

CFG GENERATOR

T7

200711438 송인근

200711457 윤홍국

200711470 정재호

Contents

- ❖ Statement of purpose
- ❖ DFD explanation
- ❖ Realization DFD
- ❖ Code example
- ❖ Demonstration
- ❖ Code constraints
- ❖ Q & A

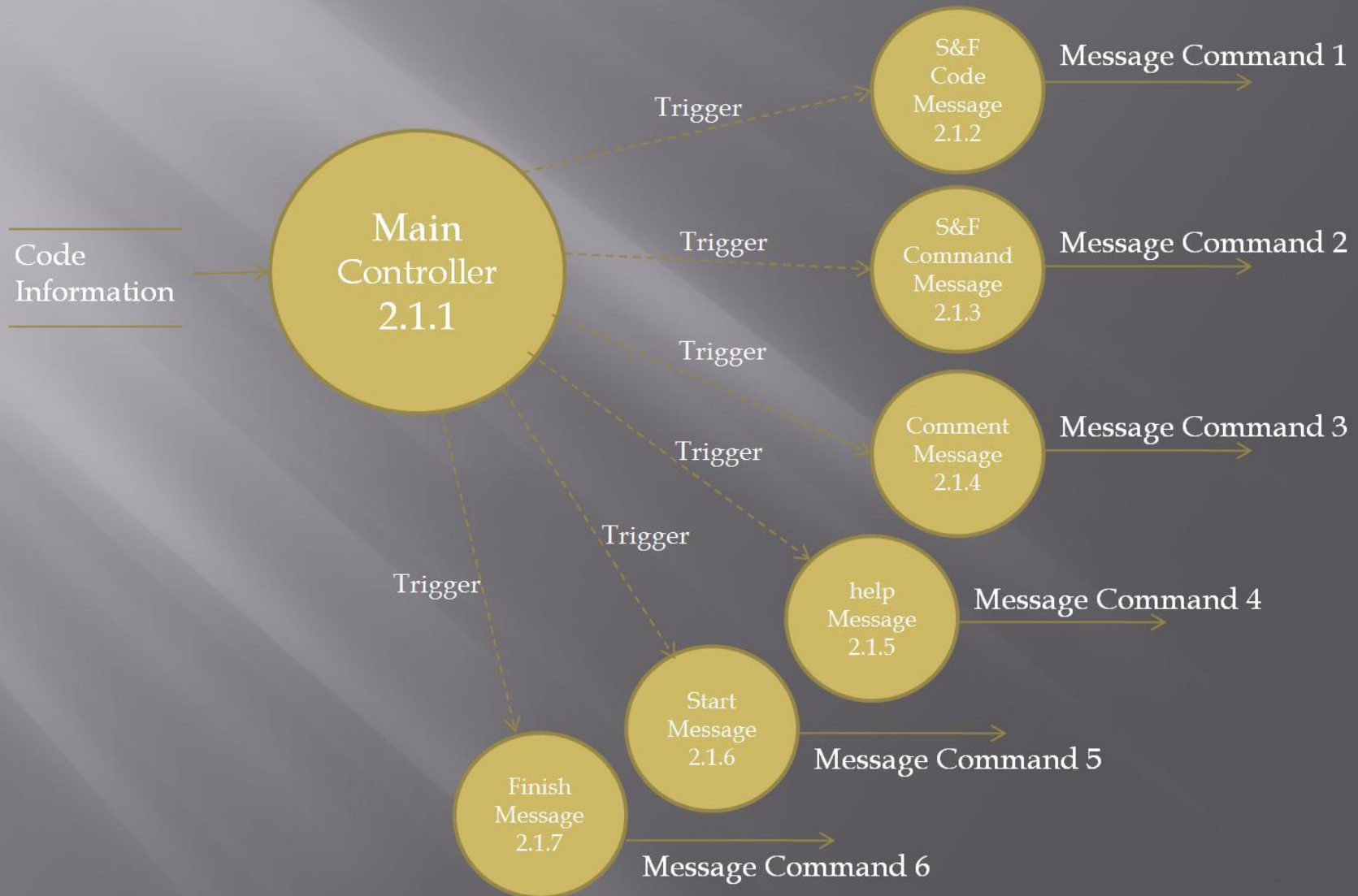
STATEMENT OF PURPOSE

Statement of purpose

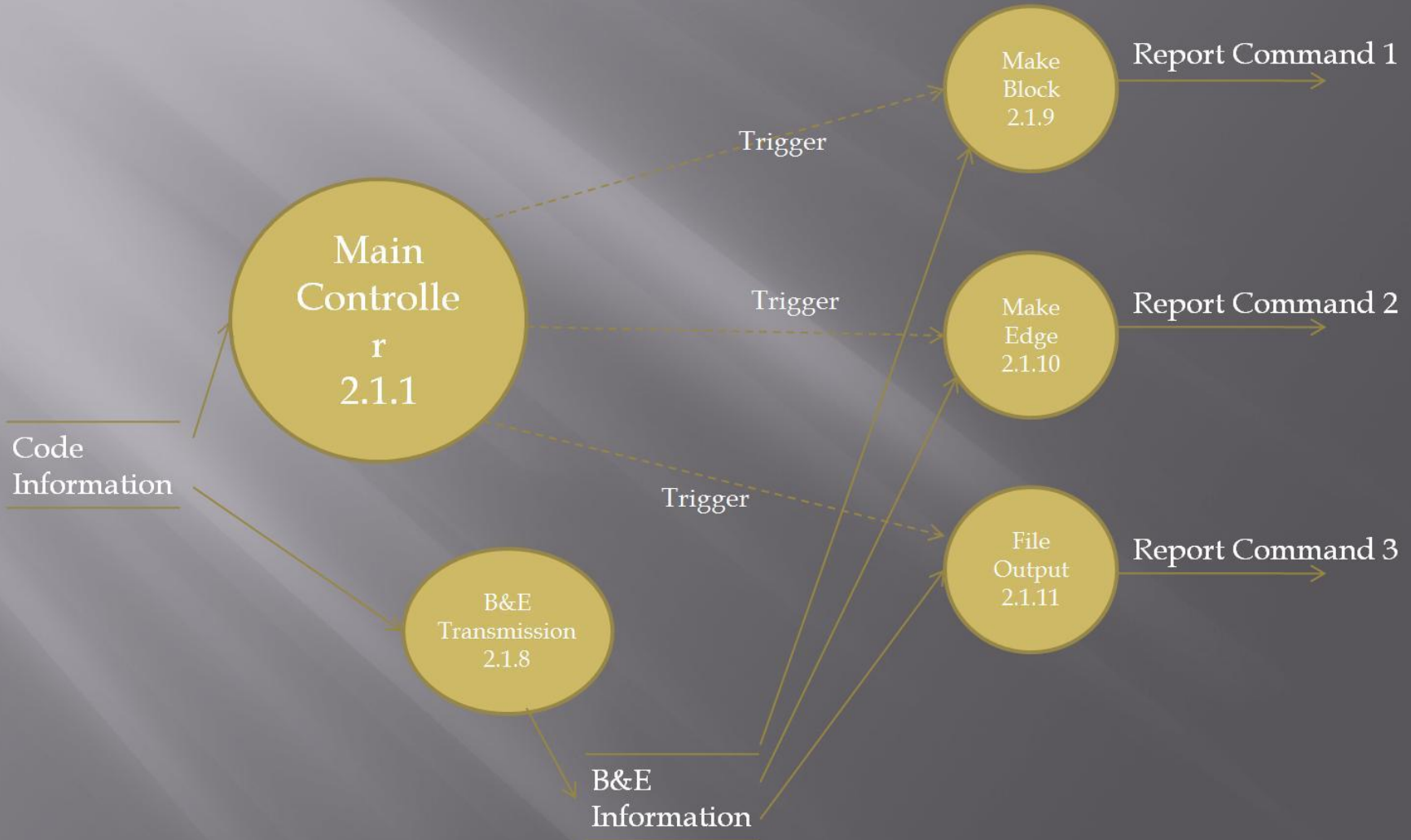
- ▣ CFG Generator draw CFG by C code
- ▣ When a user inputted unpermitted command, the program shows 'help' (that includes command syntax)
- ▣ C source code has 100~200 lines. It includes main function
- ▣ The C source code doesn't include pointers
- ▣ When c source code inputted successfully, program shows "success" message. Or in error case, the program shows "error" and terminates the program
- ▣ Before the program converting CFG, shows "converting" message
- ▣ UI is CUI(Character User Interface).A user uses command line in order to command our program. Report of our program shows list of 'states' and 'edges' of CFG
- ▣ Report with a text(*.txt) file

DFD EXPLANATION

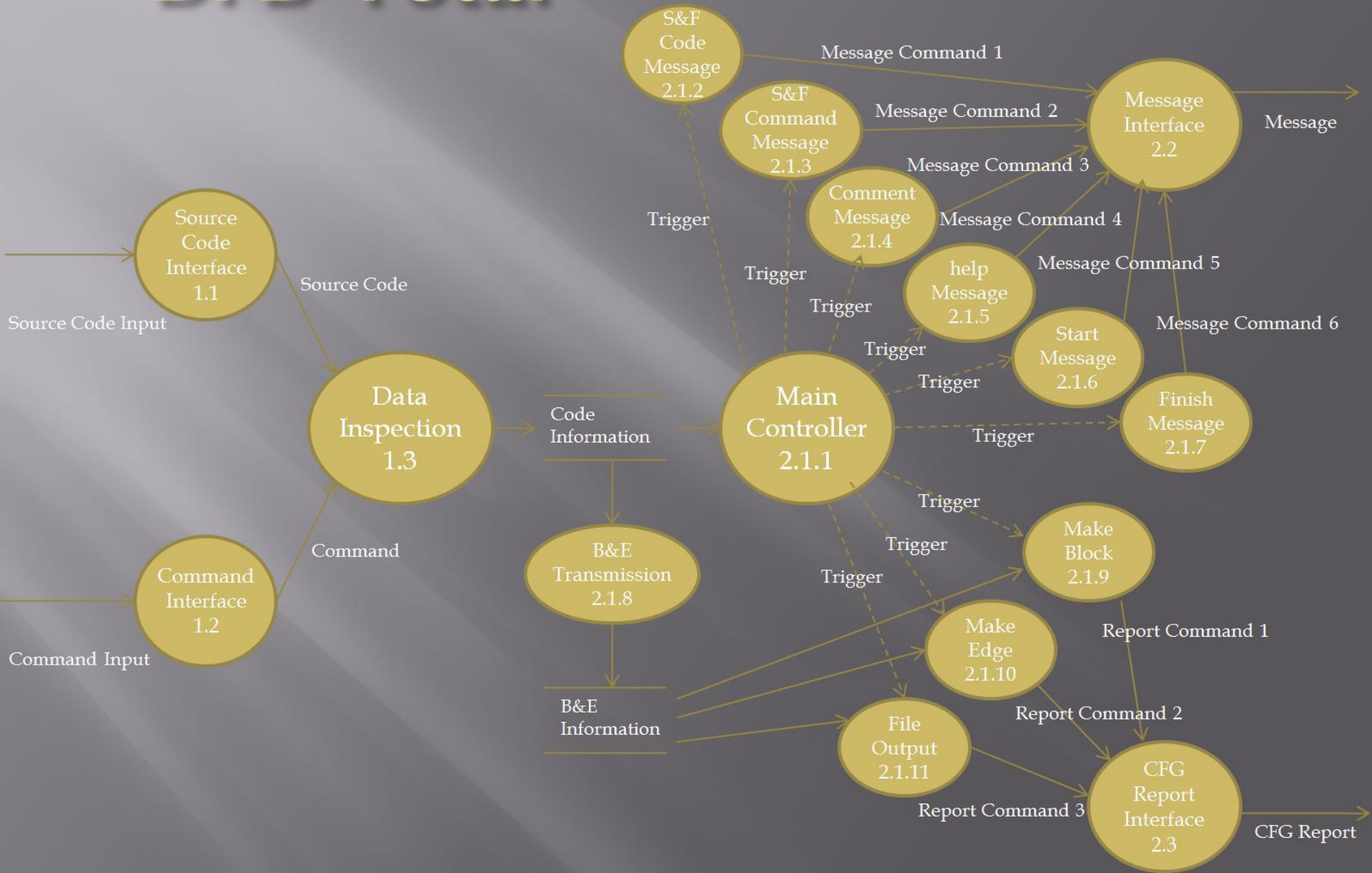
DFD Level 3 – CFG Generator



DFD Level 3 – CFG Generator

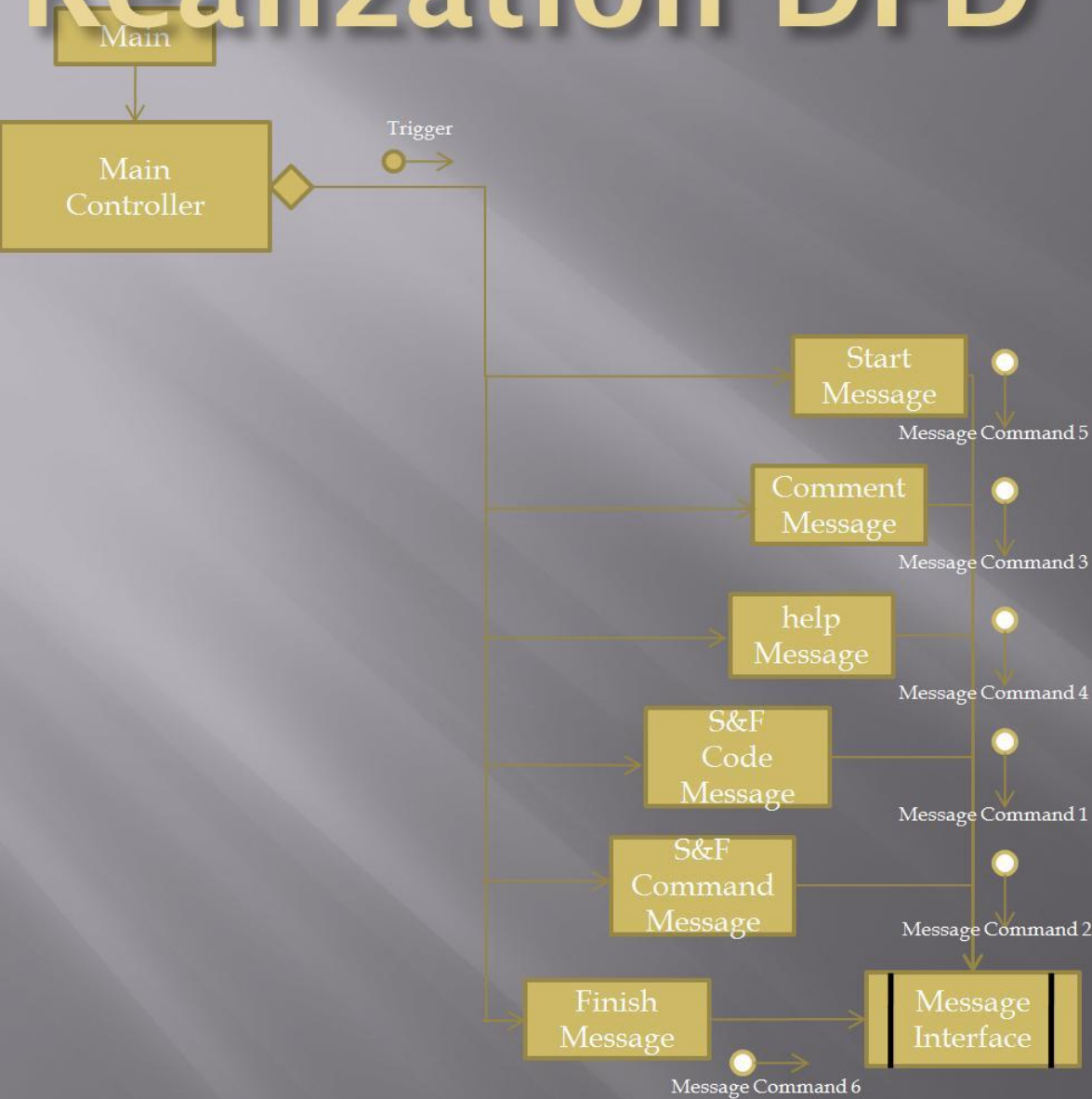


DFD Total



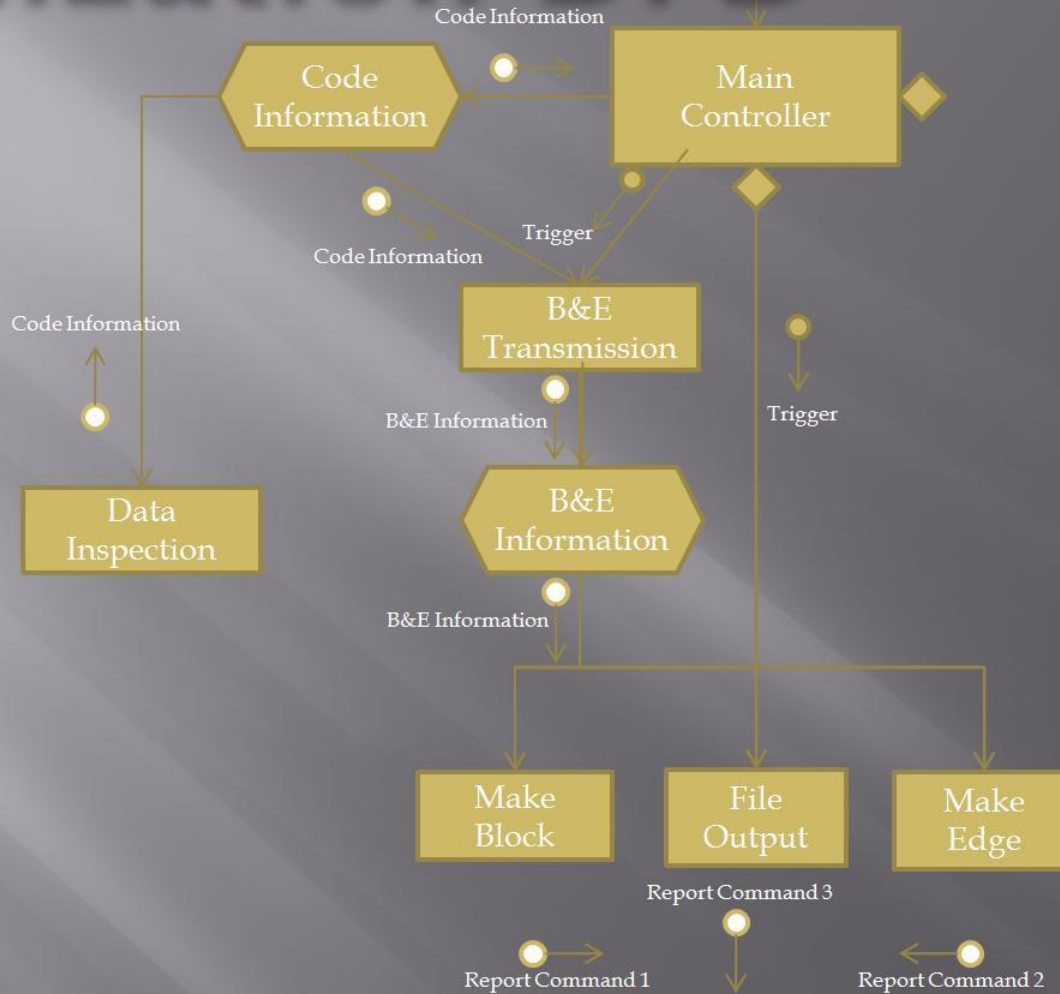
REALIZATION DFD

Realization DFD



```
void SnF_code_message(int num);  
void SnF_command_message(int num);  
void comment_message( );  
void help_message( );  
void start_message( );  
void finish_message(char* argv);
```

Realization DFD



```
void Make_Block(BE* be_set, FILE* file2);  
void Make_Edge(BE* be_set, FILE* file2, int* sub);  
void File_Output(BE* be_set, FILE* file2);  
void BE_Transmission(BE* be_set, int sel, char *buf, char *con);  
void Data_Inspection(char *buf, BE* be_set, int* block, int* edge, int* file, int* sub, char* copy, char* p);
```

CODE EXAMPLE

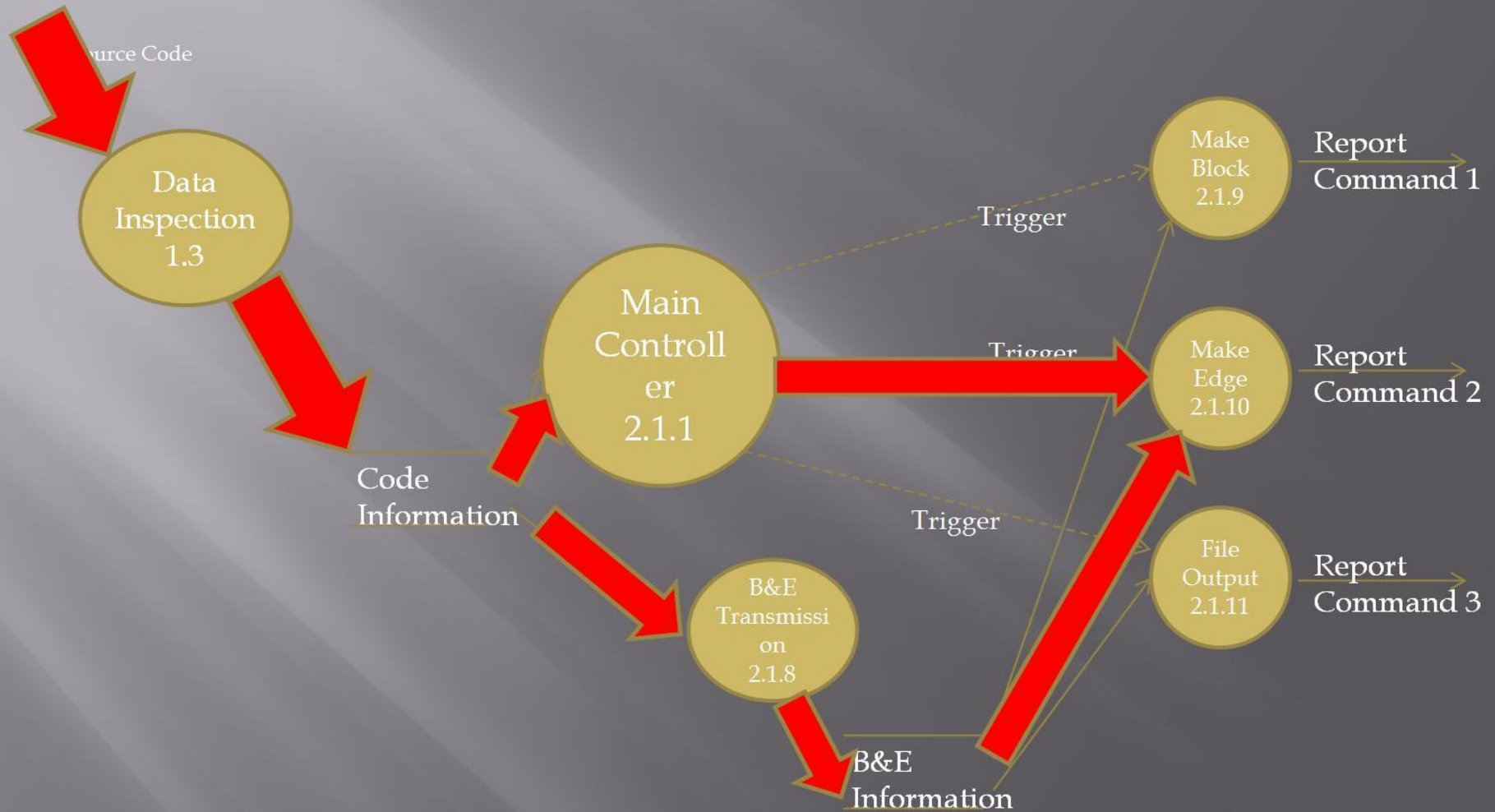
Code example

```
정재호@~ -PC ~  
$ ./CG test.c report.txt  
Command is valid  
File open succeeded  
File Transformation Start  
File Name : report.txt  
Program completed normally
```

Code example

```
if(ch == 'Y' || ch == 'y')
{
    printf("Continue? (Y / N) : ");
    scanf("%c", ch);
    continue;
}
```

Code example



Code example

```
if(ch == 'Y' || ch == 'y')
{
    printf("Continue? (Y / N) : ");
    scanf("%c", ch);
    continue;
}
```



```
-----
Block #0
if(ch == 'Y' || ch == 'y')
-----
    ↓ ch == 'Y' || ch == 'y'
-----
Block #1
printf("Continue? (Y / N) : ");
scanf("%c", ch);
continue;
-----
    ↓ (if is ended)
-----
Block #2
```


Code example

```
for(i = 0 ; i < choice ; i++)
{
    printf("&d x %d = %d #n", i, i, i*i);
    printf("Continue? (Y / N) : ");
    scanf("%c", ch);
}
```



```
-----
Block #0
for(i = 0 ; i < choice ; i++)
-----
    ↓ i = 0 ; i < choice ; i++
-----
Block #1
printf("&d x %d = %d #n", i, i, i*i);
printf("Continue? (Y / N) : ");
scanf("%c", ch);
-----
    ↓ (for is ended)
-----
Block #2
```

Code example

```
while(choice>30)
{
    printf("Your Choice (1 / 2) : ");
    scanf("%d", choice);
}
```



```
-----
Block #0
while(choice>30)
-----
    ↓ choice>30
-----
Block #1
printf("Your Choice (1 / 2) : ");
scanf("%d", choice);
-----
    ↓ (while is ended)
-----
Block #2
```

Code example

```
switch(choice)
{
    case 1:
        printf("&d x %d = %d \n", i, i, i+i);
        break;
    case 2:
        printf("&d x %d = %d \n", i, i, i+i);
        break;
    default:
        printf("Invalid! \n");
        break;
}
```



```
-----
Block #0
switch(choice)
-----
    ↓ choice
-----

Block #1
case 1:
printf("&d x %d = %d \n", i, i, i+i);
break;
case 2:
printf("&d x %d = %d \n", i, i, i+i);
break;
default:
printf("Invalid! \n");
break;
-----
    ↓ (switch is ended)
-----

Block #2
|
```

Code example

```
while(choice>30)
{
    printf("Your Choice (1 / 2) : ");
    scanf("%d", choice);
    if(ch == 'Y' || ch == 'y')
    {
        continue;
    }
}
```



```
-----
Block #0
while(choice>30)
-----
    ↓ choice>30
-----
Block #1
printf("Your Choice (1 / 2) : ");
scanf("%d", choice);
if(ch == 'Y' || ch == 'y')
-----
    ↓ ch == 'Y' || ch == 'y'
-----
Block #2
continue;
-----
    ↓ (if is ended)
-----
Block #3
-----
    ↓ (while is ended)
-----
Block #4
```

DEMONSTRATION

CODE CONSTRAINTS

Code constraints

- ❖ Branch must be in this form

- ❖ Do not make else-case-CFG

```
Branch(condition)
```

```
{
```

```
//////////
```

- ❖ Do not make do-while-case-CFG

Q & A

THANK YOU!!!!!!!