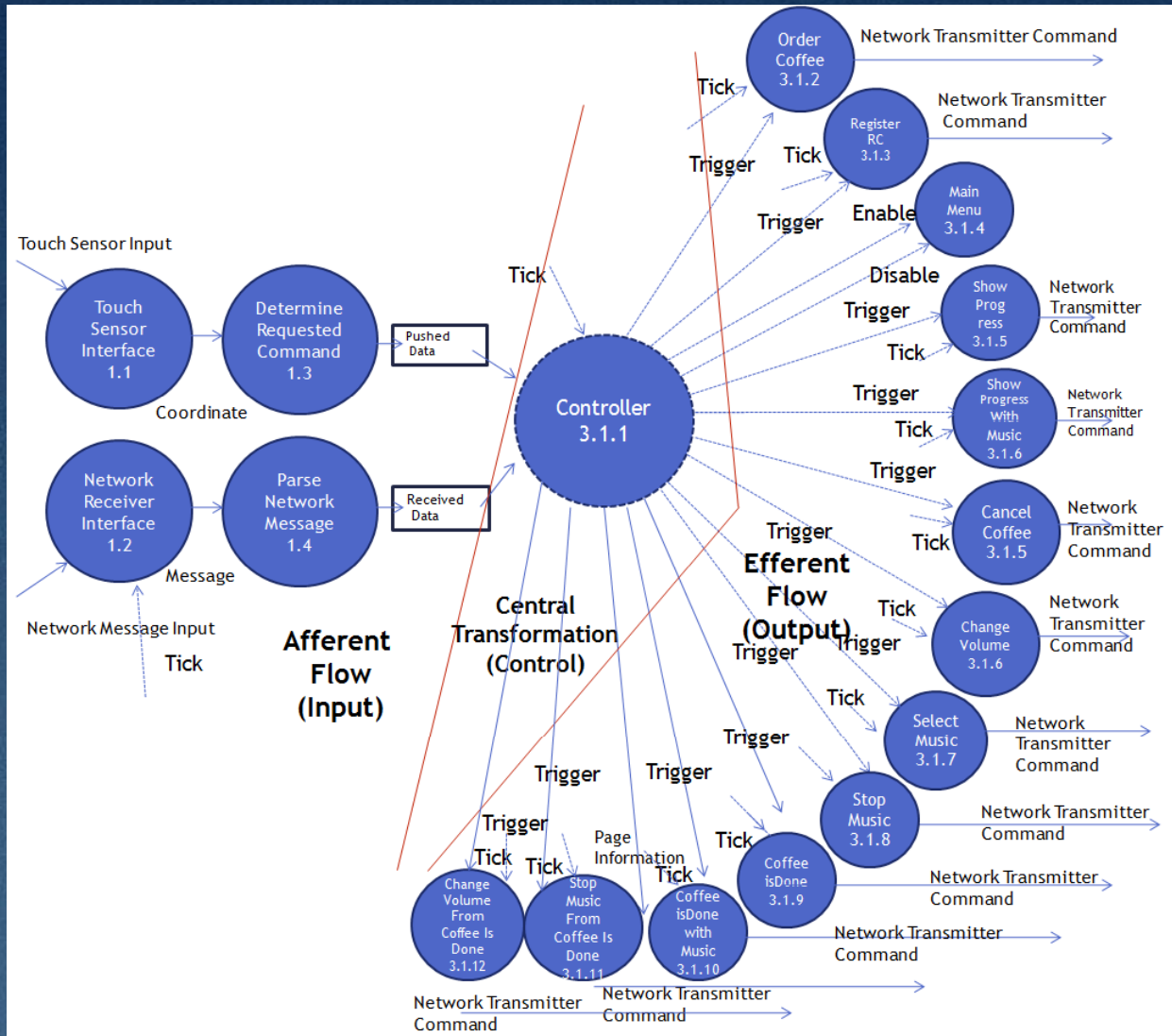


- Wrote three sample applications in C



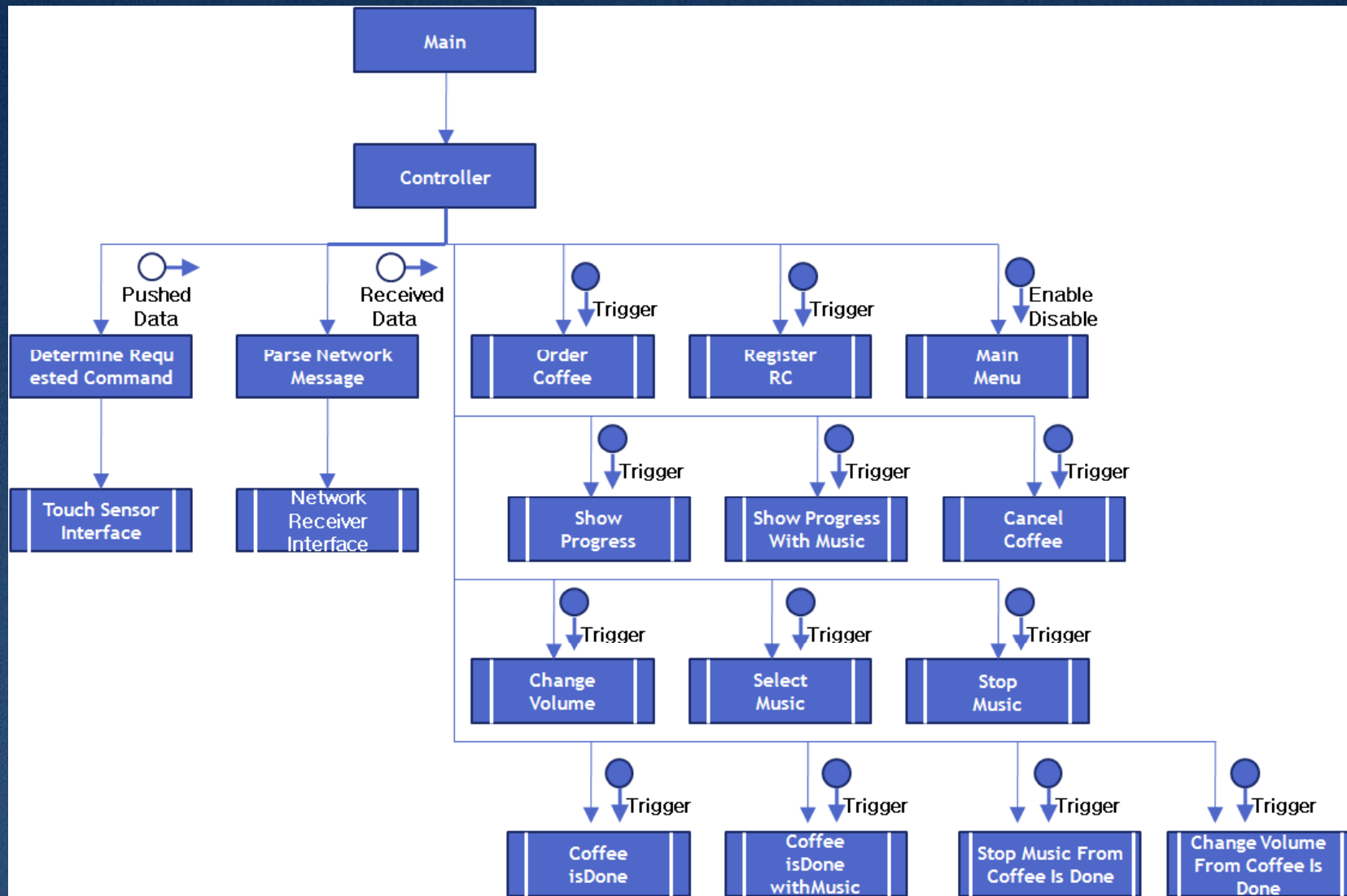
Transform Analysis

Remote Controller



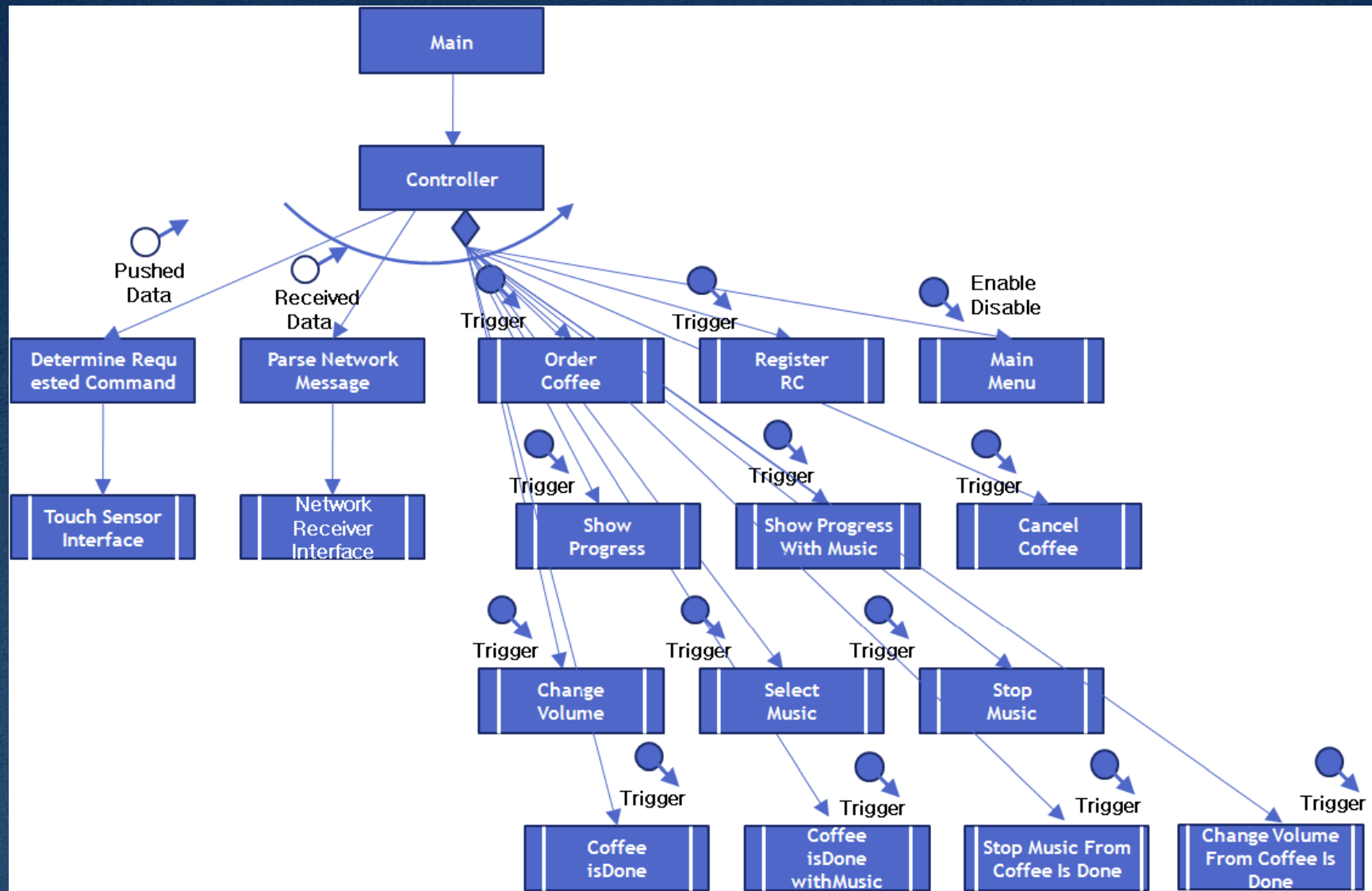
Structured Chart(Basic)

Remote Controller



Structured Chart(Advanced)

Remote Controller



Code Generation

Remote Controller

```
int mainLoop( )
{
    printf( "\n" );
    printf( "\n" );
    printf( "-----\n" );
    PrintCurrentState( );

    DetermineRequestedCommand( );
    ParseNetworkMessage( );

    switch( g_nCurrentState ) {
    case -1:
        MainMenu( );
        break;
    case ST_OrderCoffee: // Order Coffee
        OrderCoffee( );
        break;
    case ST_SelectMusic: // SelectMusic
        SelectMusic( );
        break;
    case ST_ShowProgress: // ShowProgress
        ShowProgress( );
        break;
    case ST_ShowProgressWithMusic: // ShowProgressWithMusic
        ShowProgressWithMusic( );
        break;
    case ST_ChangeVolume: // Change Volume
        ChangeVolume( );
        break;
    case ST_StopMusic: // Stop Music
        StopMusic( );
        break;
    }
```

Code Generation

Remote Controller

```
void ParseNetworkMessage()
{
    char input[100];
    int i;
    printf("ParseNetworkMessage : #n");

    for(i = 0; ; i++) {
        if( NAME_ParseNetworkMessage[i] == NULL ) {
            break;
        }
        printf(" [%d] %s#n", i+1, NAME_ParseNetworkMessage[i]);
    }
    printf("Select : ");
    scanf("%s", input);

    g_nRecvData = atoi( input );
}

void InputSensorManager()
{
    char input[100];
    int i;
    printf("InputSensorManager : #n");

    for(i = 0; ; i++) {
        if( NAME_InputSensorManager[i] == NULL ) {
            break;
        }
        printf(" [%d] %s#n", i+1, NAME_InputSensorManager[i]);
    }
    printf("Select : ");
    scanf("%s", input);

    g_nSensorData = atoi( input );
}
```


Code Generation

Remote Controller

```
void OrderCoffee( )
{
    printf("[!!!!] OrderCoffee\n");
    if( g_nPushedData == -1 ) {

    } else {
        switch( g_nPushedData ) {
            case ConfirmCoffeePushed: // ConfirmCoffeePushed
                g_nCurrentState = ST_SelectMusic;
                NetworkTransmitterCommand(2);
                PageInformation(ST_SelectMusic);
                printf("[!!!!] From OrderCoffee to SelectMusic\n");
                break;

        }
    }
}

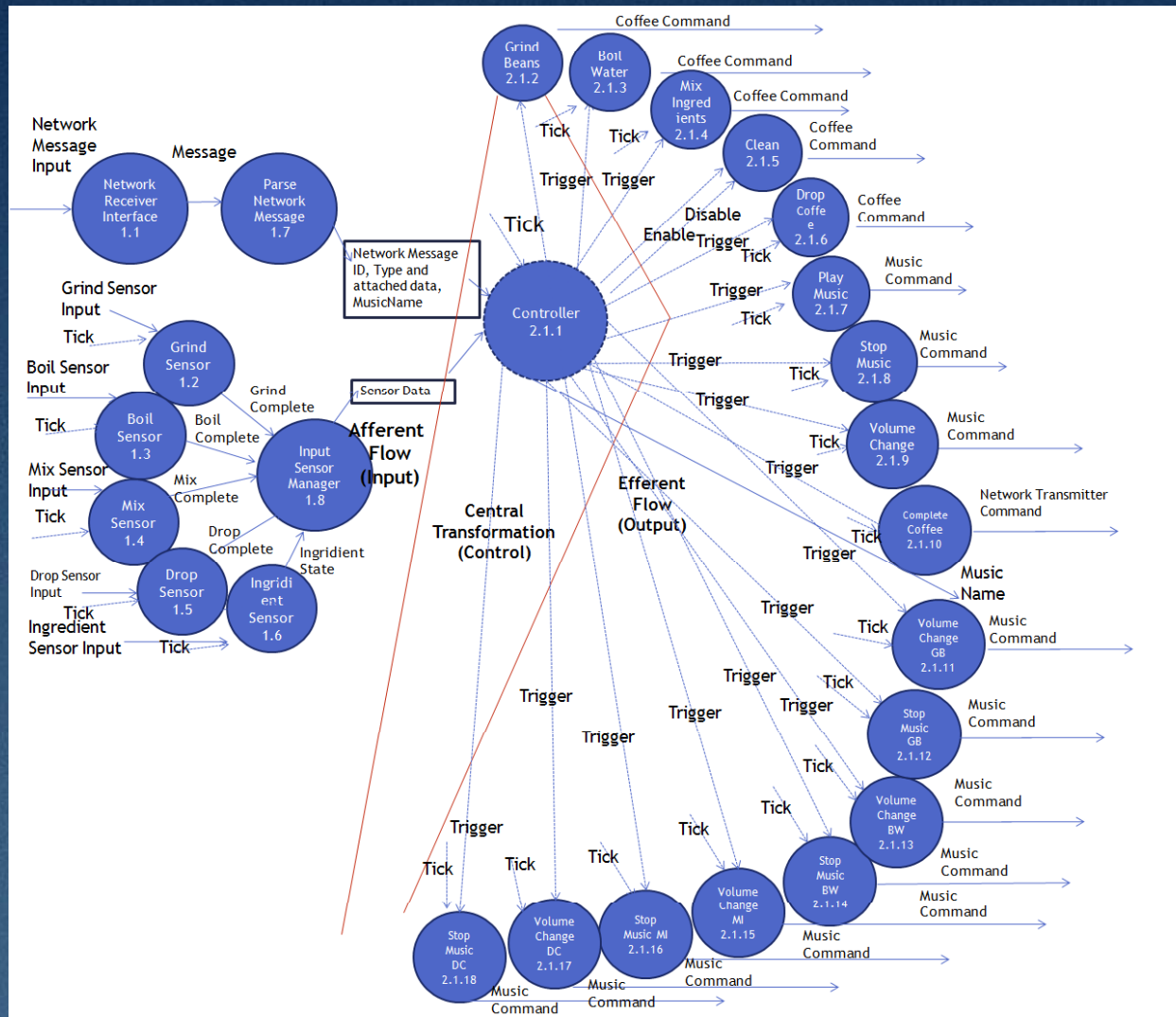
void MainMenu( )
{
    printf("[!!!!] MainMenu\n");
    switch( g_nPushedData ) {
        case 1: //OrderCoffeePushed
            g_nCurrentState = ST_OrderCoffee; // order coffee
            NetworkTransmitterCommand(1);
            PageInformation(ST_OrderCoffee);
            printf("[!!!!] From MainMenu to OrderCoffee\n");
            break;

        case RegisterRCPushed:
            g_nCurrentState = ST_RegisterRC;
            NetworkTransmitterCommand(8);
            PageInformation(ST_RegisterRC);
            printf("[!!!!] From MainMenu to ST_RegisterRC\n");
            break;

    }
}
```

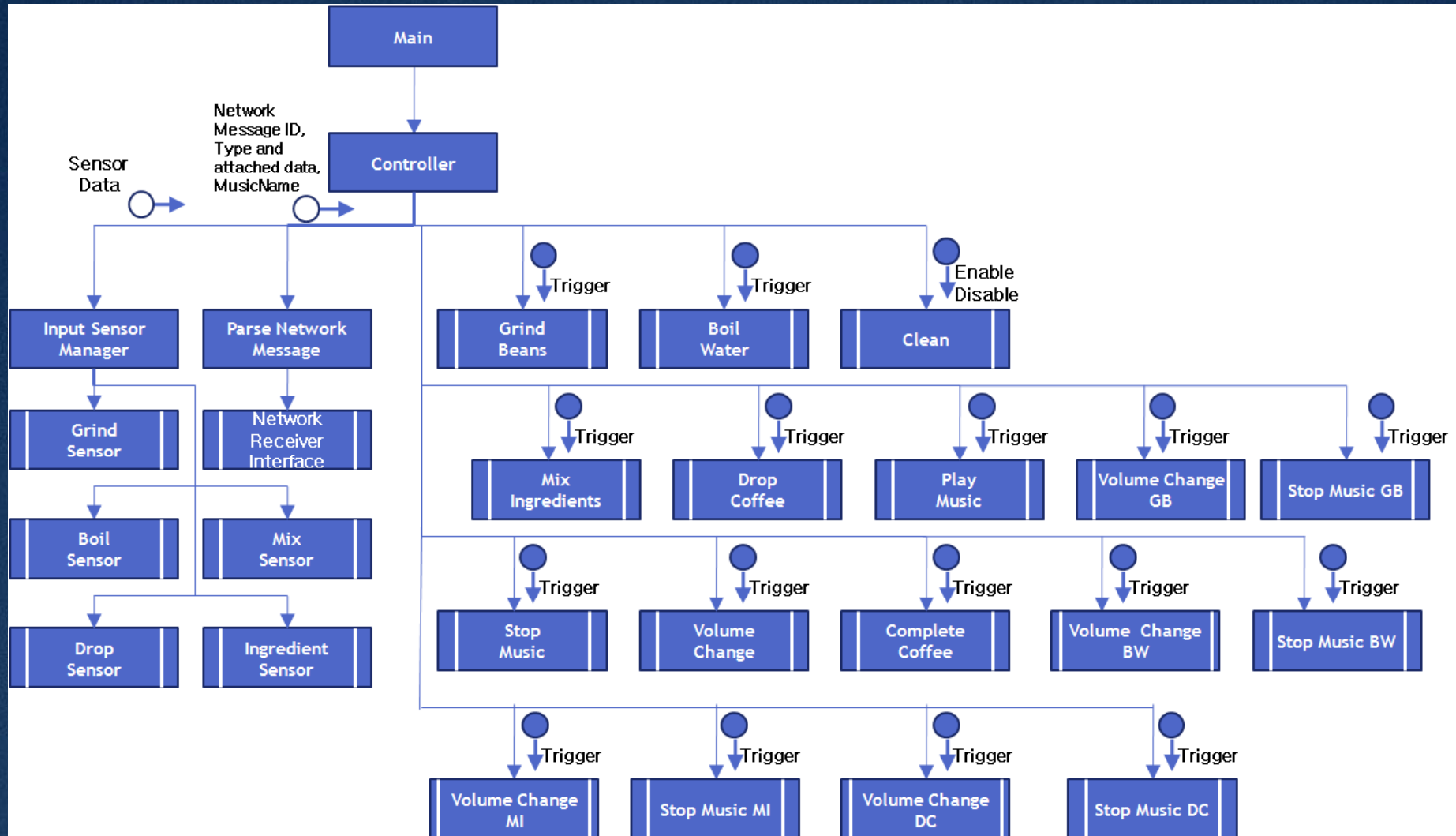
Transform Analysis

Sweet Heart



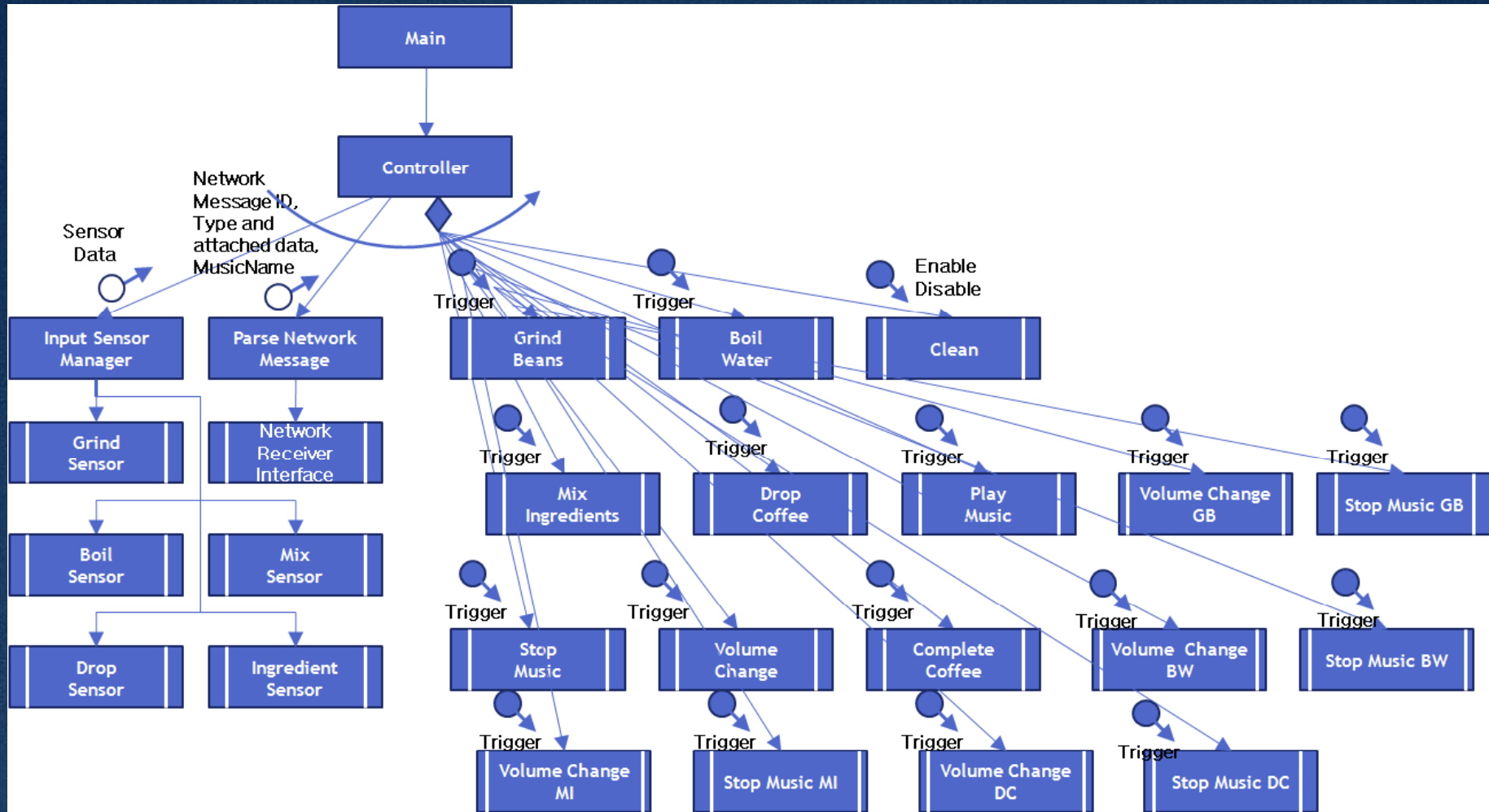
Structured Chart(Basic)

Sweet Heart



Structured Chart(Advanced)

Sweet Heart



Code Generation

Sweet heart

```
int mainLoop()
{
    printf("#n");
    printf("#n");
    printf("-----#n");
    PrintCurrentState();

    ParseNetworkMessage();
    InputSensorManager();

    switch( g_nCurrentState ) {
    case -1:
        Clean();
        break;
    case ST_GrindBeans:
        GrindBeans();
        break;
    case ST_BoilWater:
        BoilWater();
        break;
    case ST_MixIngredients:
        MixIngridients();
        break;
    case ST_DropCoffee:
        DropCoffee();
        break;
    case ST_CompleteCoffee:
        CompleteCoffee();
        break;
    case ST_VolumeChange:
        VolumeChange();
        break;
    }
```

Code Generation

Sweet heart

```
void ParseNetworkMessage( )
{
    char input[100];
    int i;
    printf("ParseNetworkMessage : \n");

    for(i = 0; ; i++ ) {
        if( NAME_ParseNetworkMessage[i] == NULL ) {
            break;
        }
        printf(" [%d] %s\n", i+1, NAME_ParseNetworkMessage[i]);
    }
    printf("Select : ");
    scanf("%s", input);

    g_nRecvData = atoi( input );
}

void InputSensorManager( )
{
    char input[100];
    int i;
    printf("InputSensorManager : \n");

    for(i = 0; ; i++ ) {
        if( NAME_InputSensorManager[i] == NULL ) {
            break;
        }
        printf(" [%d] %s\n", i+1, NAME_InputSensorManager[i]);
    }
    printf("Select : ");
    scanf("%s", input);

    g_nSensorData = atoi( input );
}
```

Code Generation

Sweet heart

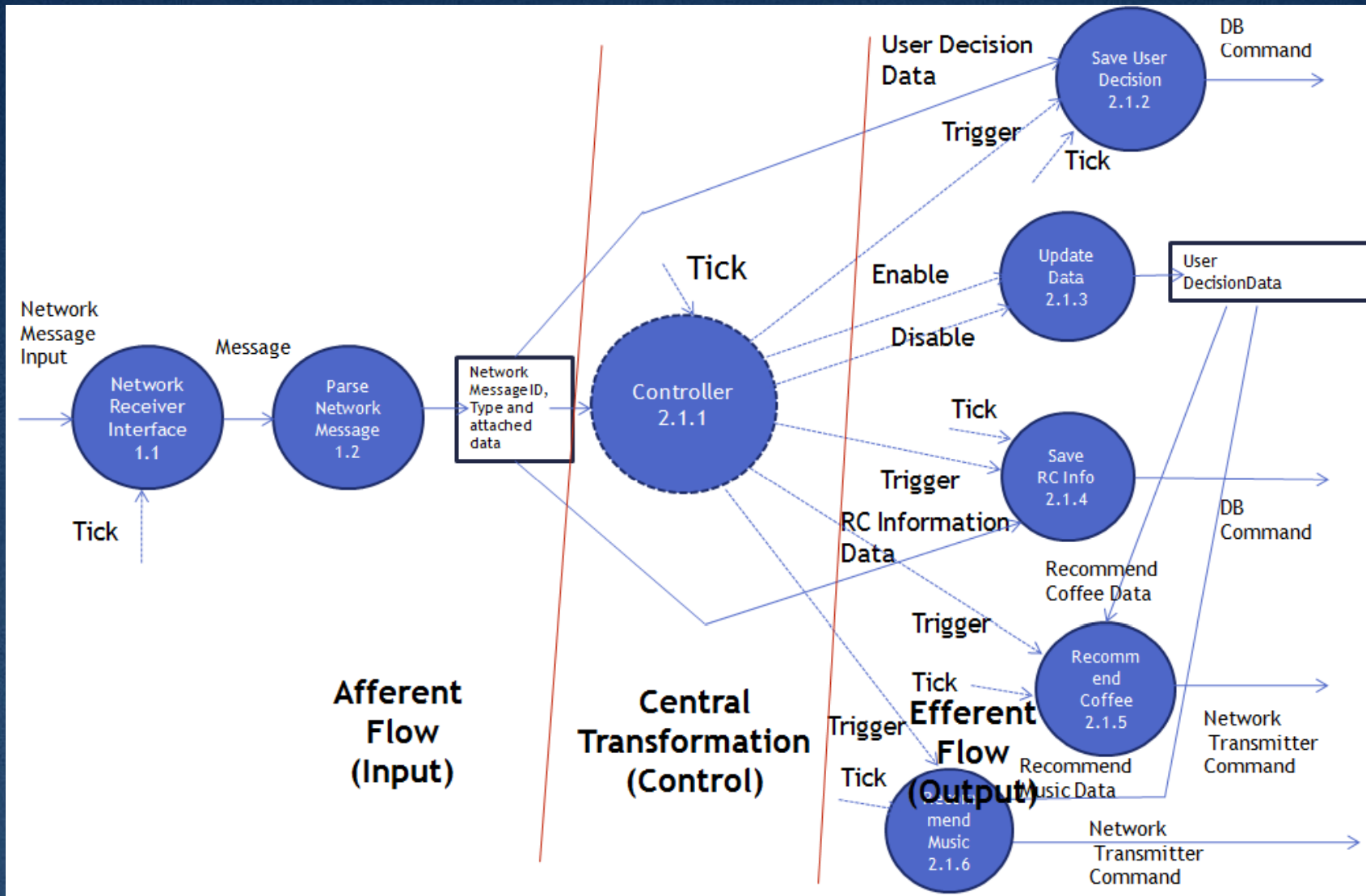
```
void GrindBeans()
{
    printf("[!!!] GrindBeans\n");

    MusicName(g_szMusicName);
    switch( g_nSensorData ) {
        case -1:
            break;
        case GrindComplete:
            g_nCurrentState = ST_BoilWater;
            CoffeeCommand(CC_BoilWater);
            printf("[!!!] From GrindBeans to ST_BoilWater\n");
            break;
    }

    switch( g_nRecvData ) {
        case -1:
            break;
        case VolumeChangeRecvd:
            g_nCurrentState = ST_VolumeChangeGB;
            printf("[!!!] From GrindBeans to ST_VolumeChangeGB\n");
            break;
        case StopMusicRecvd:
            g_nCurrentState = ST_StopMusicGB;
            printf("[!!!] From GrindBeans to ST_StopMusicGB\n");
            break;
        case CancelCoffeeRecvd:
            g_nCurrentState = -1;
            strcpy(g_szMusicName, "");
            printf("[!!!] From GridBeans to Clean\n");
            break;
    }
}
```

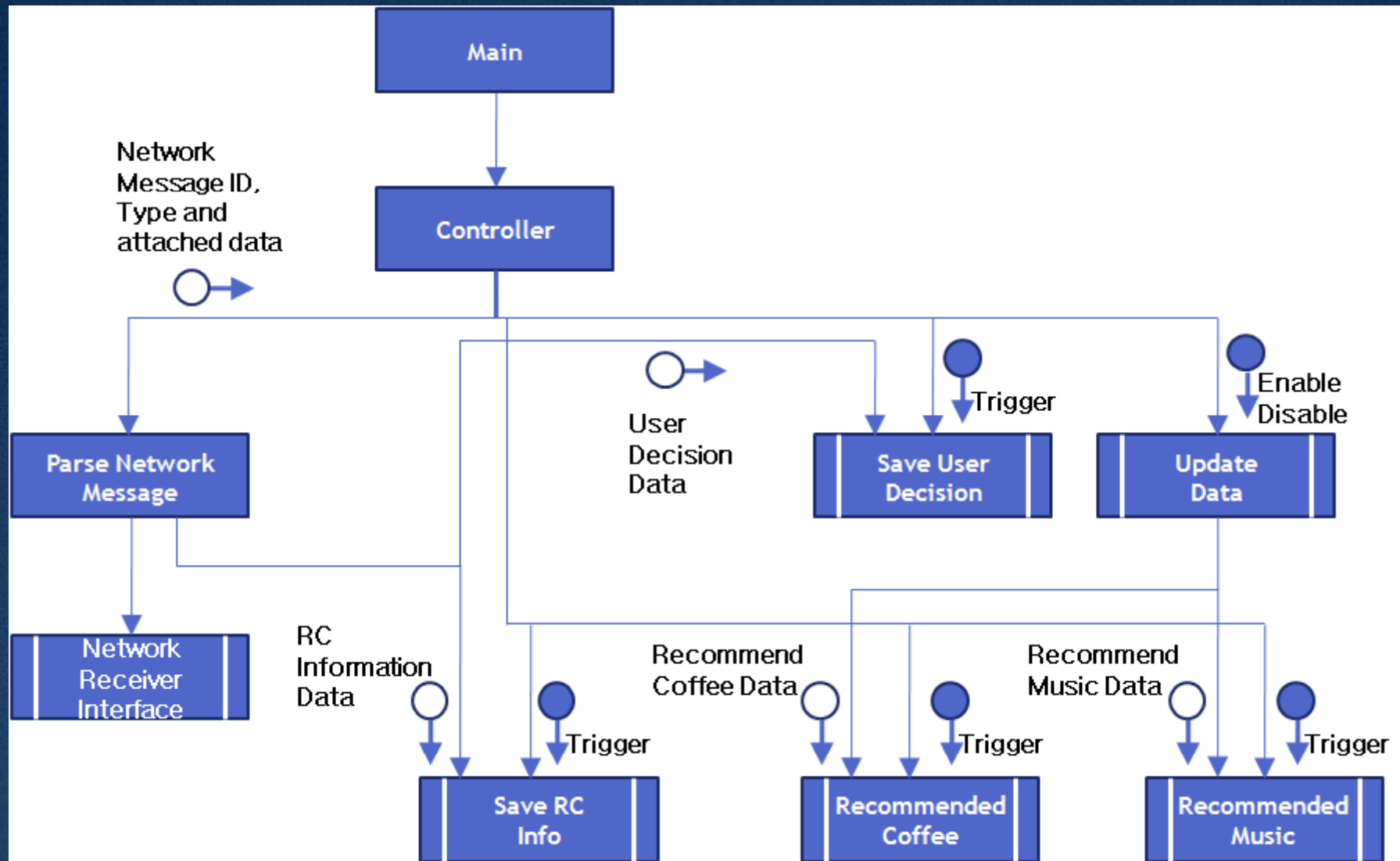
Transform Analysis

Web Server



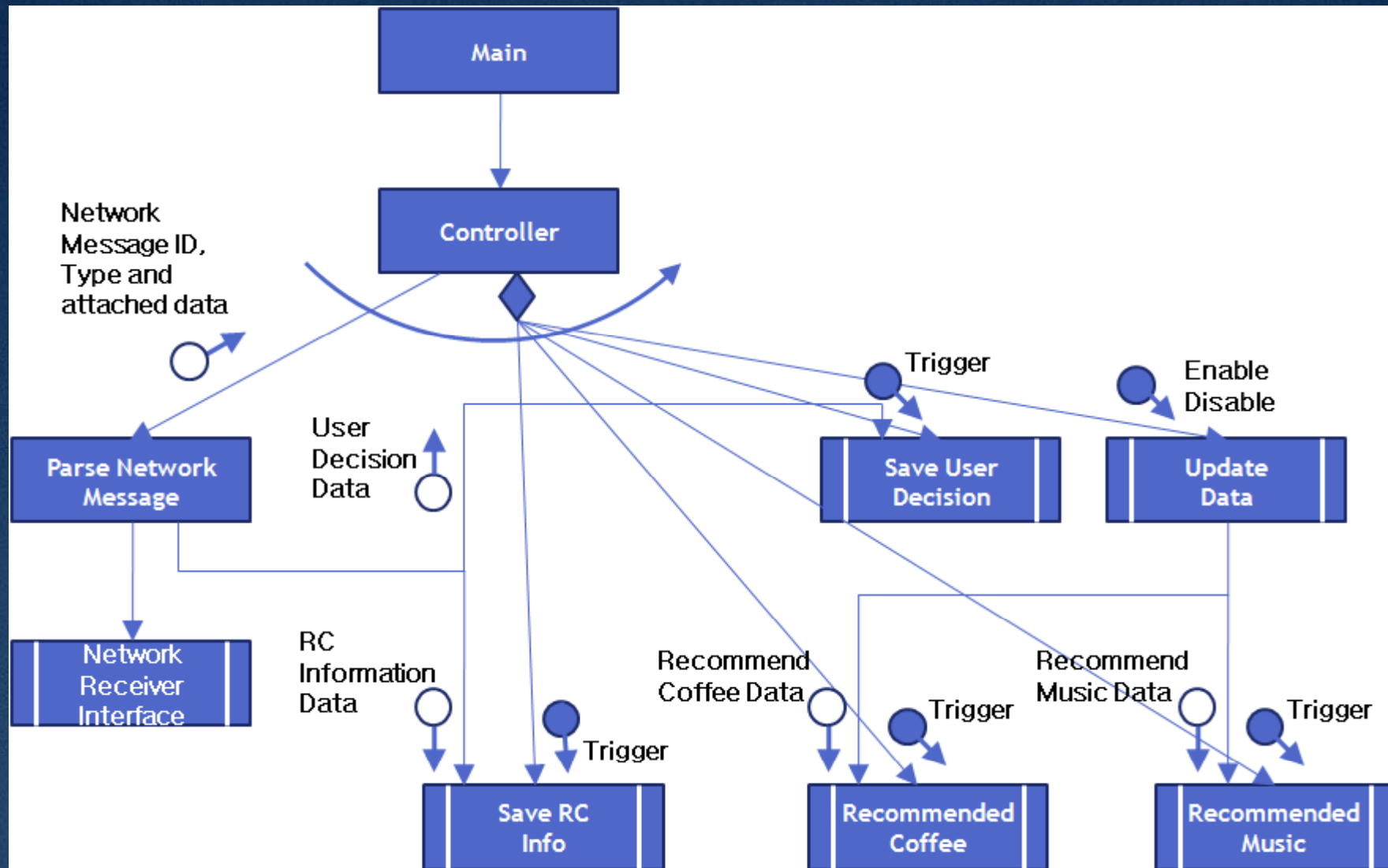
Structured Chart(Basic)

Web Server



Structured Chart(Advanced)

Web Server



Code Generation

Web server

```
int mainLoop()
{
    printf( "\n" );
    printf( "\n" );
    printf( "-----\n" );
    PrintCurrentState();

    ParseNetworkMessage();

    switch( g_nCurrentState ) {
    case -1:
        UpdateData();
        break;
    case ST_SaveUserDecision:
        SaveUserDecision();
        break;
    case ST_SaveRCInfo:
        SaveRCInfo();
        break;
    case ST_RecommendCoffee:
        RecommendCoffee();
        break;
    case ST_RecommendMusic:
        RecommendMusic();
        break;
    }
}
```

Code Generation

Web server

```
int PrintCurrentState()
{
    printf("[!!!] Current State : ");
    switch( g_nCurrentState ) {
        case -1:
            printf("Update Data");
            break;
        case ST_SaveUserDecision:
            printf("SaveUserDecision");
            break;
        case ST_SaveRCInfo:
            printf("SaveRCInfo");
            break;
        case ST_RecommendCoffee:
            printf("RecommendCoffee");
            break;
        case ST_RecommendMusic:
            printf("RecommendMusic");
            break;
    }
    printf("#n");
}
```

```
void ParseNetworkMessage()
{
    char input[100];
    int i;
    printf("ParseNetworkMessage : #n");

    for(i = 0; ; i++) {
        if( NAME_ParseNetworkMessage[i] == NULL ) {
            break;
        }
        printf(" [%d] %s#n", i+1, NAME_ParseNetworkMessage[i]);
    }
    printf("Select : ");
    scanf("%s", input);

    g_nRecvData = atoi( input );
}
```

Code Generation

Web server

```
void UpdateData()
{
    printf("[!!!] UpdateData#\n");

    switch( g_nRecvData ) {
        case -1:
            break;
        case ConfirmCoffeeRecvd:
            g_nCurrentState = ST_SaveUserDecision;
            DBCommand(1);
            printf("[!!!] From UpdateData to ST_SaveUserDecision#\n");
            break;
        case SelectMusicRecvd:
            g_nCurrentState = ST_RecommendMusic;
            NetworkTransmitterCommand(2);
            printf("[!!!] From UpdateData to ST_RecommendMusic#\n");
            break;
        case RegisterRCRecvd:
            g_nCurrentState = ST_SaveRCInfo;
            DBCommand(2);
            printf("[!!!] From UpdateData to ST_SaveRCInfo#\n");
            break;
        case OrderCoffeeRecvd:
            g_nCurrentState = ST_RecommendCoffee;
            NetworkTransmitterCommand(1);
            printf("[!!!] From UpdateData to ST_RecommendCoffee#\n");
            break;
    }
}
```

Code Generation (Demo)

Demonstration

Questions and Answers

Thank you