

PROJECT Public Transportation System

Final Presentation
TEAM 2

201111341 김성민
201111391 진청현
201311259 권오승
201311303 이정은

Contents

STR Analysis

- STR에서 발견된 문제점 분석

Modification

- 수정 사항 / 결과

Change of PTS

- PTS의 변화 과정

I. STR Analysis

- 특정 요금(1050~1250)에서 **지하철 IN이 되지 않아야 하나 IN이 되는 경우**
- 버스 환승 시 **30초 당 추가 요금을 받지 않음**
- 특정 요금(600~1050)에서 **환승 탑승 불가**
- 탑승 후 정산 시간이 지나고 다시 탑승하면 **미정산 요금을 받지 않음**
- **정산 계산식이 틀림**
- Card.txt에서 ‘ , ‘ 이 아닌 **띄어쓰기로 구분**

2. Modification - 수정 사항

특정 요금(1050~1250)에서 **지하철 IN이 되지 않아야 하나 IN이 되는 경우**

```
break;
case 3: //정산 후 IN
    if(strcmp(dpp[1].inout, "IN")==0){
        if(strcmp(dpp[1].means, "BUS")==0)
            a=1;
        else if(strcmp(dpp[1].means, "METRO")==0)
            a=2;
    }
    else{
        if(strcmp(dpp[1].means, "BUS")==0)
            a=0;
        else if(strcmp(dpp[1].means, "METRO")==0)
            a=3;
    }
    break;
}
```

**TransferCheck
수정**

2. Modification - 수정 사항

버스 환승 시 **30초 당 추가 요금을 받지 않음**

```
int TransferCheck(CARD *dpp, int usc){
    int a=0;

    switch(usc){
    case 0: //IN -> OUT
        if(strcmp(dpp[1].means, "METRO")==0) //모류
            a=2;
        else if(strcmp(dpp[2].means, "METRO")==0 &&
                strcmp(dpp[2].inout, "OUT")==0 &&
                (timeCalculate(dpp,1)<=120)) //환승후 시간 계산 추가요금
            a=1;
        else
            a=0;
    break;
}
```

TransferCheck 수정

2. Modification - 수정 사항

버스 환승 시 **30초 당 추가 요금을 받지 않음**

```
int FareCalculator(CARD *dpp, int ucr, int tcr){
    int a=0;

    switch(ucr){
        case 0:
            if(tcr==1){
                a = (timeCalculate(dpp,2)/240)*100;
                if(a>700)
                    a=700;
            }else if(tcr==0)
                a = 0;
            break;
```

FareCalculator 수정

2. Modification - 수정 사항

특정 요금(600~1050)에서 환승 탑승 불가

```
case 2 :  
    if(tcr==1)  
        a=700;  
    else  
        a=1050;  
    break;  
case 2 :  
    if(tcr==1)  
        a=600;  
    else  
        a=1050+200;  
    break;
```

**BalanceCheck
수정**

2. Modification - 수정 사항

특정 요금(600~1050)에서 환승 탑승 불가

```
case 2: //OUT -> IN
    if(strcmp(dpp[1].means, "METRO")==0 && (timeCalculate(dpp,2)<=120)) //0원
        a=1;
    else
        a=0;|
    //기본요금
    break;
case 2: //OUT -> IN
    if(strcmp(dpp[1].means, "BUS")==0 &&(timeCalculate(dpp,2)<=120)) //0원
        a=1;
    else
        a=0; //기본요금
    break;|
```

TransferCheck 수정

2. Modification - 수정 사항

탑승 후 정산 시간이 지나고 다시 탑승하면 **미정산 요금을 받지 않음**

```
int UnsettledCheck(CARD *dpp){
    int a=0;

    if(strncmp(dpp[0].time,dpp[1].time,8)!=0)
        a=3;
    else if(strcmp(dpp[1].inout,"IN")==0){
        a=1;
    }
    else if(strcmp(dpp[1].inout,"OUT")==0){
        a=2;
    }else
        printf("카드정보가 잘못되었습니다.!\n");

    strcpy(dpp[0].inout,"IN");

    return a;
}
```

UnsettledCheck 수정

2. Modification - 수정 사항

탑승 후 정산 시간이 지나고 다시 탑승하면 **미정산 요금을 받지 않음**

```
case 3: //IN->정산->IN
    if(strcmp(dpp[1].inout, "IN")==0){
        if(strcmp(dpp[1].means, "BUS")==0)
            a=1;
        else if(strcmp(dpp[1].means, "METRO")==0)
            a=2;
    }
    else{
        if(strcmp(dpp[1].means, "BUS")==0)
            a=0;
        else if(strcmp(dpp[1].means, "METRO")==0)
            a=3;
    }
    break;
}
return a;
```

TransferCheck
수정

2. Modification - 수정 사항

탑승 후 정산 시간이 지나고 다시 탑승하면 **미정산 요금을 받지 않음**

```
case 3 :  
    if(tcr==1)  
        a=1050+700;  
    else if(tcr==2)  
        a=1050+600;  
    else if(tcr==3)|  
        a=1050;  
    else if(tcr==0)  
        a=1050;  
    break;
```

FareCalculator
수정

2. Modification - 수정 사항

정산 계산식이 틀림- 정산 금액에 마이너스가 나오는 경우 수정

```
else if(dpp[i].fare == 0 || (dpp[i].fare >= 100 && dpp[i].fare <= 700))
{
    if(i>=3 &&strcmp(dpp[i-1].means,"BUS") == 0 && strcmp(dpp[i-1].inout, "IN") == 0 && dpp[i-1].fare == 0)
    {
        total = (dpp[i-3].fare+dpp[i].fare);
        *busMoney += round(total*total/(1050+total));
        *metroMoney += round(total*1050/(1050+total)) - dpp[i-3].fare;
    }
}
else if(dpp[i].fare == 0 || dpp[i].fare == 300 || dpp[i].fare == 600)
{
    if(i>=3 && strcmp(dpp[i-1].means,"METRO") == 0 && strcmp(dpp[i-1].inout, "IN") == 0 && dpp[i-1].fare == 0)
    {
        total = (dpp[i-3].fare+dpp[i].fare);
        *busMoney += round(total*1050/(1050+total)) - dpp[i-3].fare;
        *metroMoney += round(total*total/(1050+total));
    }
}
else if(dpp[i].fare == 1250 || dpp[i].fare == 1650)
    *metroMoney += dpp[i].fare;
```

2. Modification - 수정 사항

정산 계산식이 틀림- 하차 전 정산시간이 지나는 경우 수정

```
if( dpp[0].fare == 1750 || dpp[0].fare == 1650 )
{
    s = 1;
    if( strcmp( dpp[0].means, "BUS" ) == 0 )
        *busMoney += 1050;
    else if( strcmp( dpp[0].means, "METRO" ) == 0 )
        *metroMoney += 1050;
    *wholeMoney -= ( dpp[0].fare - 1050 );
    dpp[0].fare = 1050;
}
```

2. Modification - 수정 사항

정산 계산식이 틀림- 하차 전 정산시간이 지나는 경우 수정

```
if(strcmp(dpp[count-1].inout, "IN") == 0)
{
    if(strcmp(dpp[count-1].means, "METRO") == 0 && dpp[count-1].fare == 1050)
    {
        *metroMoney += 200;
        *wholeMoney += 200;
    }
    else if(strcmp(dpp[count-1].means, "METRO") == 0 && dpp[count-1].fare == 0)
    {
        *busMoney += 382;
        *metroMoney += 218;
        *wholeMoney += 600;
    }
    else if(strcmp(dpp[count-1].means, "BUS") == 0 && dpp[count-1].fare == 1050)
    {
        *busMoney += 700;
        *wholeMoney += 700;
    }
    else if(strcmp(dpp[count-1].means, "BUS") == 0 && dpp[count-1].fare == 0)
    {
        *busMoney += 280;
        *metroMoney += 420;
        *wholeMoney += 700;
    }
}
```

2. Modification - 수정 사항

Card.txt에서 ‘,’ 이 아닌 **띄어쓰기로 구분**

```
if ((fp2 = fopen("../../Card.txt", "r")) == NULL)
    printf("파일 열기 실패!");
else{
    for(i=1; i<3; i++){
        if(((i==1)||i==2)) && (dpp[i].time=="")
            break;
        fgets(str, 255, fp2);
        p = (char*)strtok(str, " ");
        strcpy(dpp[i].time, p);
        p = (char*)strtok(NULL, " ");
        strcpy(dpp[i].means, p);
        p = (char*)strtok(NULL, " ");
        strcpy(dpp[i].inout, p);
        p = (char*)strtok(NULL, " ");
        dpp[i].money = atoi(p);
        p = (char*)strtok(NULL, " ");
        strcpy(temp, p);
        p = (char*)strtok(NULL, " ");

        my_strcpy(dpp[i].ID, temp, 1);
        c = change_num(strlen(temp)-2, c, temp);
        dpp[i].count = c;
    }
}
```

cardread
수정

2. Modification - 수정 사항

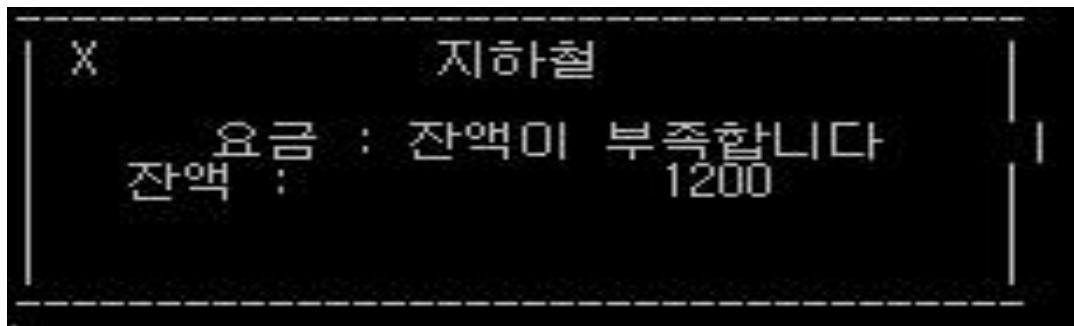
Card.txt에서 ‘,’ 이 아닌 **띄어쓰기로 구분**

```
if((fp3 = fopen("../..../Card.txt", "w")) == NULL)
    printf("파일 열기 실패!");
else{
    for(i=0; i<2; i++){
        fprintf(fp3, "%s,", dpp[i].time);
        fprintf(fp3, "%s,", dpp[i].means);
        fprintf(fp3, "%s,", dpp[i].inout);
        fprintf(fp3, "%d,", dpp[i].money);
        fprintf(fp3, "%s_", dpp[i].ID);
        fprintf(fp3, "%d", dpp[i].count);
        fprintf(fp3, "\r\n");
    }
}
fclose(fp3);
```

**CardDataUpdate
수정**

2. Modification - 결과

특정 요금(1050~1250)에서 **지하철 IN이 되지 않아야 하나 IN이 되는 경우**

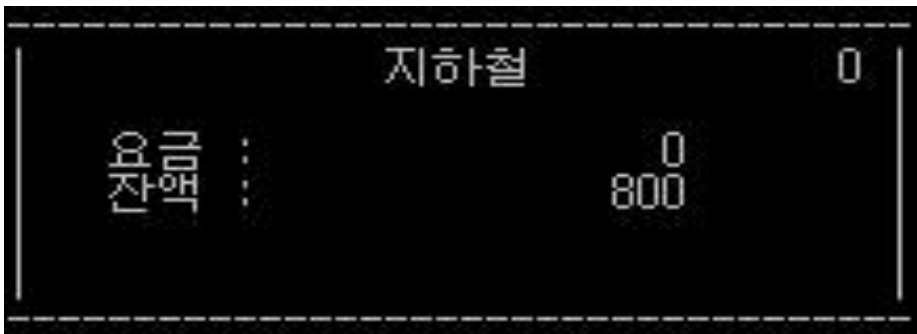
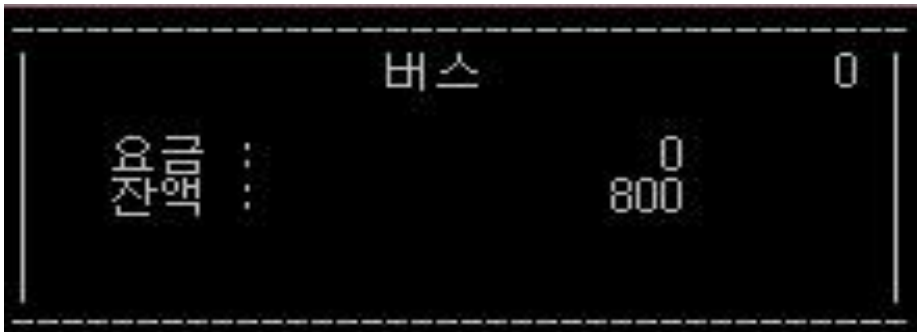


버스 환승 시 **30초 당 추가 요금을 받지 않음**



2. Modification - 결과

특정 요금(600~1050)에서 환승 탑승 불가



2. Modification - 결과

탑승 후 정산 시간이 지나고 다시 탑승하면 **미정산 요금을 받지 않음**
하차(환승이 됐는지 Check)

탑승

	지하철	0
요금 :	1750	
잔액 :	12200	

	지하철	0
요금 :	0	
잔액 :	12200	

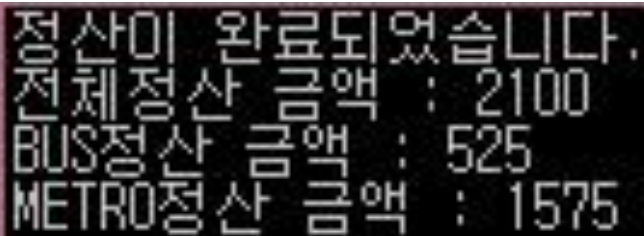
	버스	0
요금 :	1750	
잔액 :	12200	

	버스	0
요금 :	0	
잔액 :	12200	

2. Modification - 결과

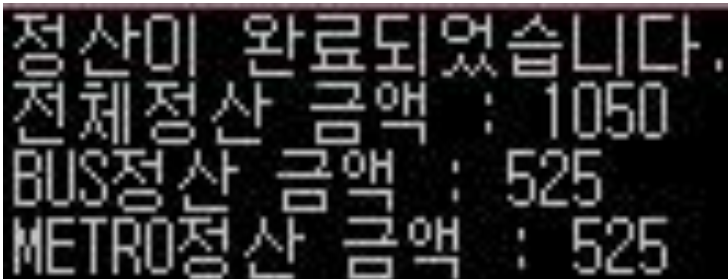
정산 계산식이 틀림

버스 IN / OUT / 지하철 환승 / OUT / 지하철 IN / OUT



정산이 완료되었습니다.
전체정산 금액 : 2100
BUS정산 금액 : 525
METRO정산 금액 : 1575

지하철 IN / OUT / 버스 환승 / OUT

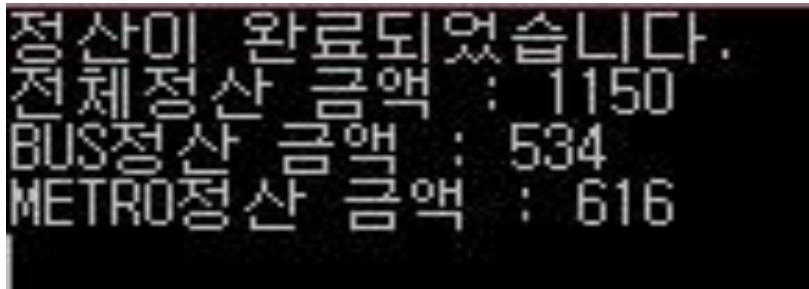


정산이 완료되었습니다.
전체정산 금액 : 1050
BUS정산 금액 : 525
METRO정산 금액 : 525

2. Modification - 결과

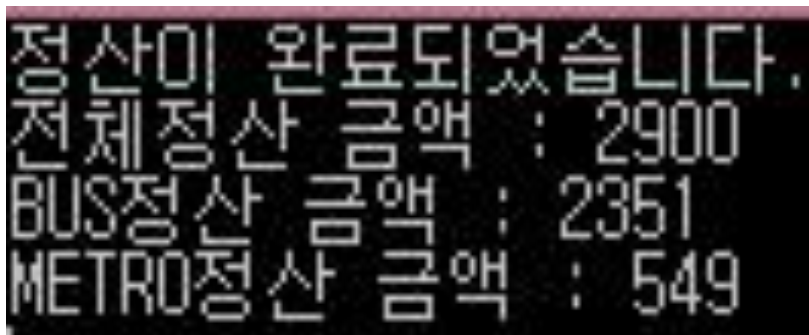
정산 계산식이 틀림

버스 IN / OUT / 지하철 환승 / OUT / 버스 환승 / 40초 후 OUT



정산이 완료되었습니다.
전체정산 금액 : 1150
BUS정산 금액 : 534
METRO정산 금액 : 616

지하철 IN / OUT / 버스 환승 / 40초 후 OUT / 버스 IN



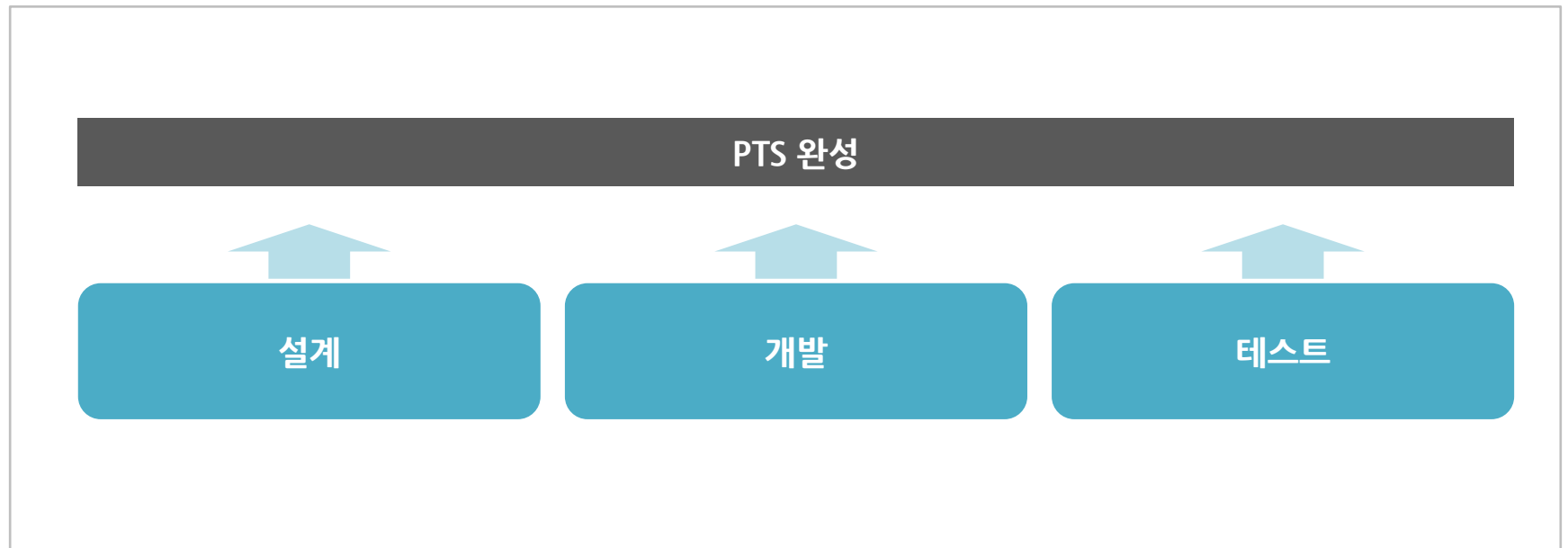
정산이 완료되었습니다.
전체정산 금액 : 2900
BUS정산 금액 : 2351
METRO정산 금액 : 549

2. Modification - 결과

Card.txt에서 ‘ , ‘ 이 아닌 띄어쓰기로 구분

```
201 407070448, ME TRO, OUT, 1200, 2_15  
201 407070416, ME TRO, IN, 9600, 1_122
```

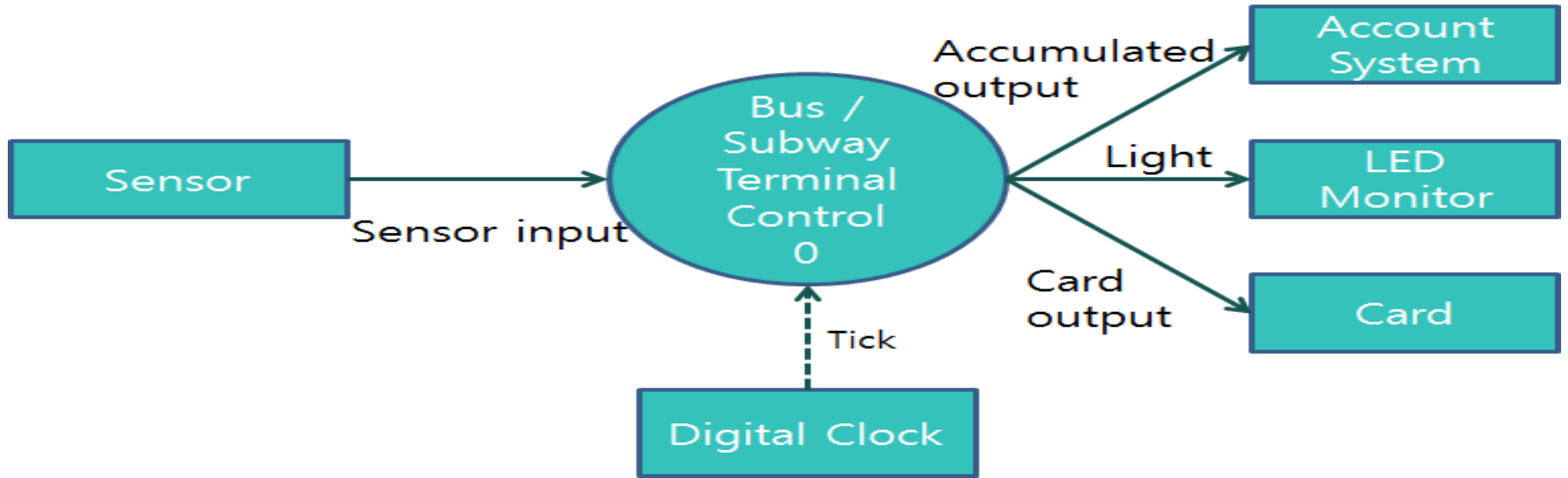
3. Change of PTS



SA

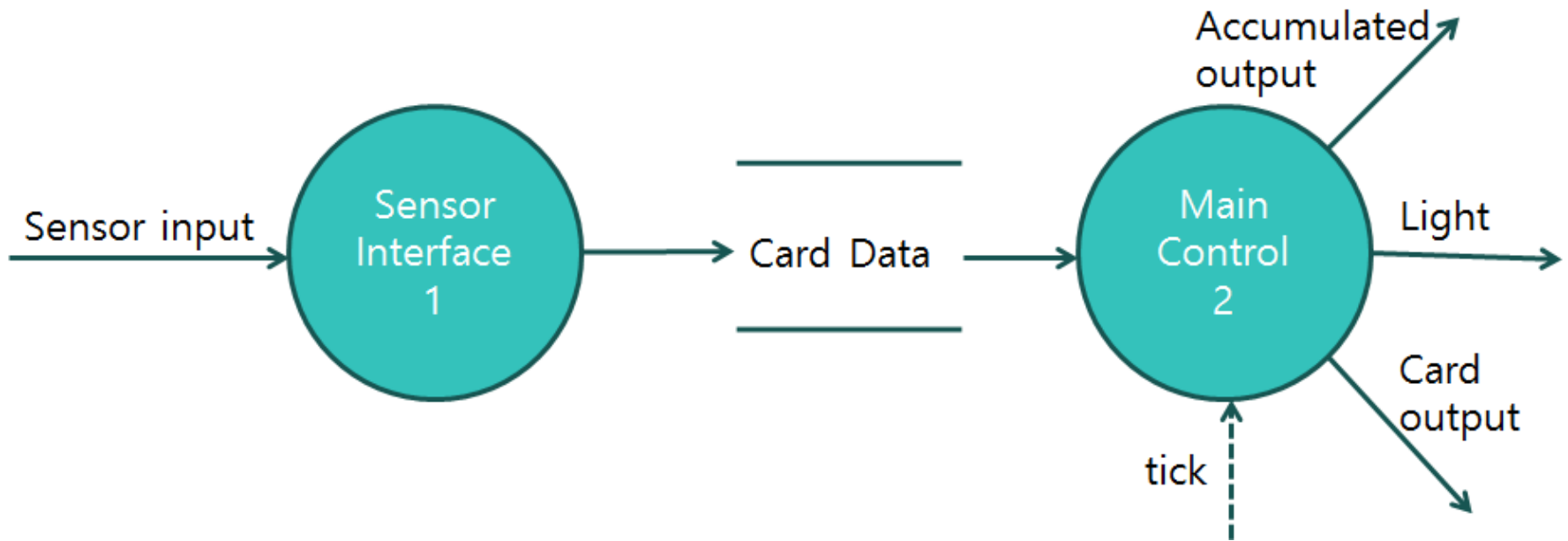
Terminal DFD

DFD level 0



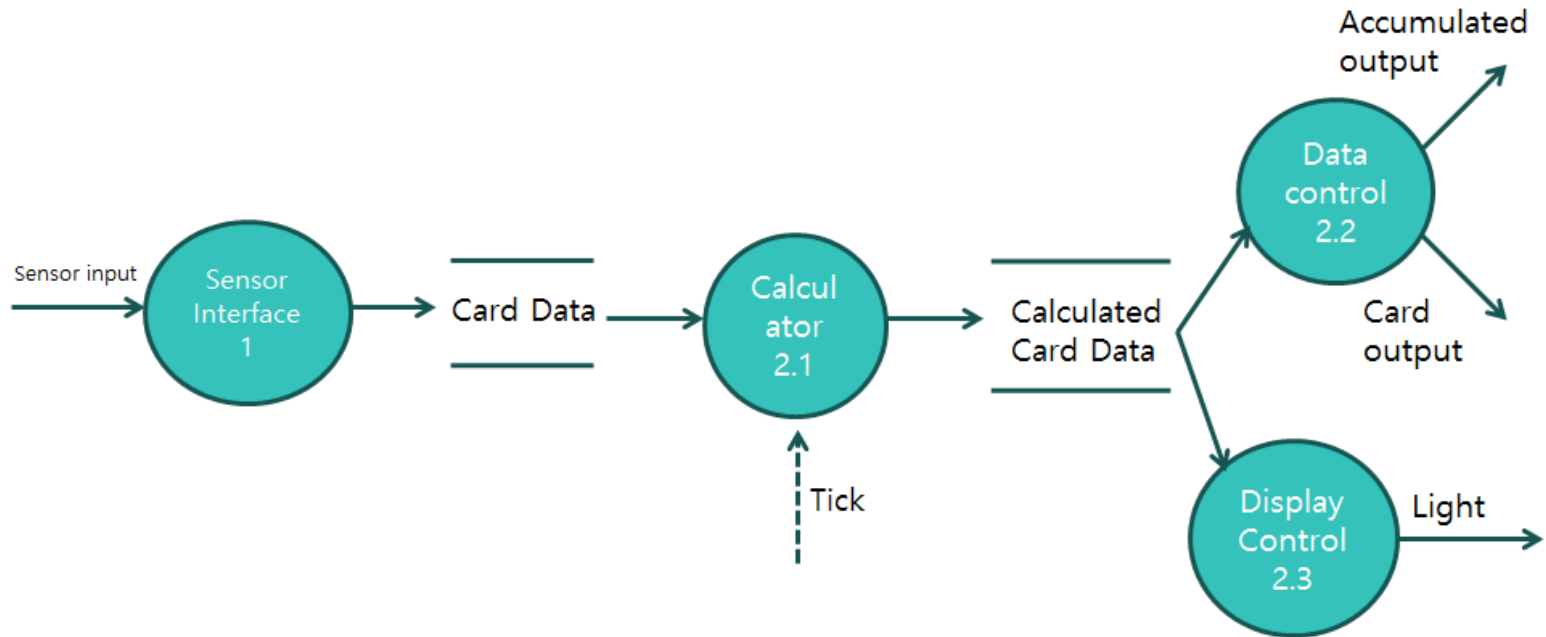
수정사항 X

DFD level 1



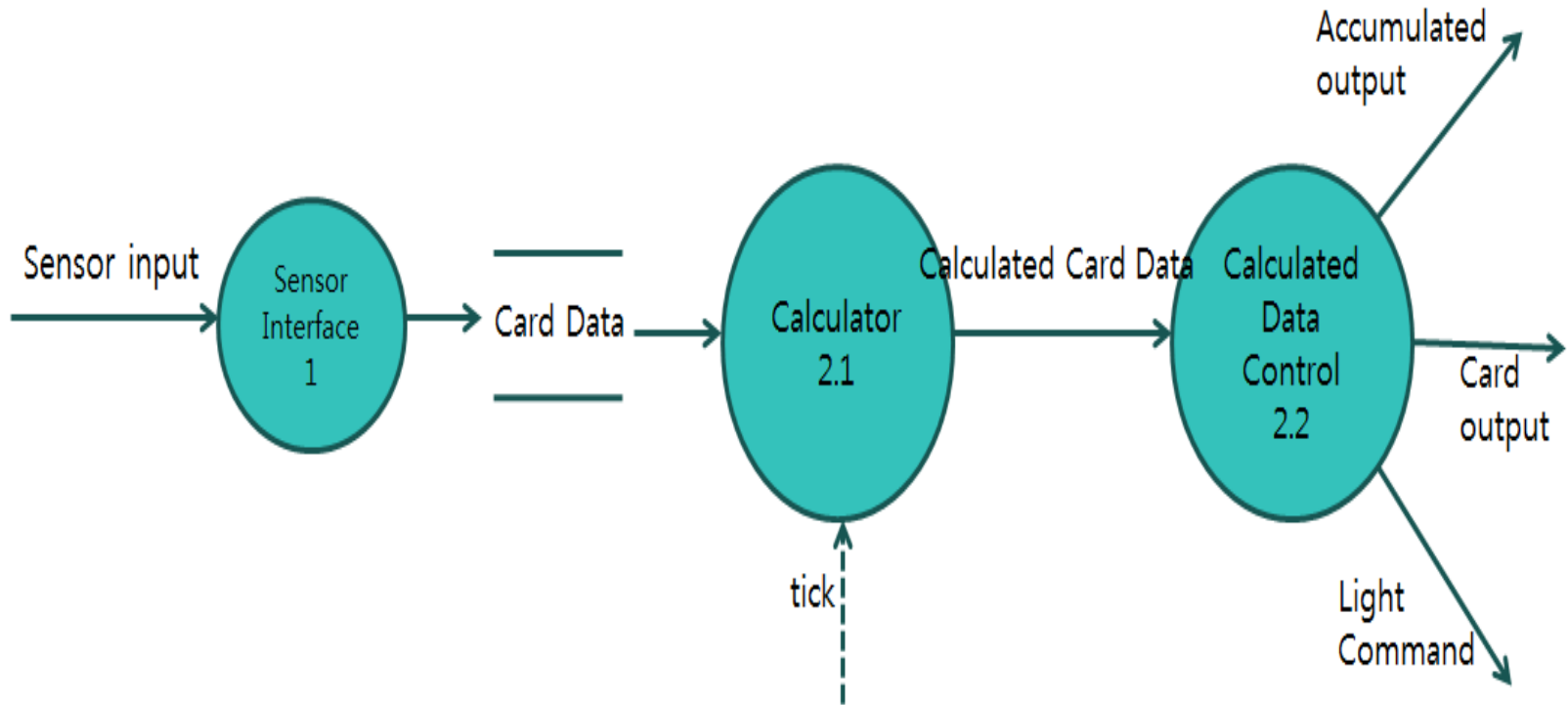
수정사항 X

DFD level 2



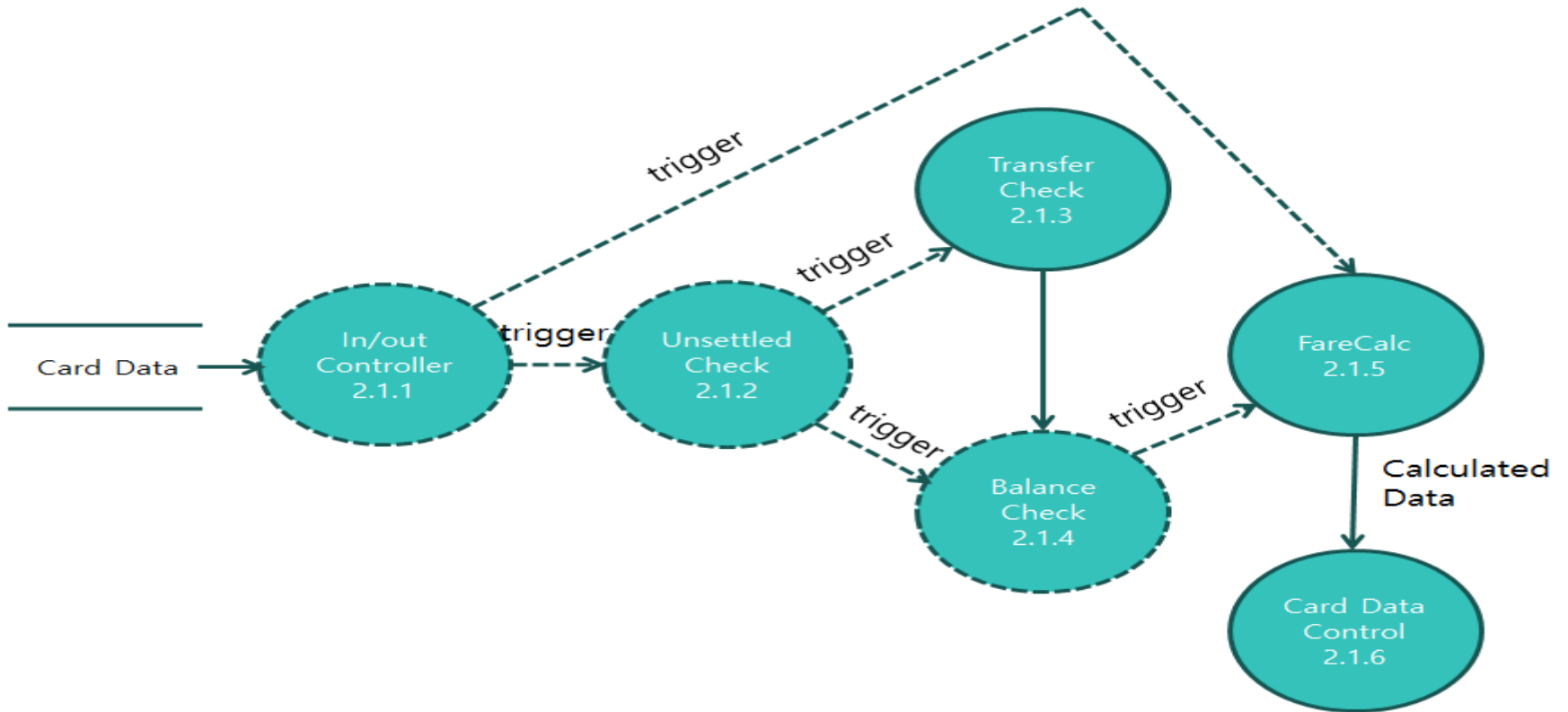
SRA ver1.0

DFD level 2



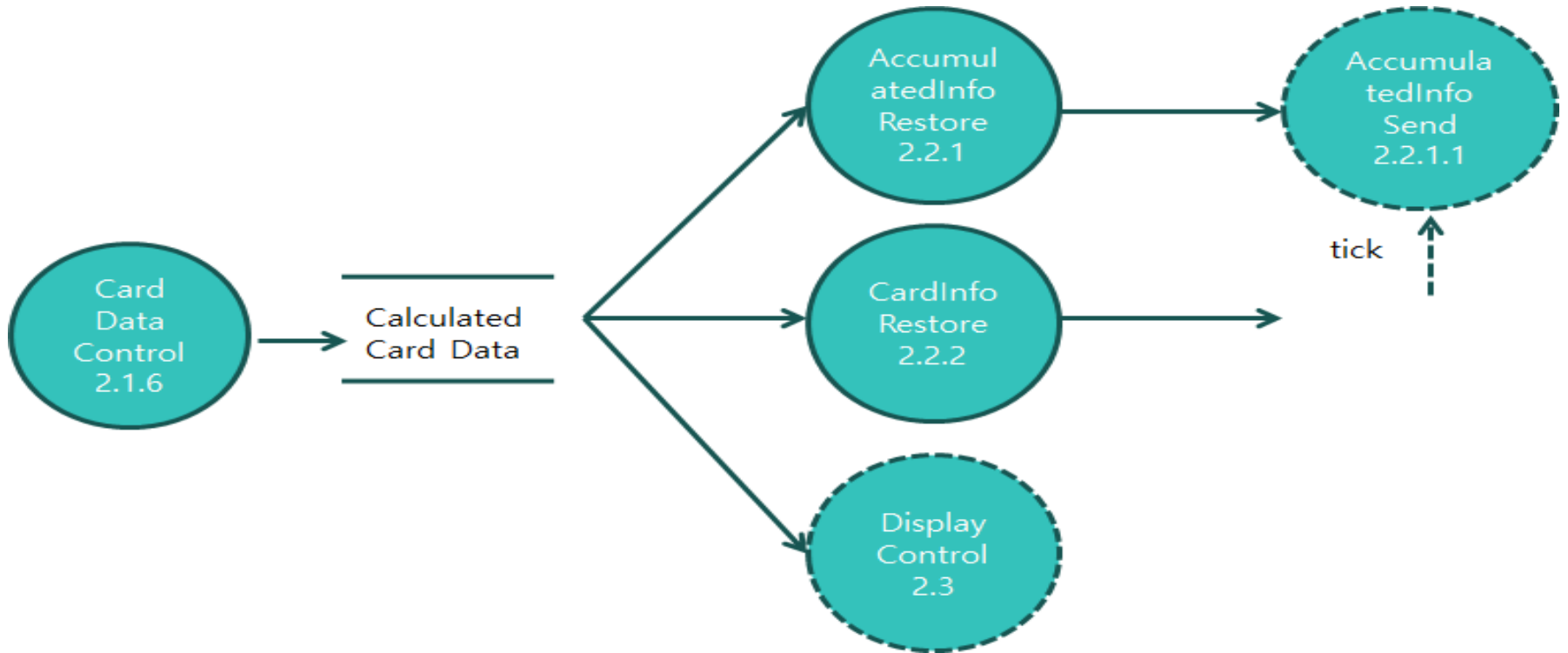
SRA ver4.0

DFD level 3



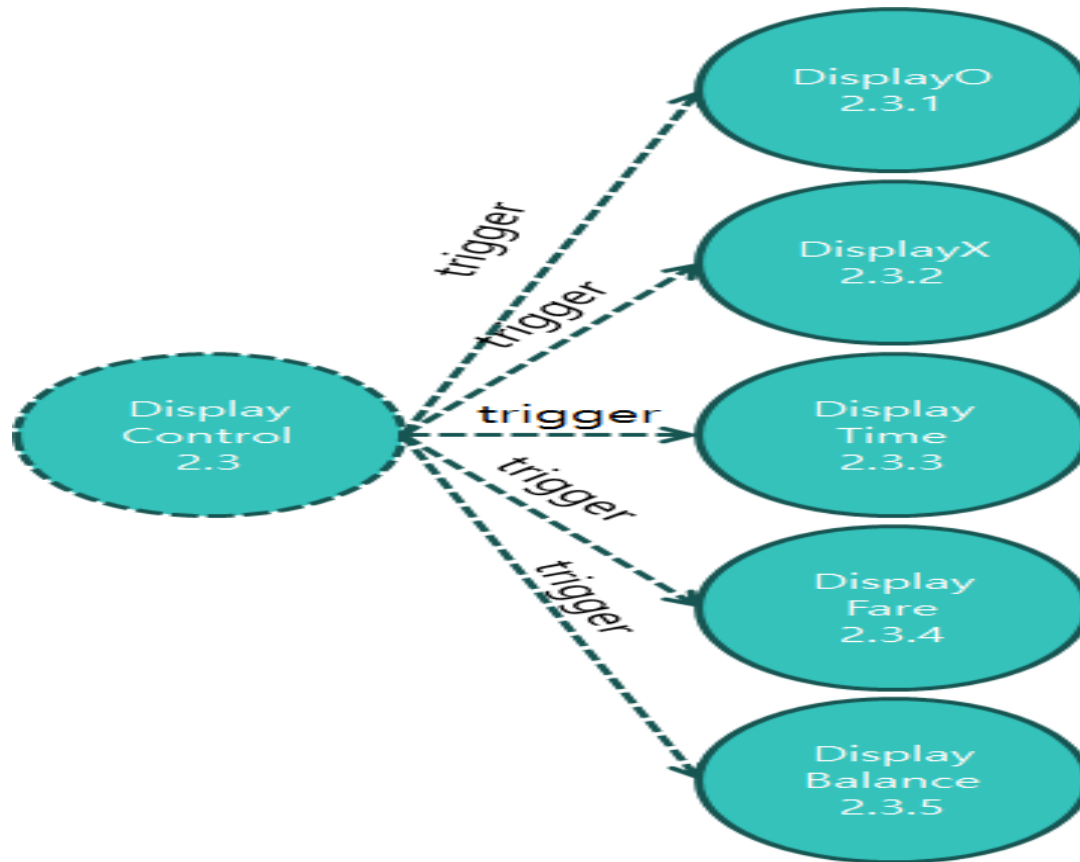
SRA ver1.0

DFD level 3



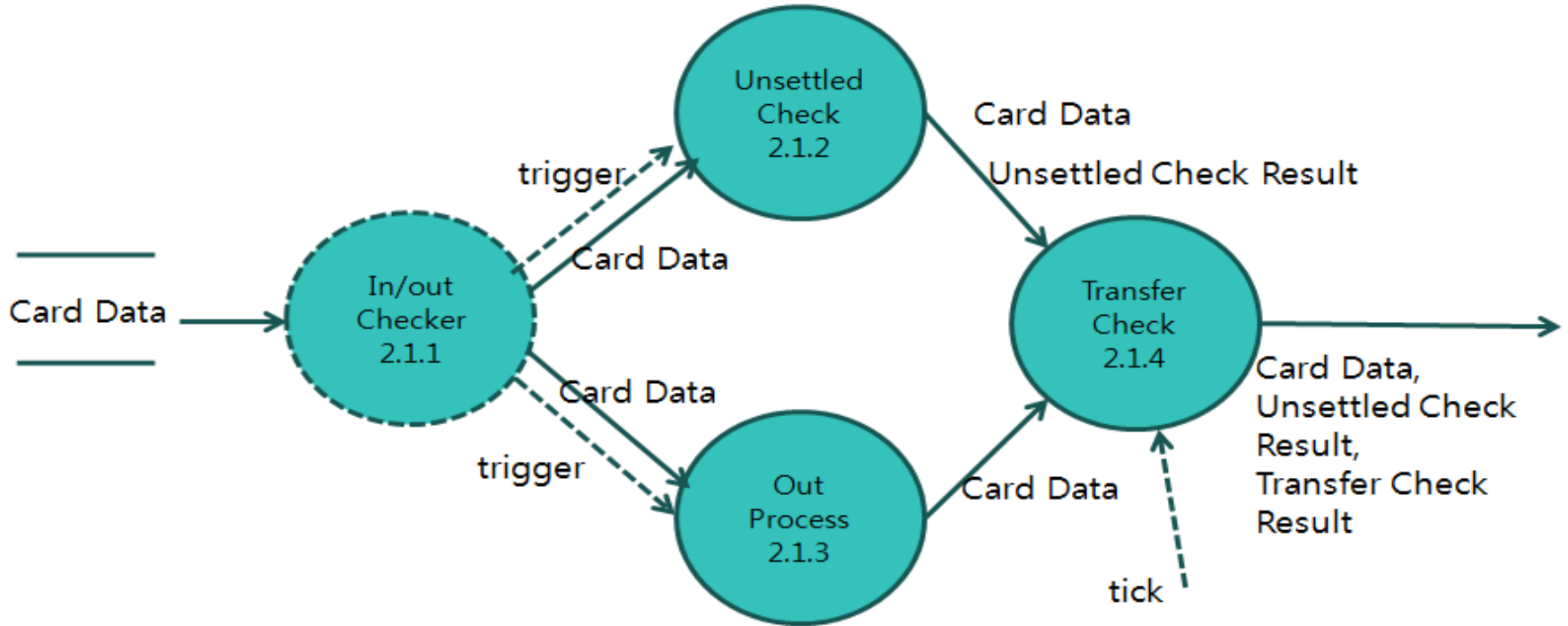
SRA ver1.0

DFD level 3



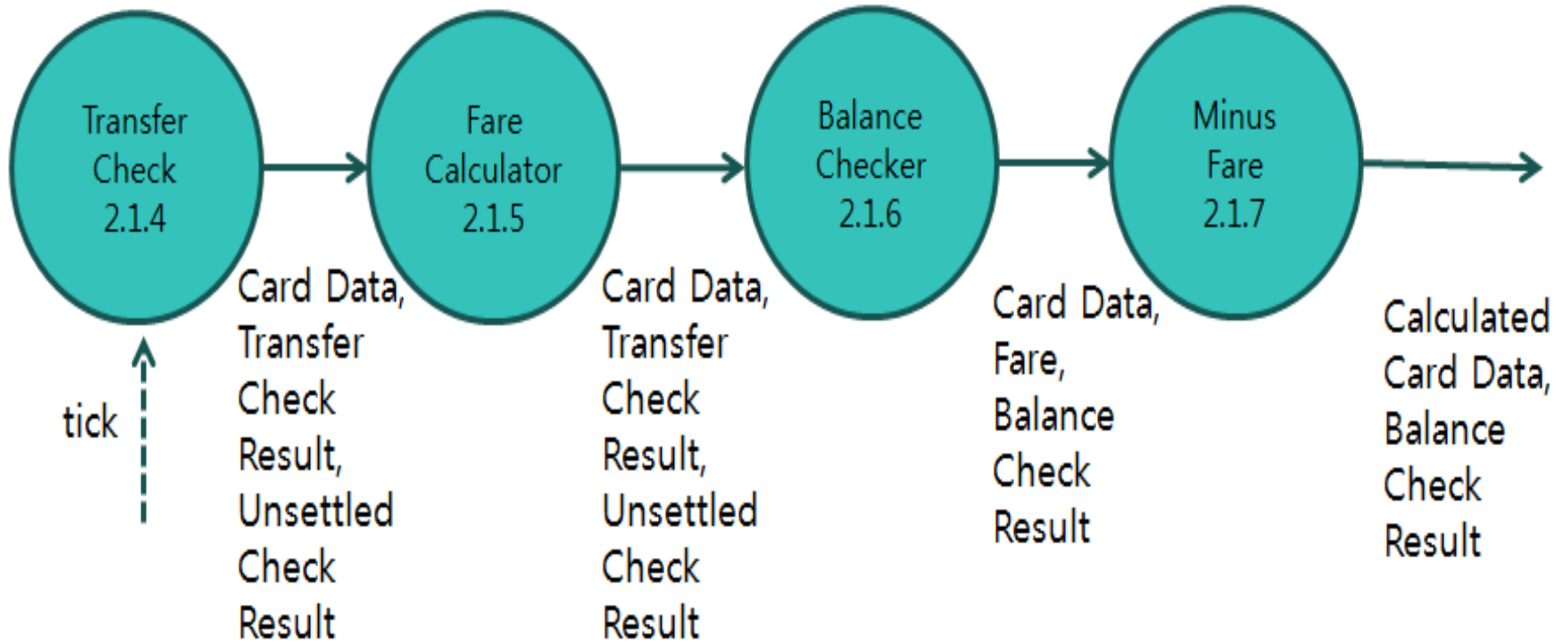
SRA ver1.0

DFD level 3



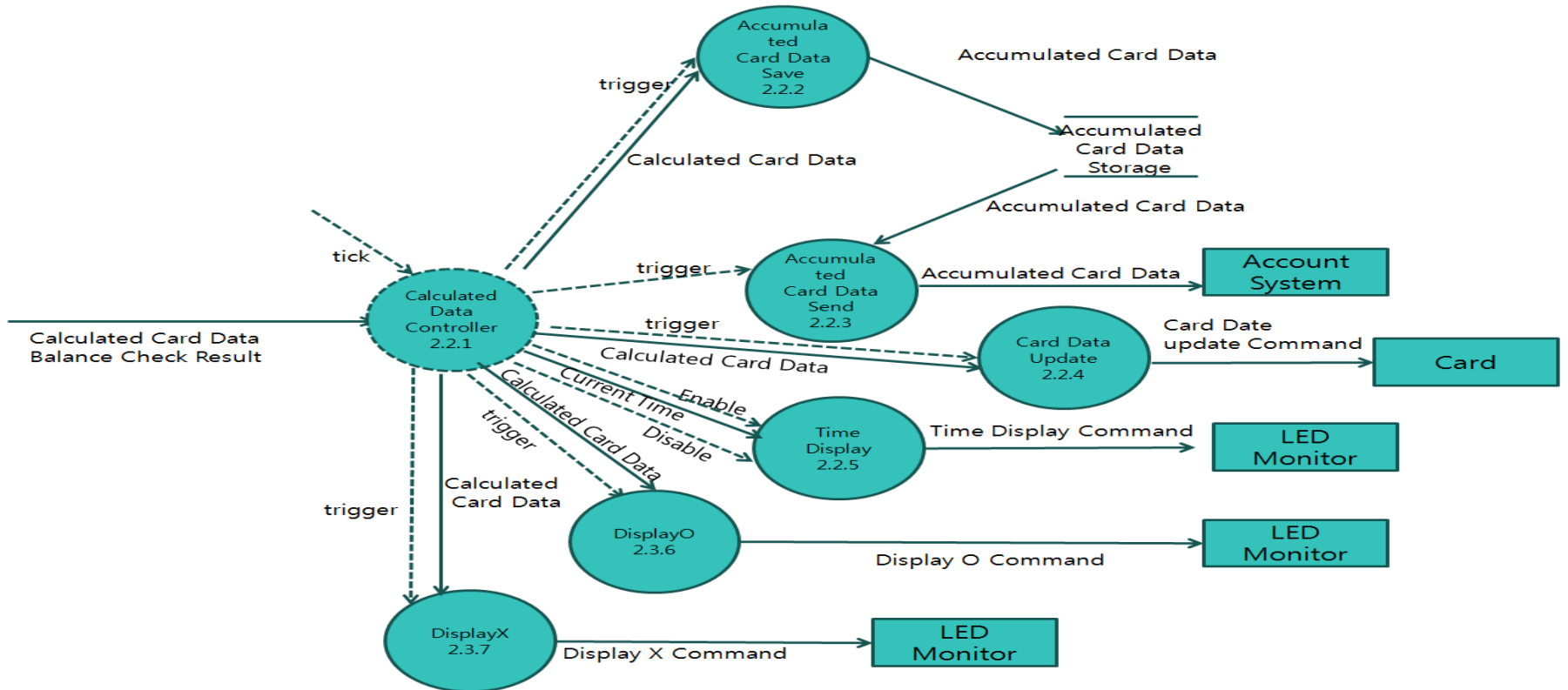
SRA ver4.0

DFD level 3



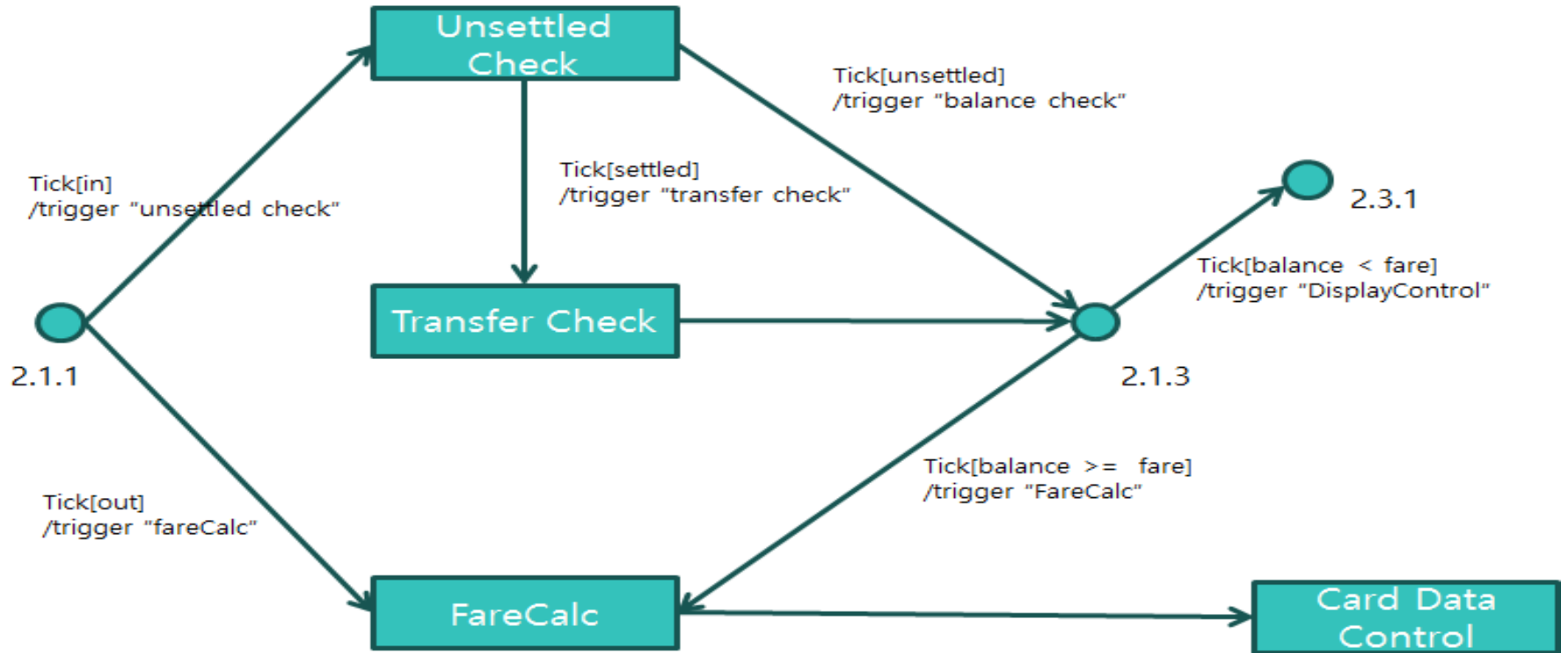
SRA ver4.0

DFD level 3



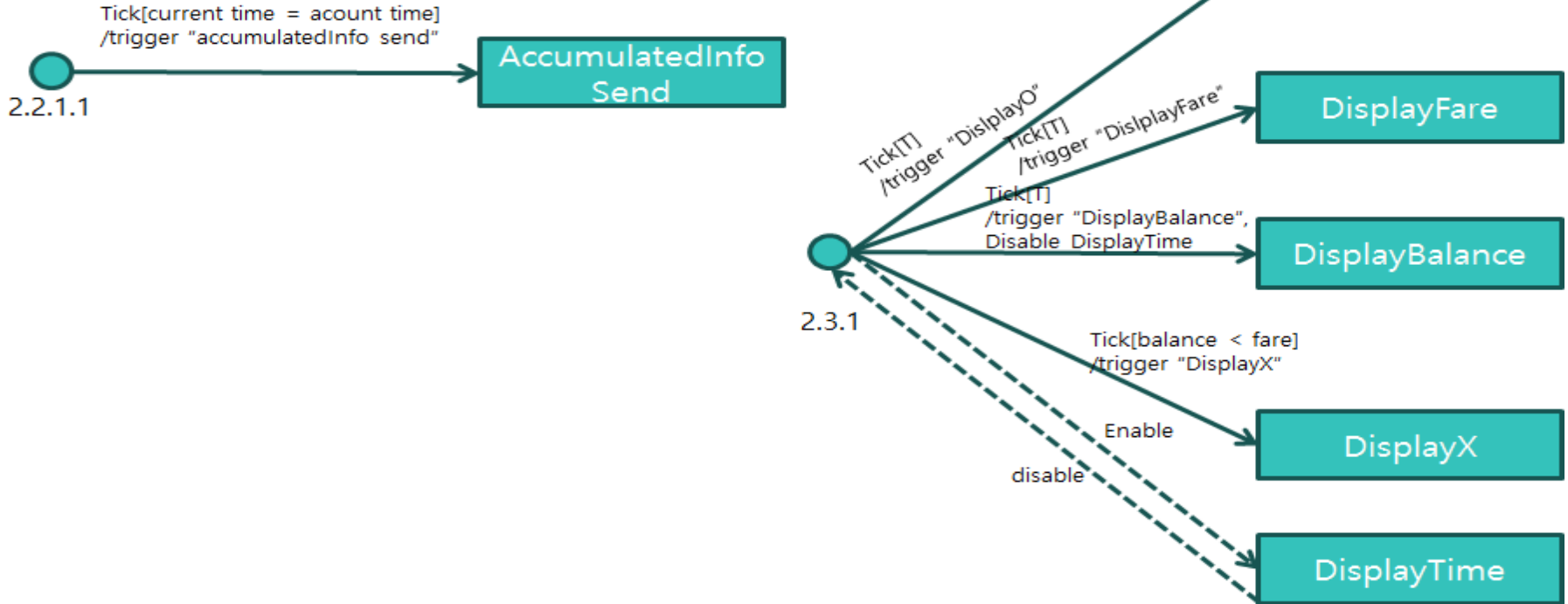
SRA ver4.0

DFD level 4



SRA ver1.0

DFD level 4



SRA ver1.0

DFD level 4

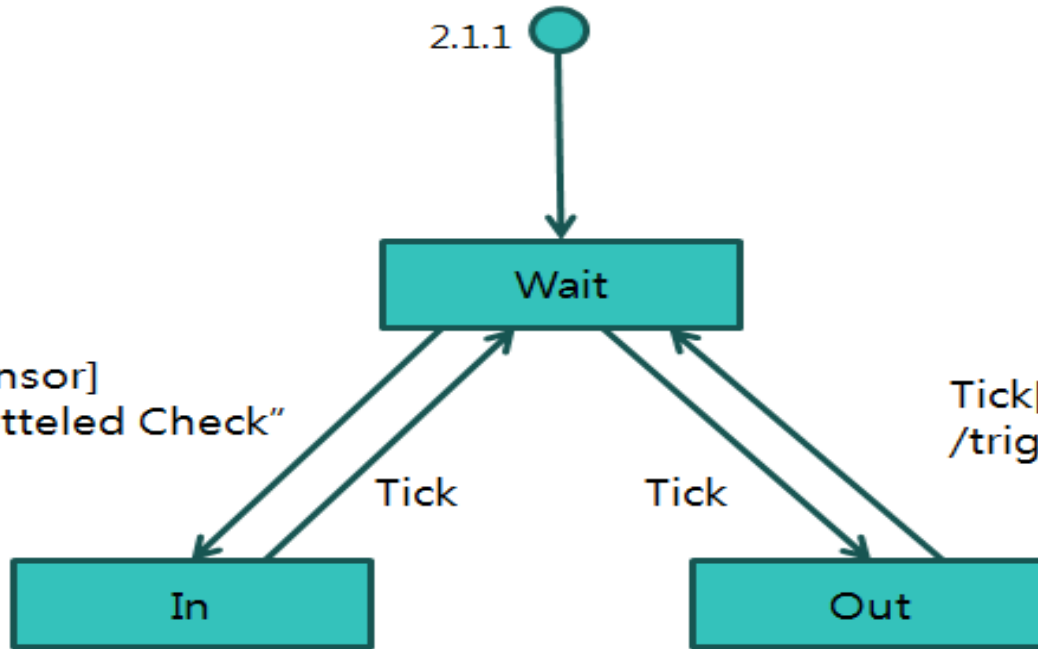


STD

2.1.1

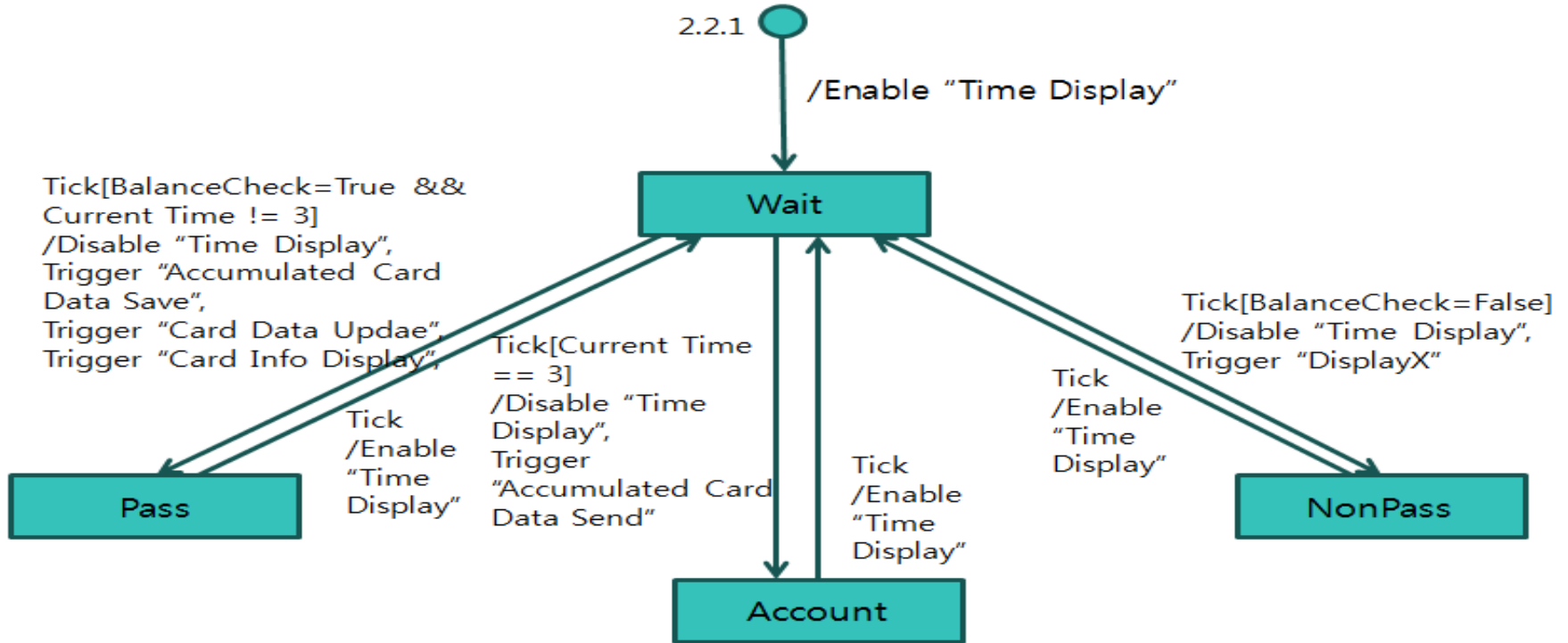
Tick[in && sensor]
/trigger "Unsetteled Check"

Tick[out && sensor]
/trigger "Out Process"



SRA ver4.0

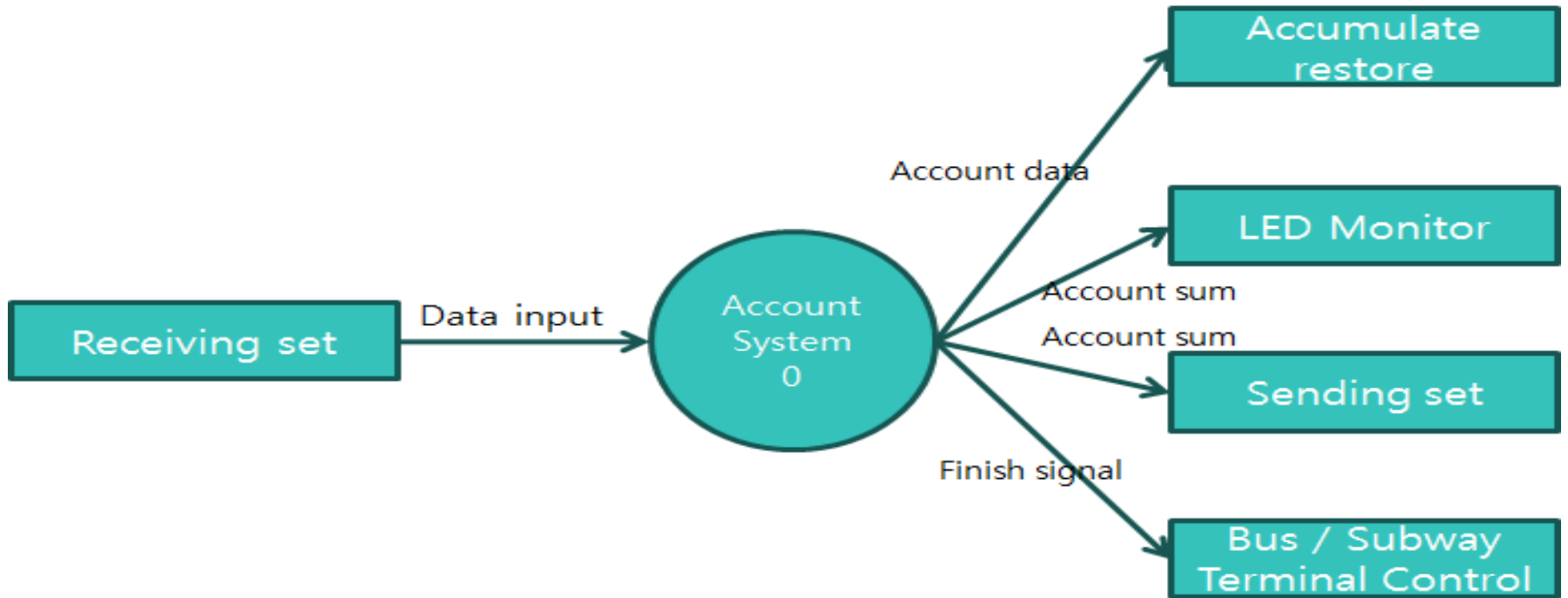
STD



SRA ver4.0

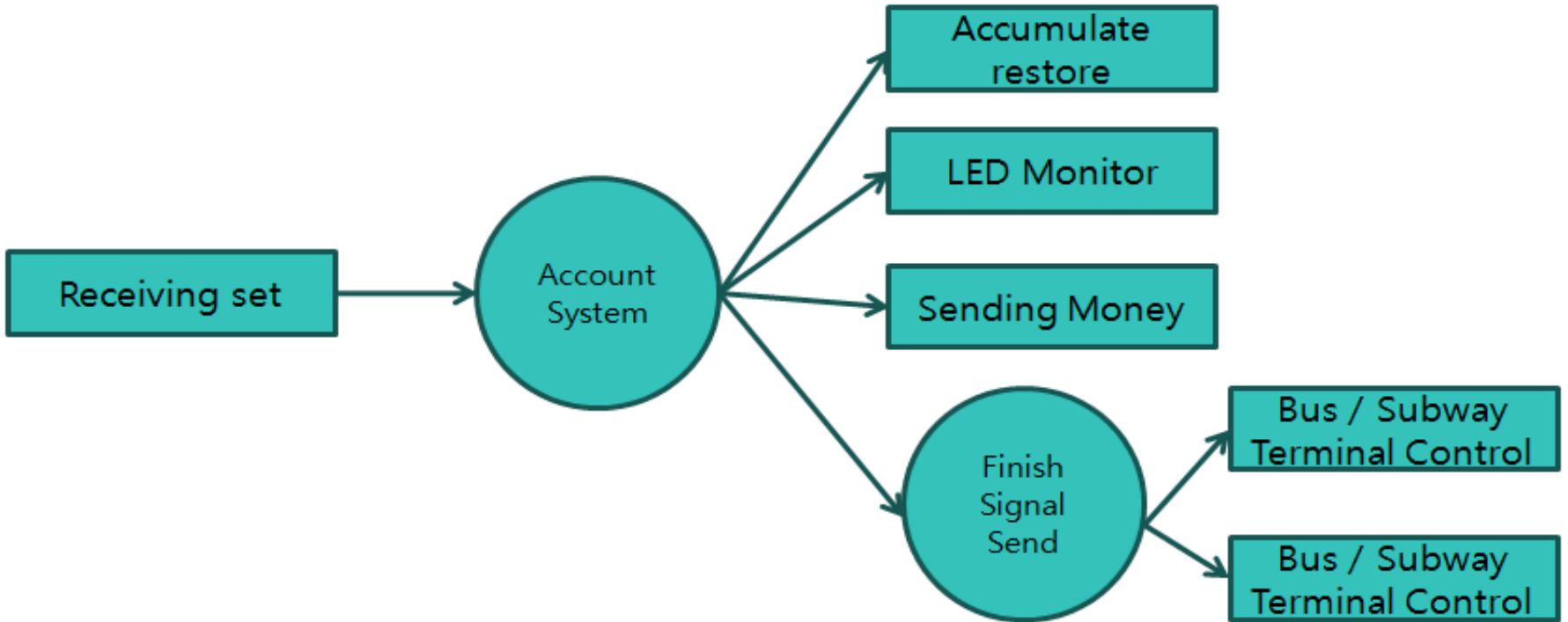
Account System DFD

DFD level 0



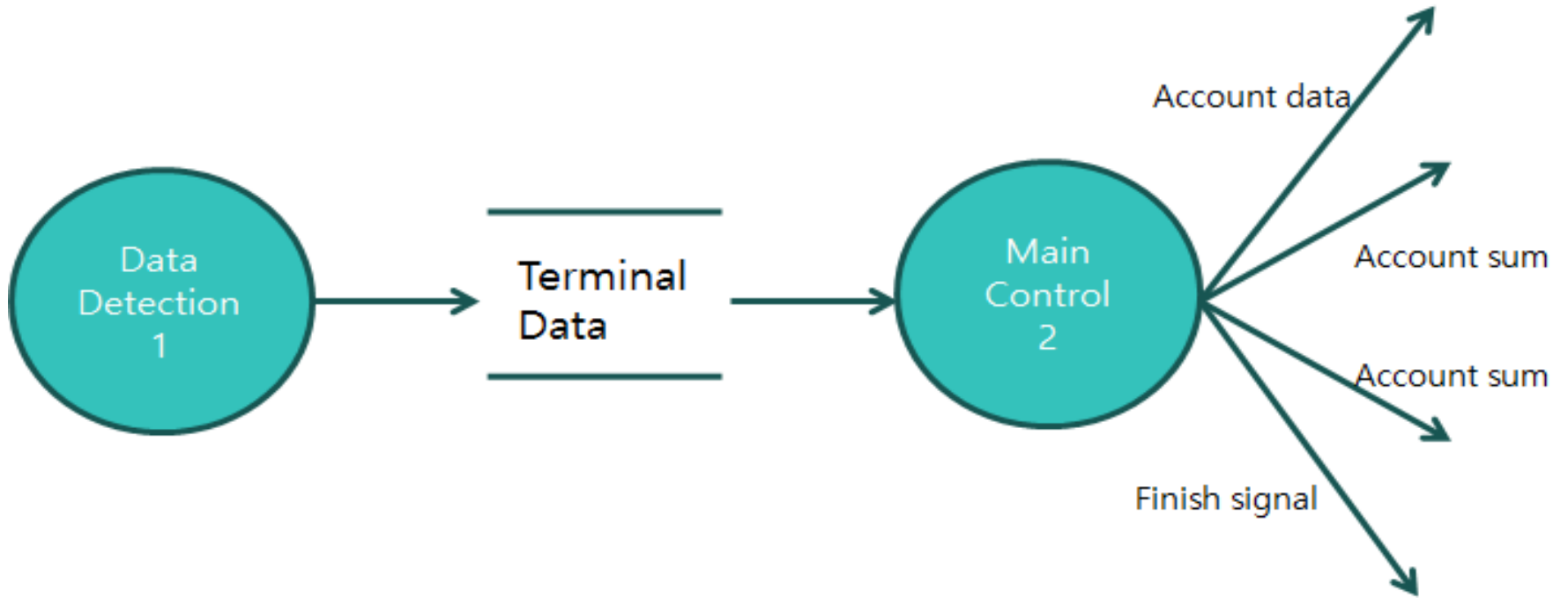
SRA ver1.0

DFD level 0



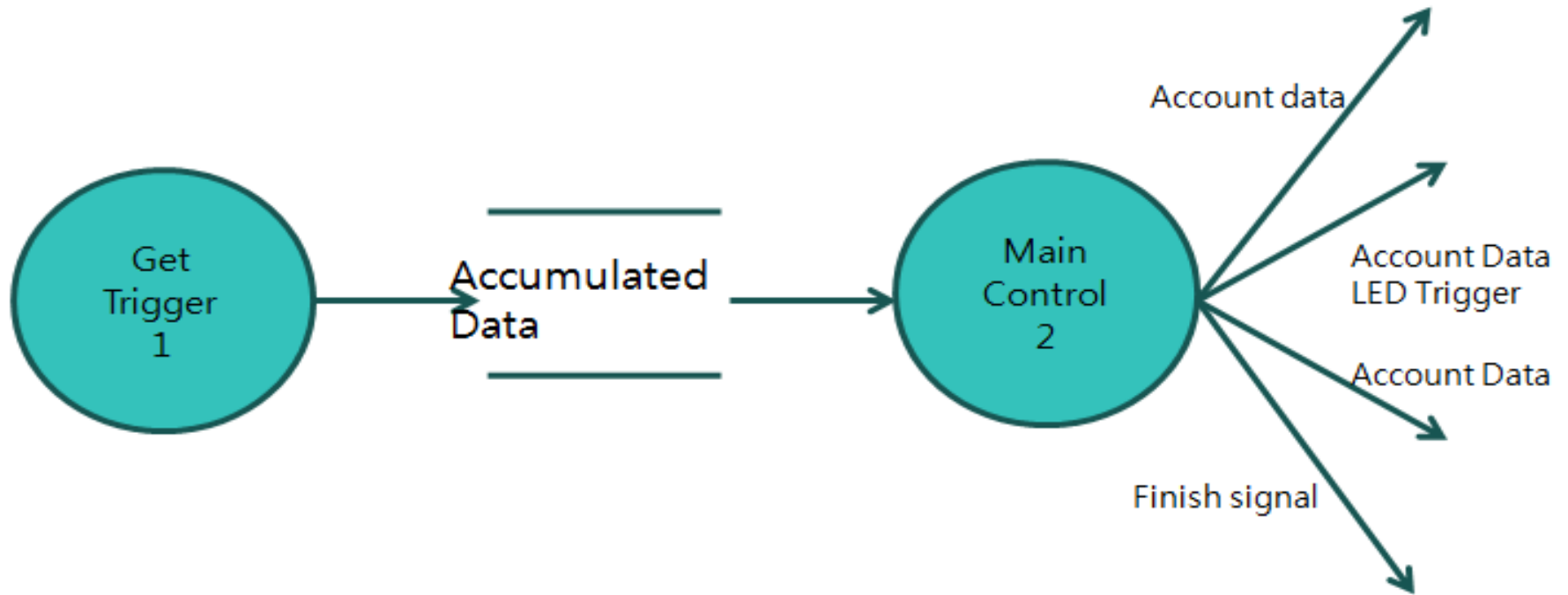
SRA ver4.0

DFD level 1



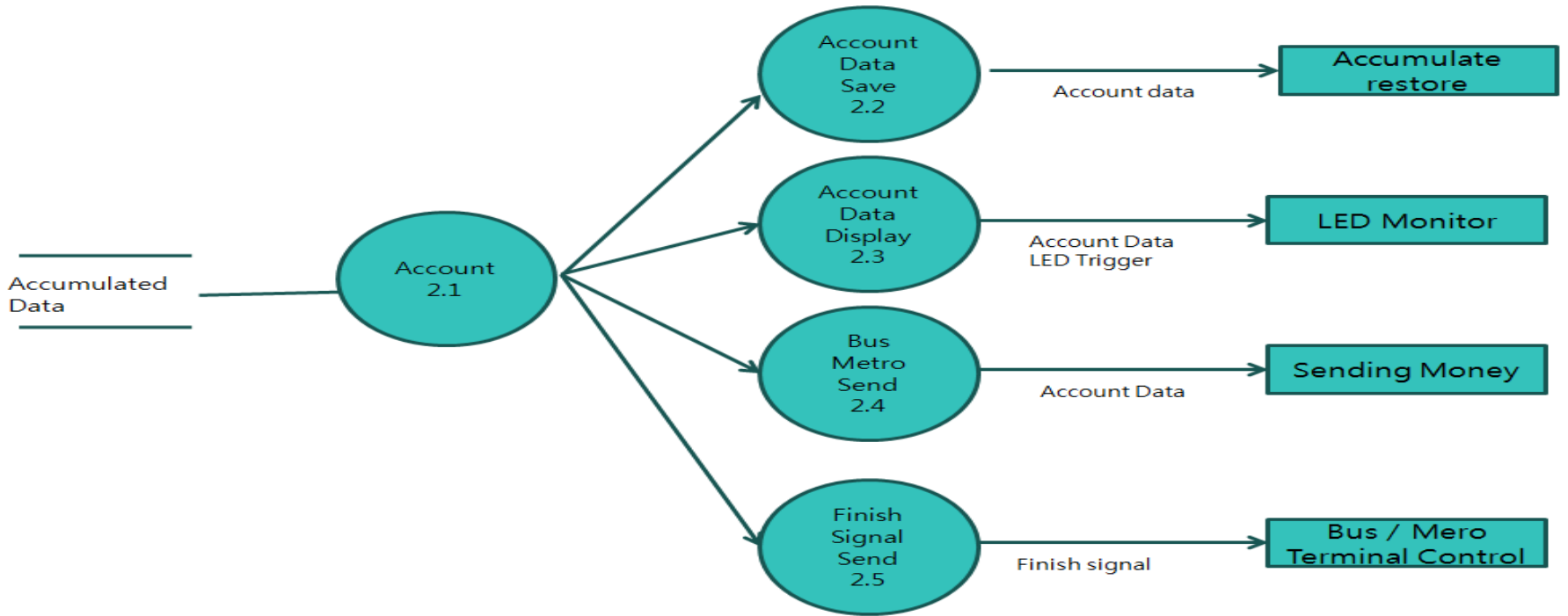
SRA ver1.0

DFD level 1



SRA ver4.0

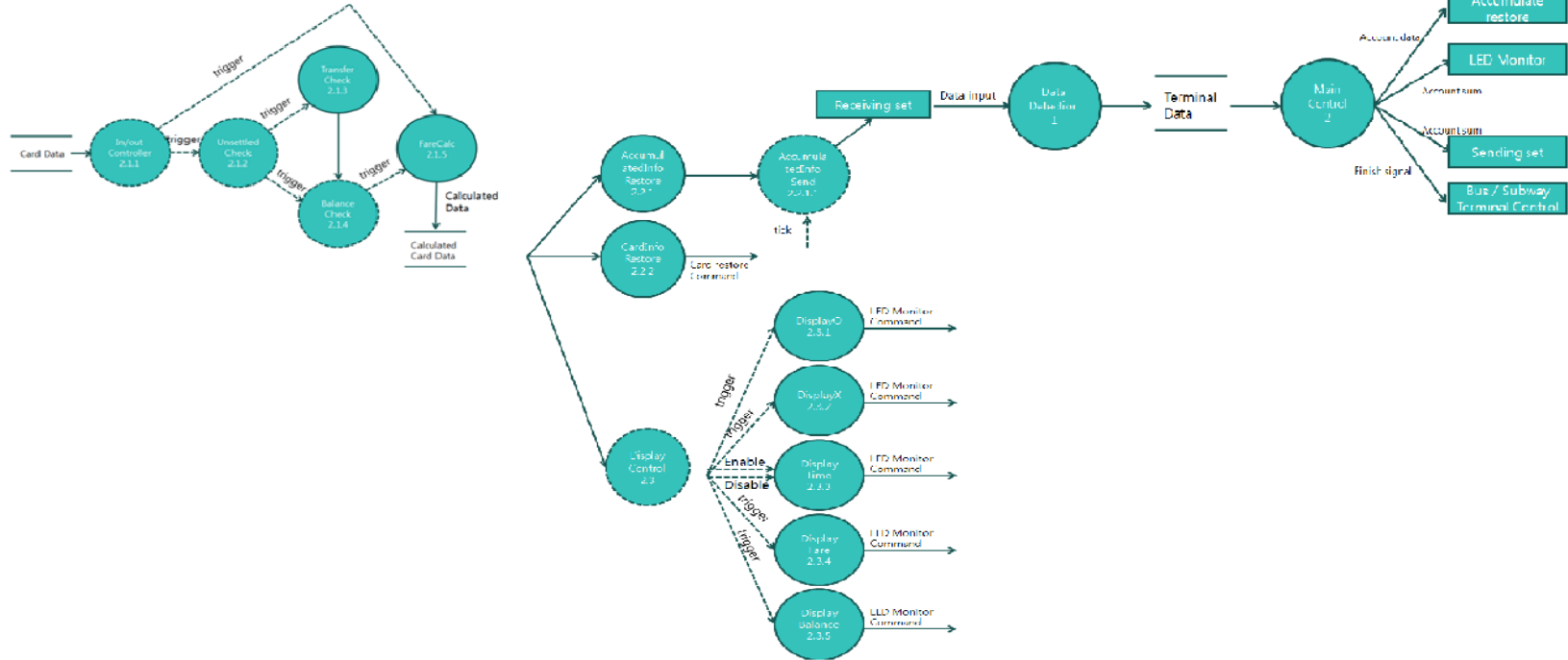
DFD level 2



SRA ver4.0 추가

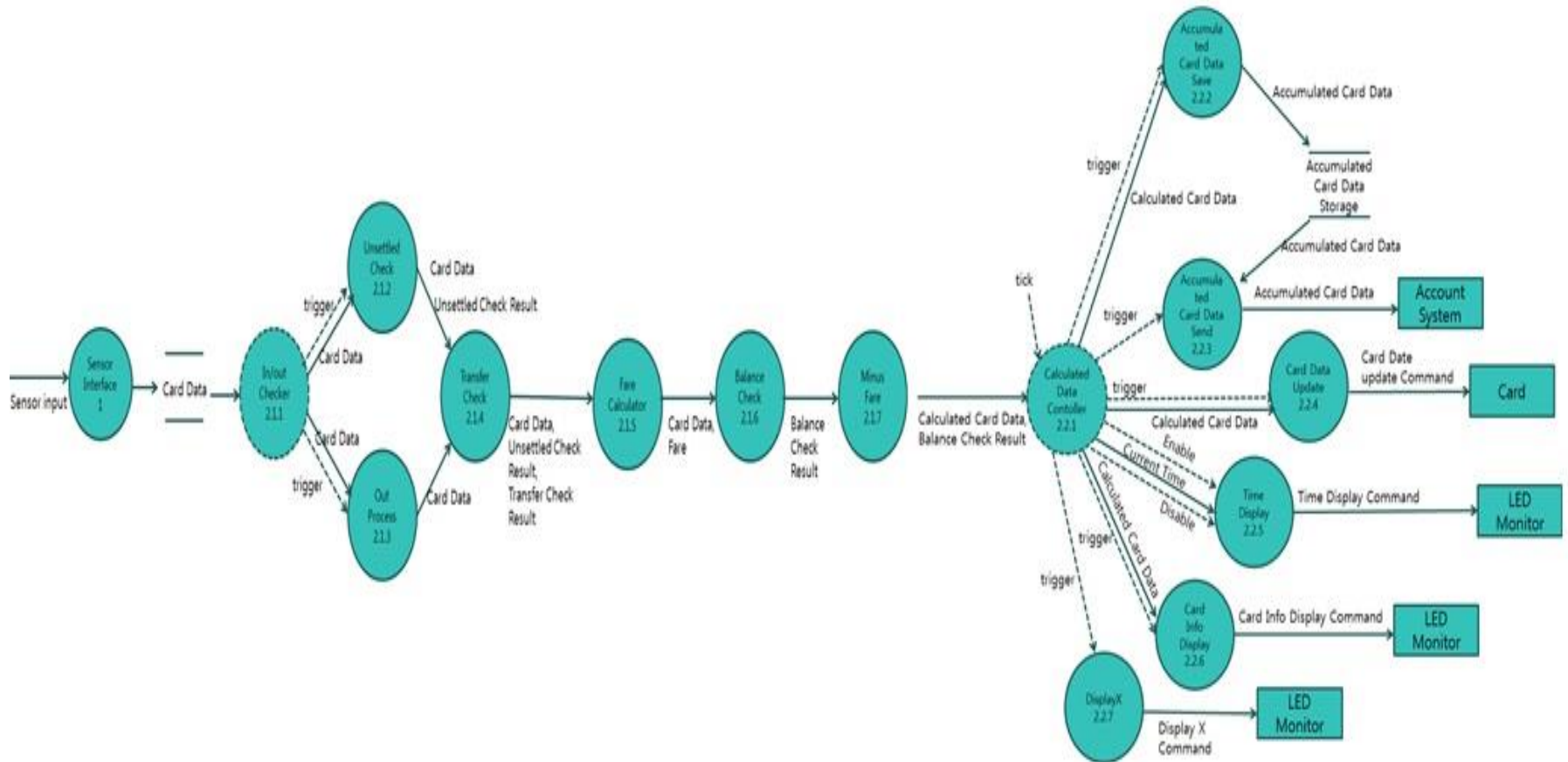
Overall

SRA ver1.0



Overall

SRA ver4.0

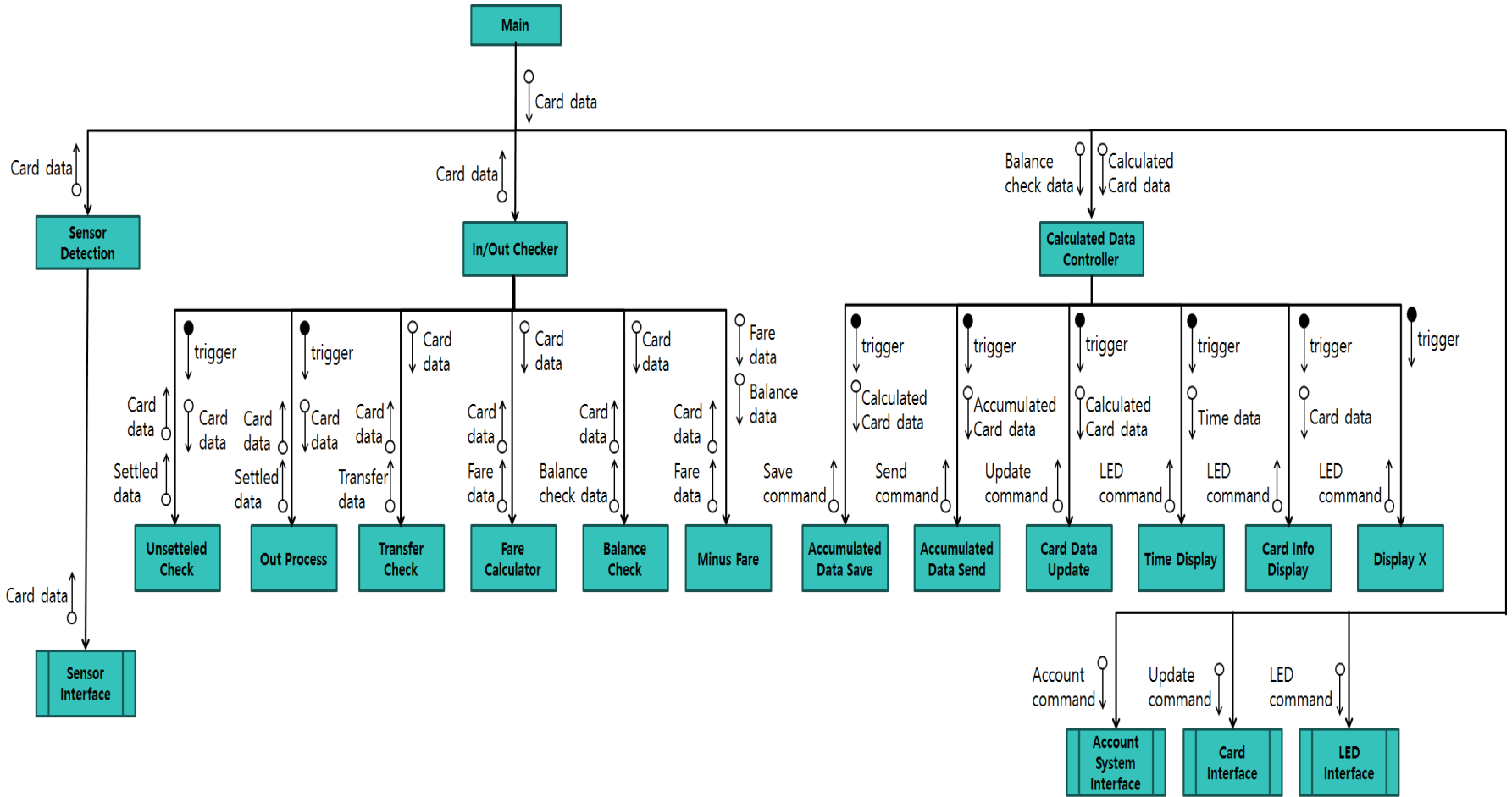


SD

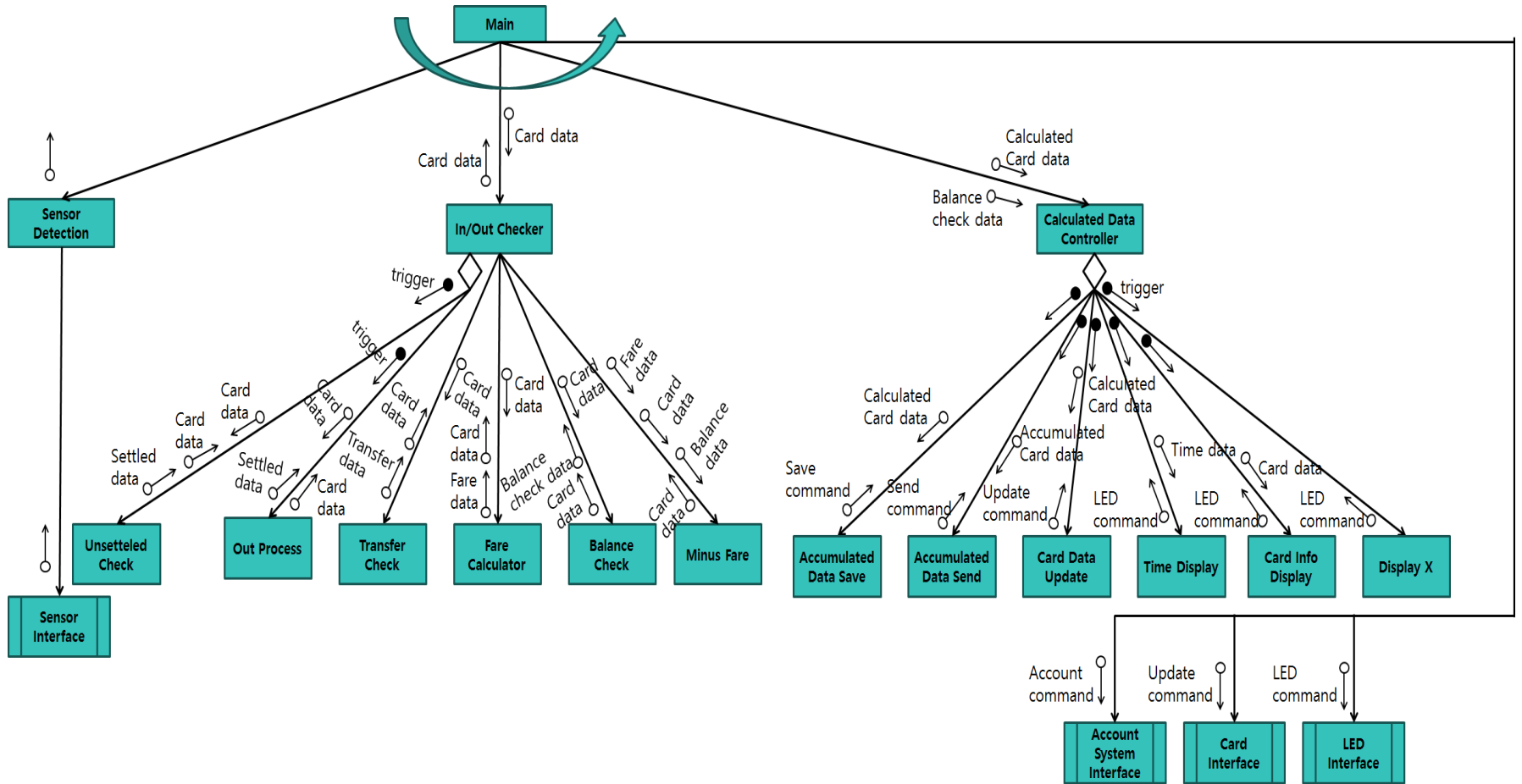
Terminal

Structured Charts (Basic)

SDS ver1.0

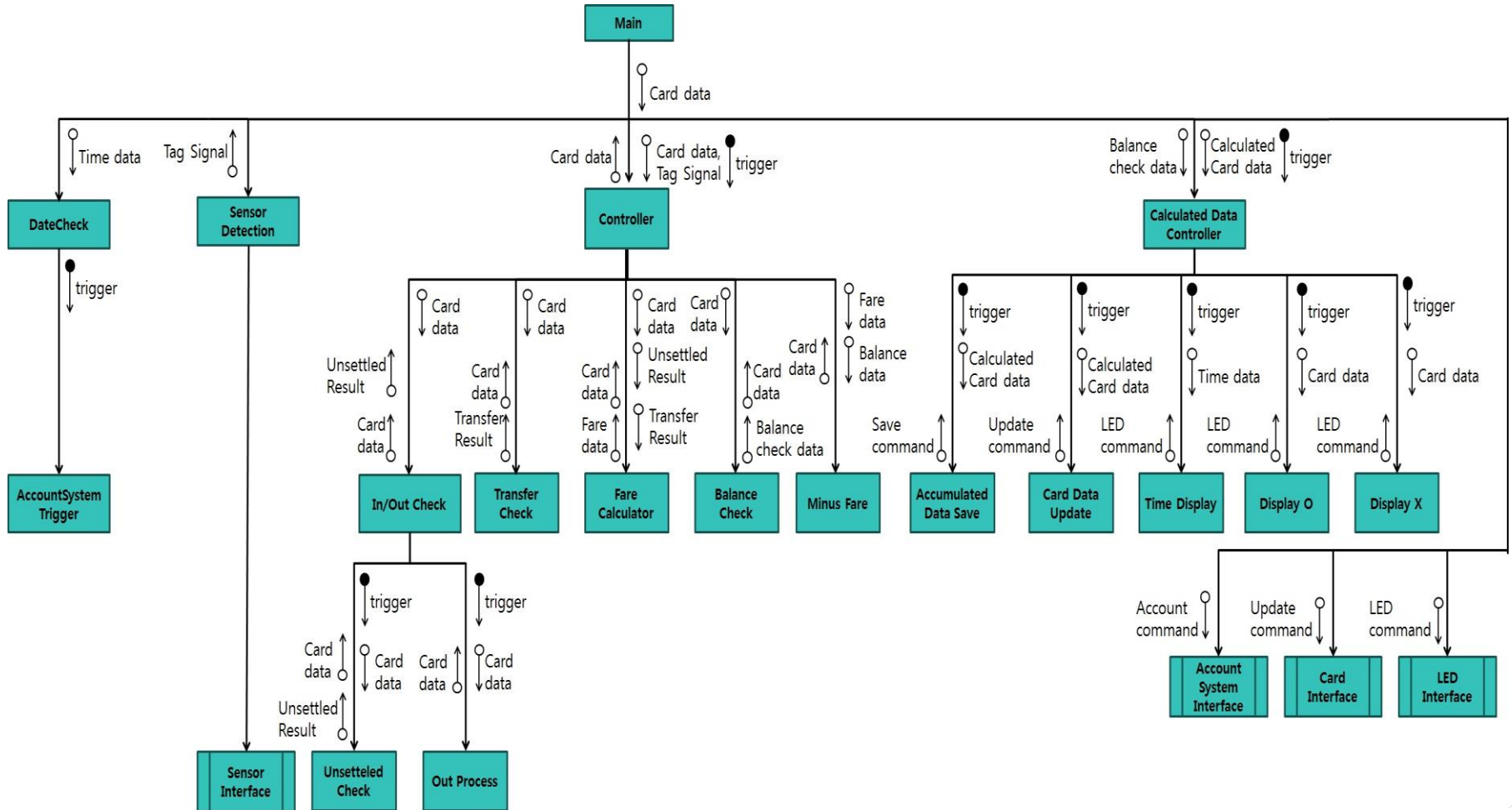


Structured Charts (Advanced) SDS ver1.0

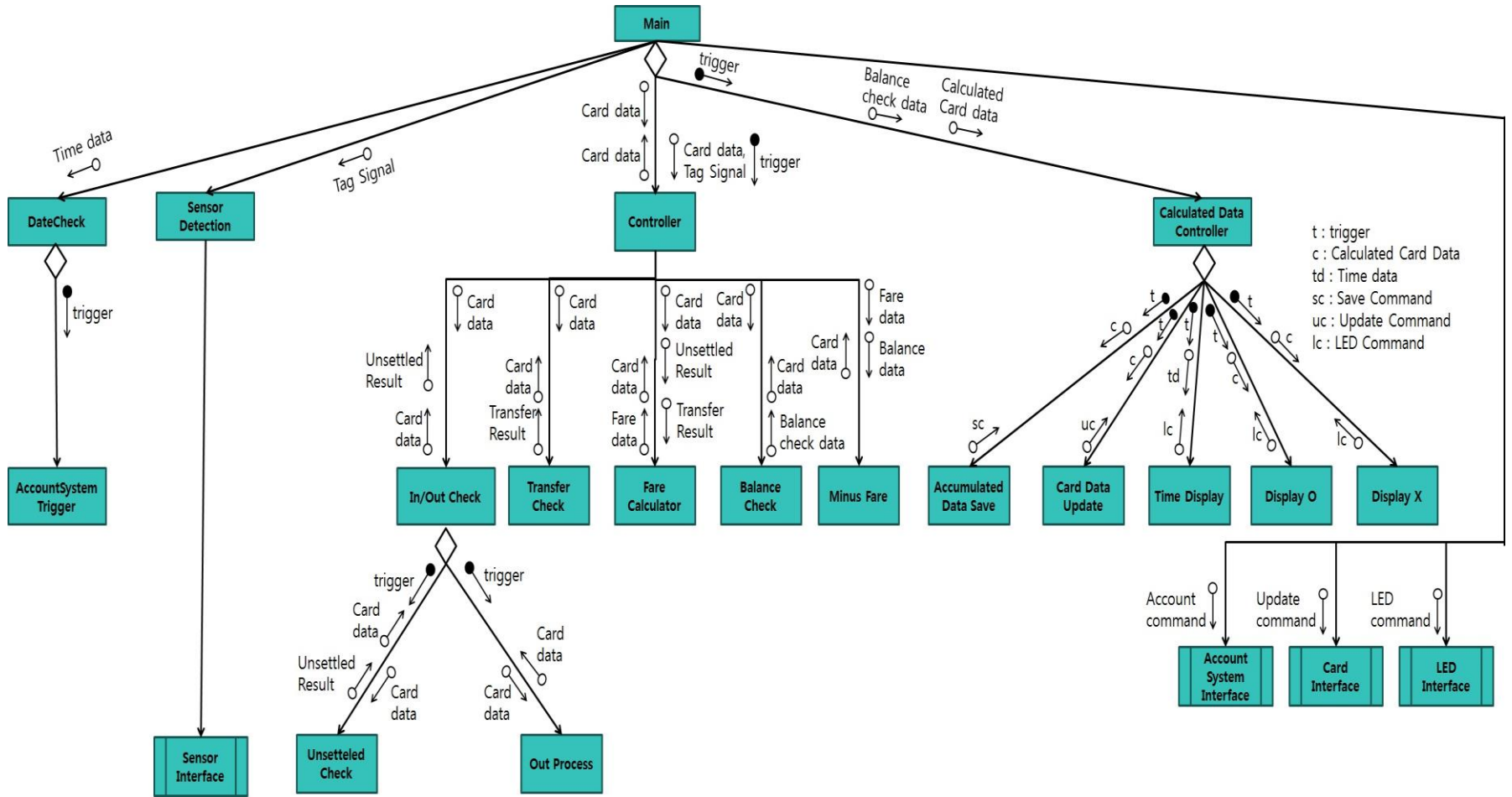


Structured Charts (Basic)

SDS ver2.5



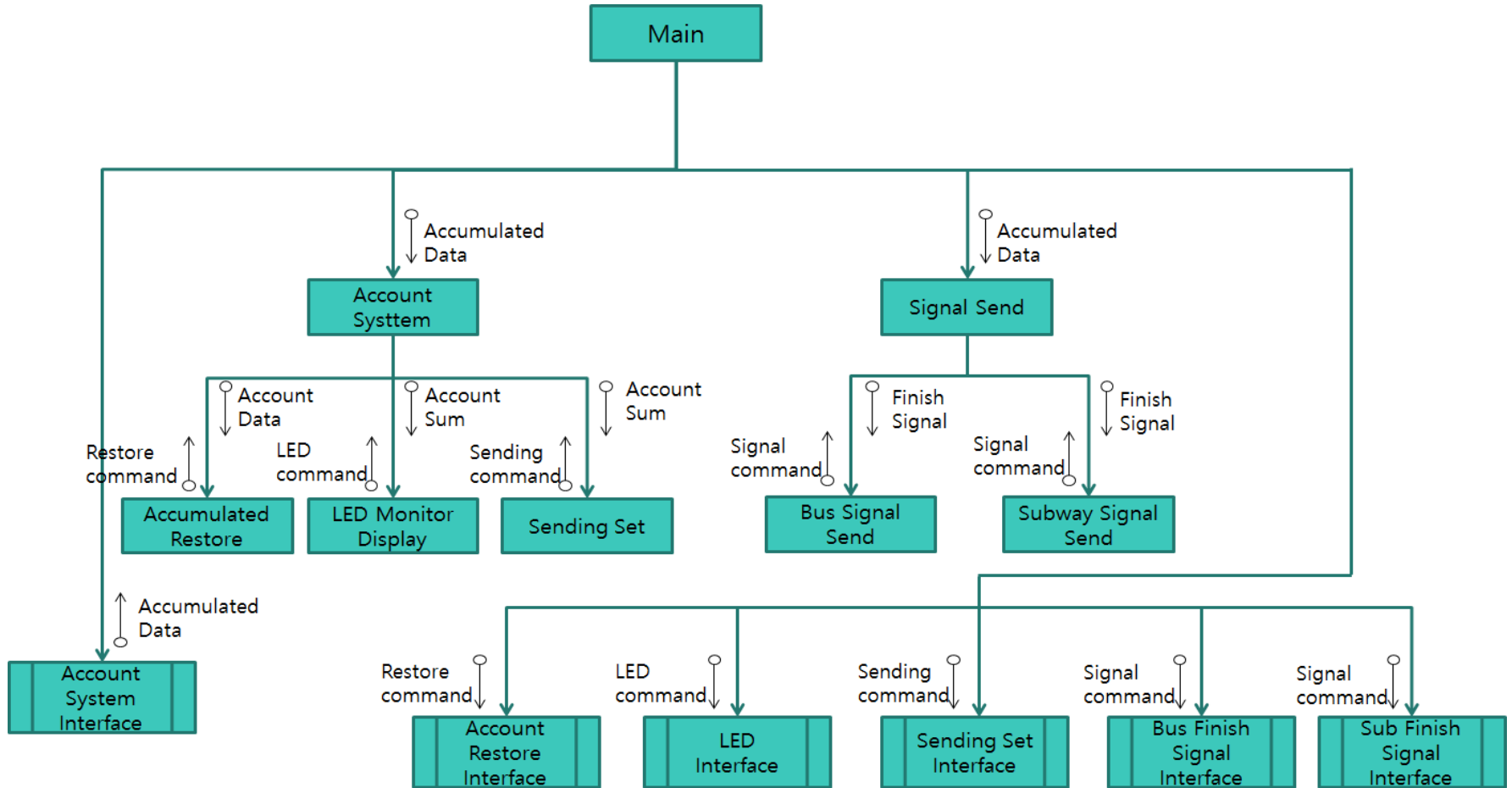
Structured Charts (Advanced) SDS ver2.5



Account System

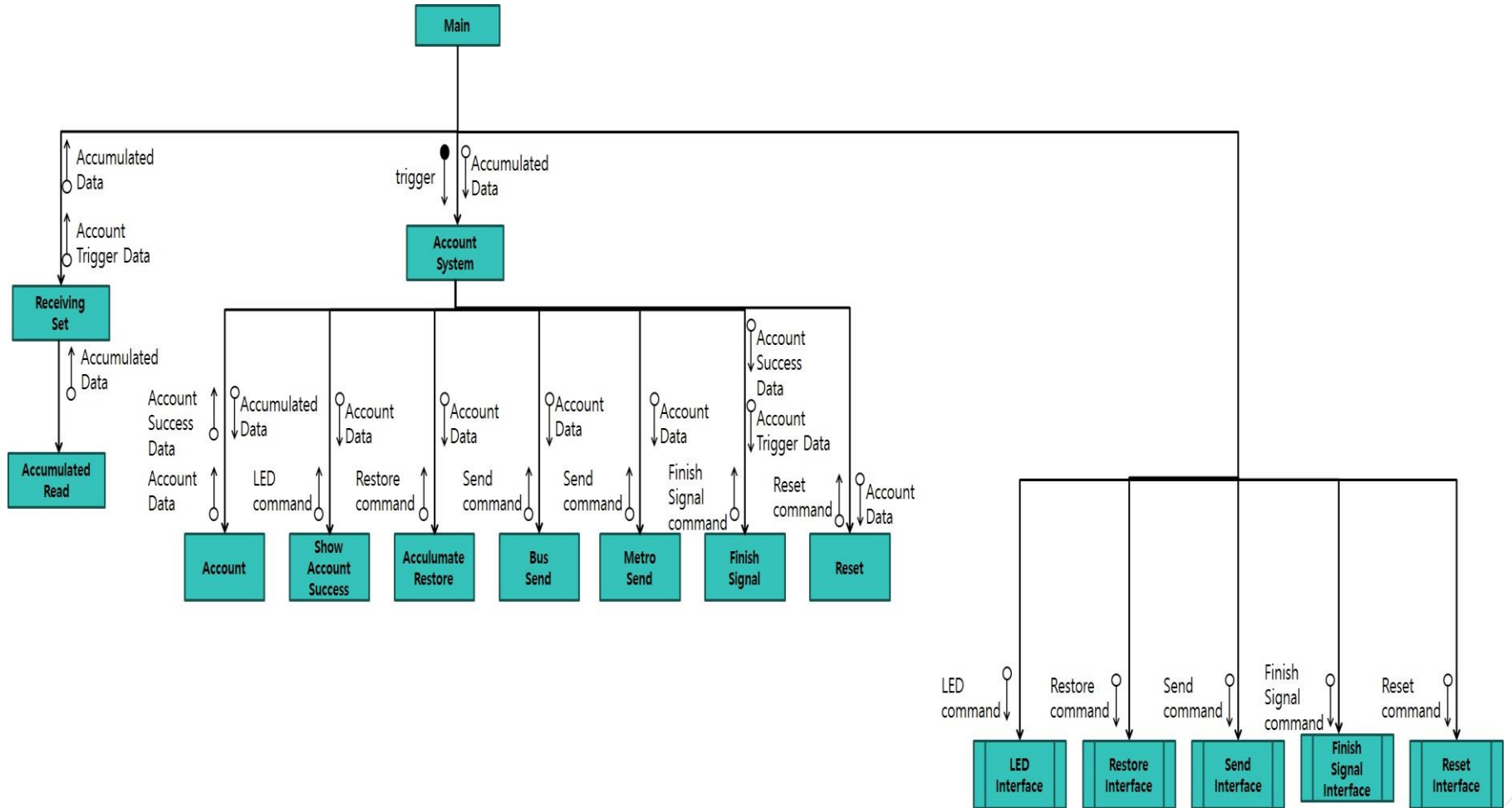
Structured Charts (Basic)

SDS ver1.0

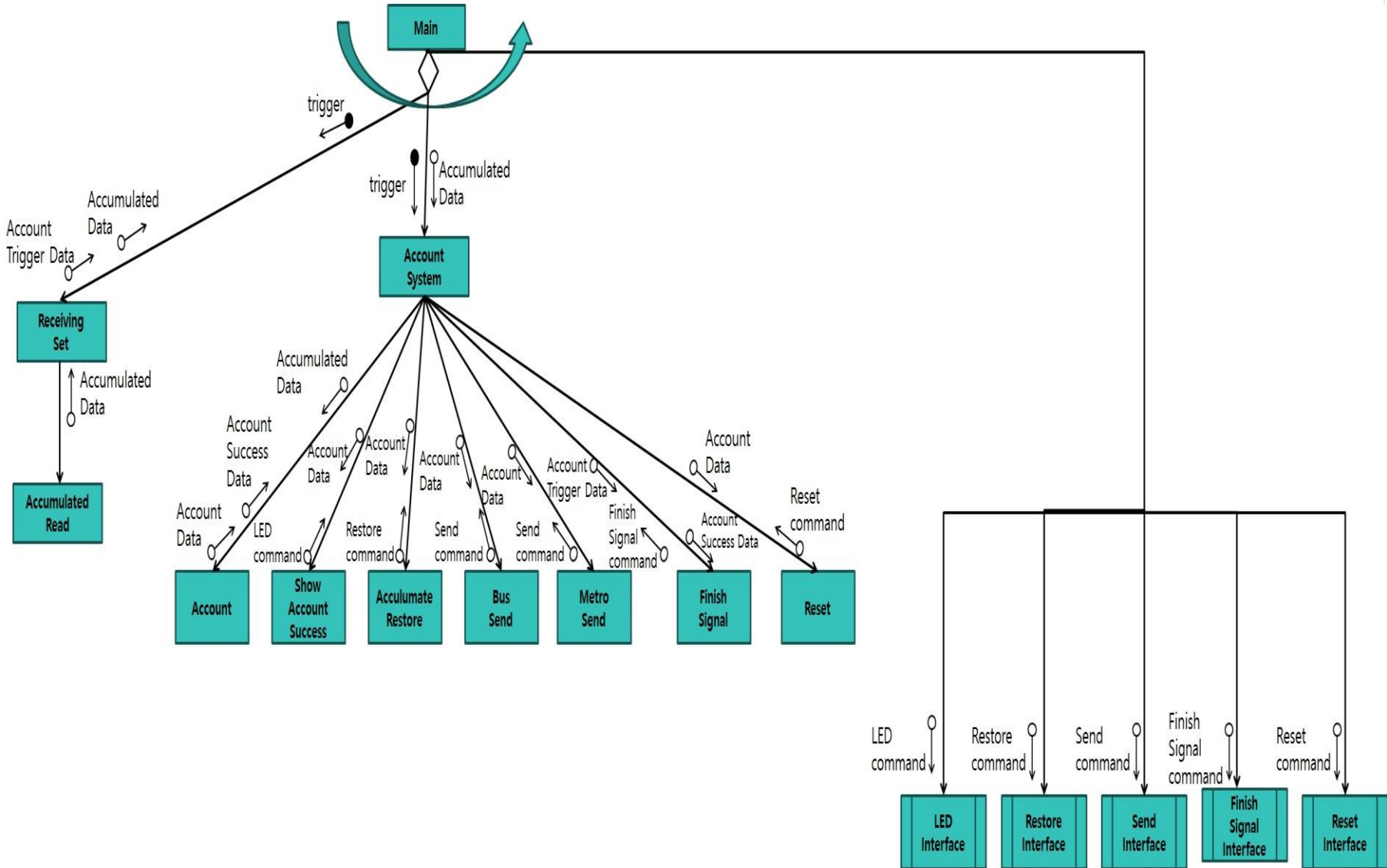


Structured Charts (Basic)

SDS ver2.5



Structured Charts (Advanced) SDS ver2.5 추가



Thank you

TEAM2의 발표를 마치겠습니다.