SOFTWARE ENGINEERING TEAM PROJECT

TEAM : T8

(이자형 김은빈 오고은)

담당교수님 : 유준범교수님

담당조교님 : 윤상현조교님

KU 건국대학교 KONKUK UNIV.

발표순서

① . STATEMENT OF PURPOSE

② . SYSTEM CONTEXT DIAGRAM

③ . EVENT LIST

④ . DATA FLOW DIAGRAM
- DATA DICTIONARY

⑤ . TOTAL DFD

SOFTWARE ENGINEERING #T8

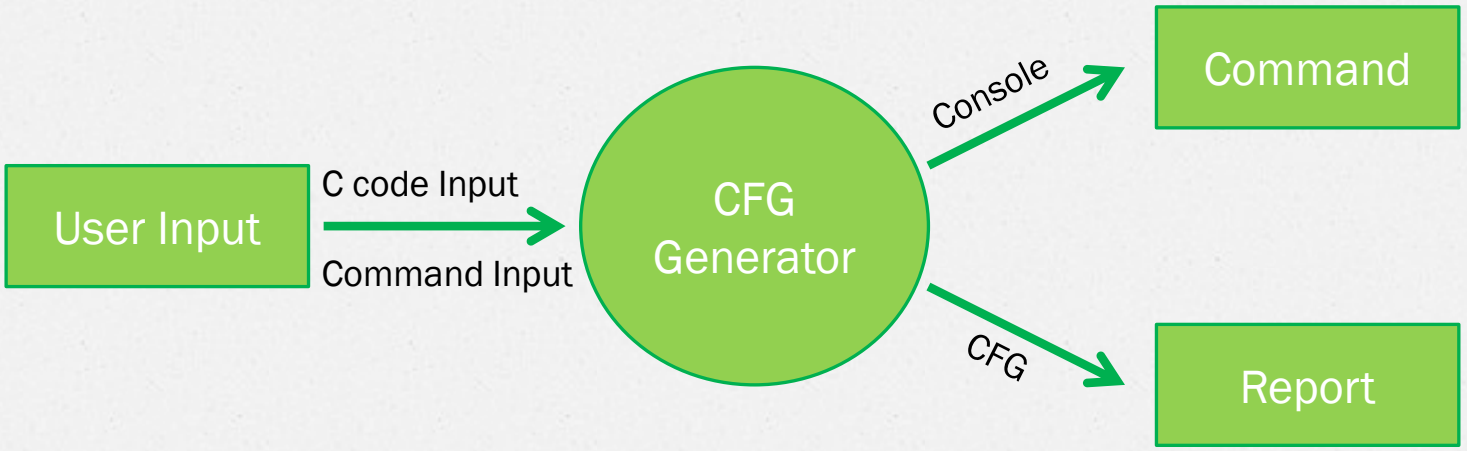
KU 건국대학교 KONKUK UNIV.

STATEMENT OF PURPOSE

- This program takes source code written in C language and its source code to draw the CFG aims to develop tools.
- The name of the software development shall be determined by the CFG GENERATOR.
- STATE REPORT of the CFG LIST list and a list of EDGE is shown in the form.
- Output a success message when it succeeds and fails, or an invalid command is entered, the error message and terminates the program.

KU 건국대학교 KONKUK UNIV.

SYSTEM CONTEXT DIAGRAM



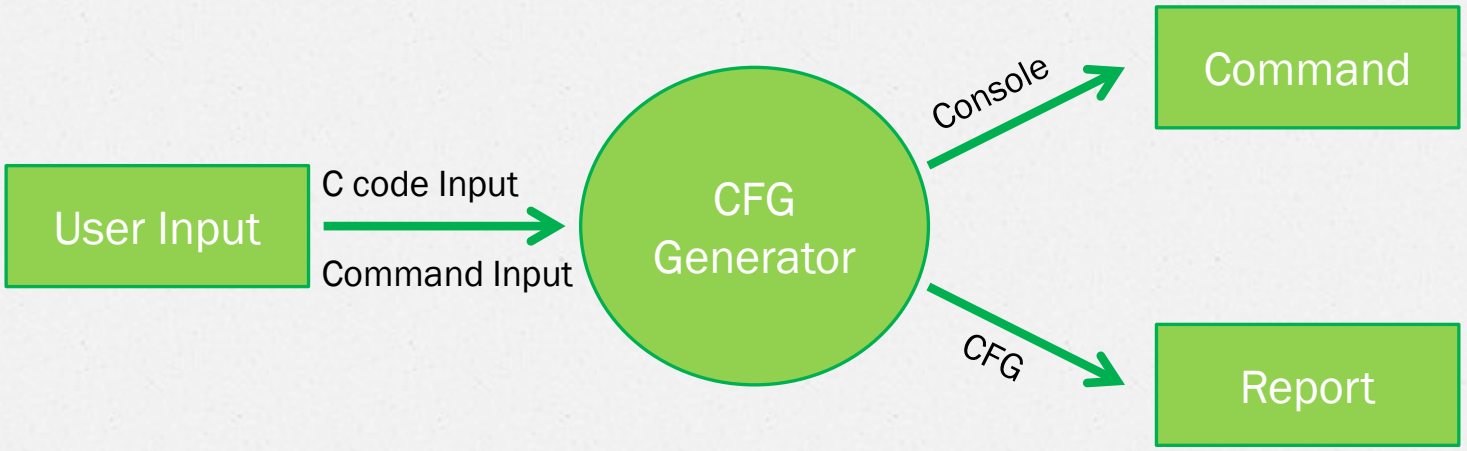
KU 건국대학교 KONKUK UNIV.

EVENT LIST

Input / output Event	description
User Input	C code, Command Input
Command	Print command Result List
Report	Print command CFG List

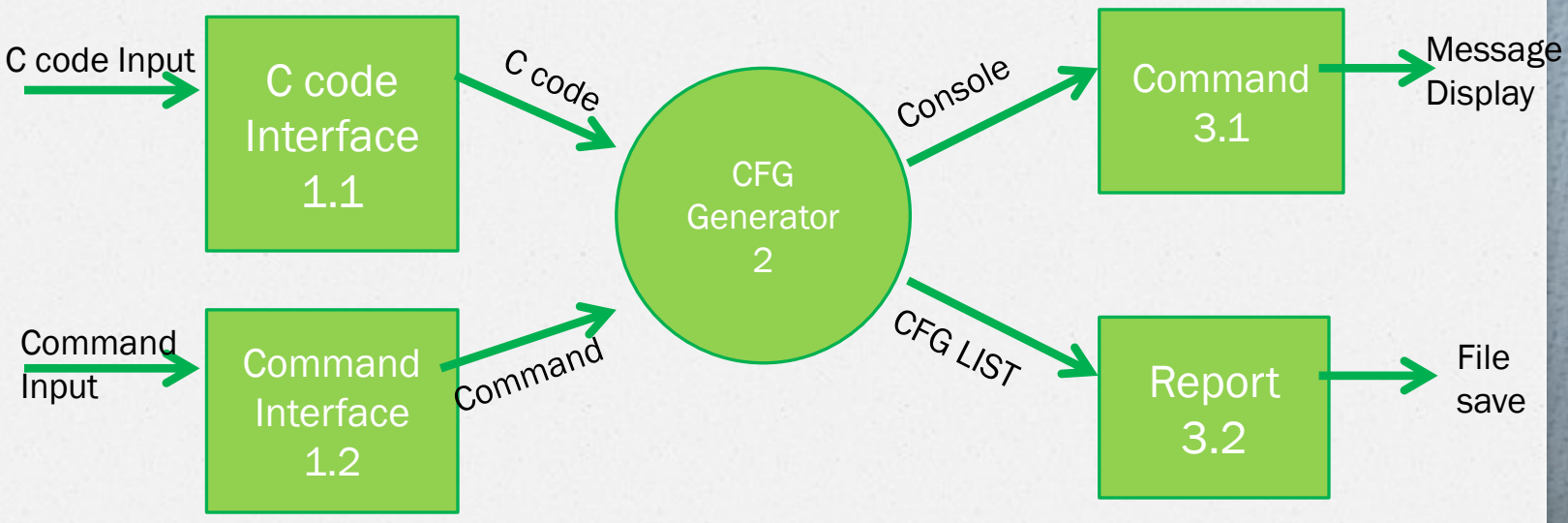
KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM LEVEL 0



KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM LEVEL 1



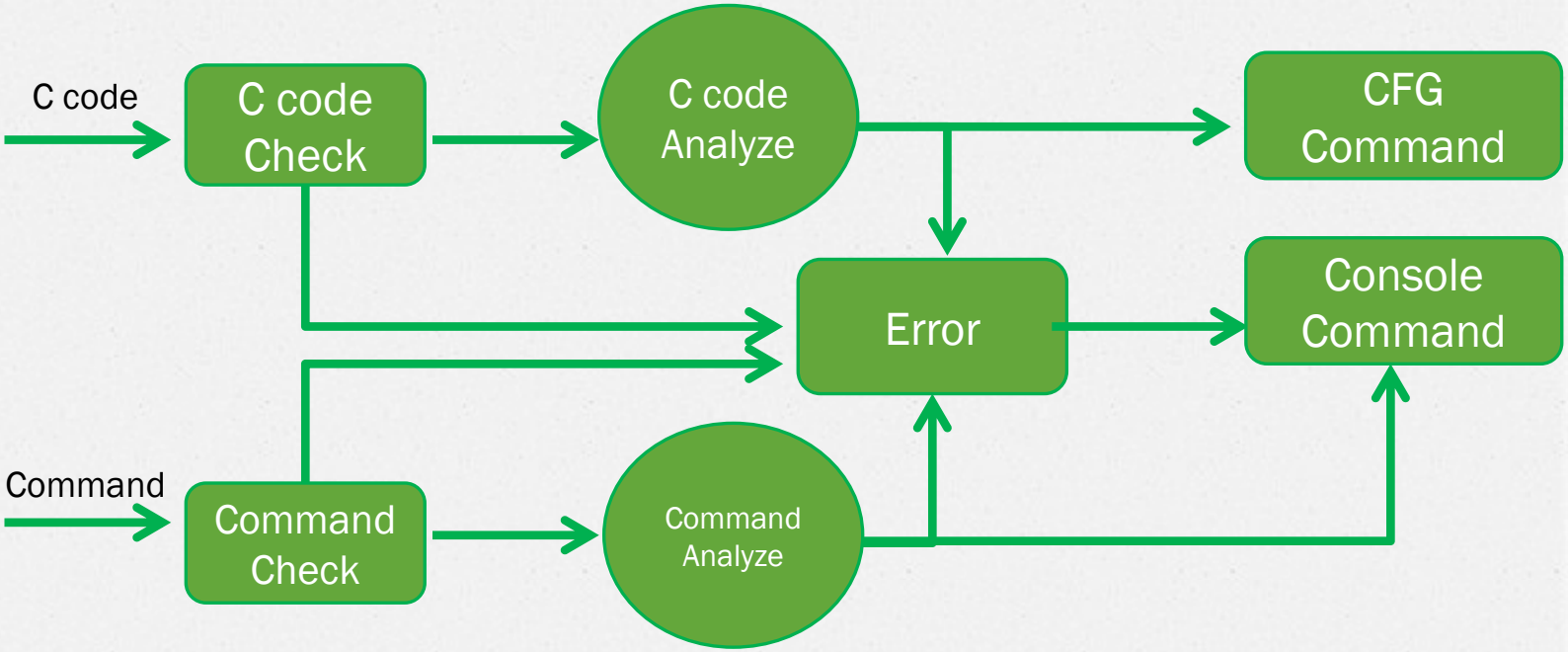
KU 건국대학교 KONKUK UNIV.

FILE IO capture

```
23    fp1 = fopen(finname, "r");
24    if(fp1==NULL)
25    {
26       printf("file %s openfail\n",finname);
27       exit(1);
28    }
29    else
30    {
31       fp2= fopen(foutname,"w");
32       if(fp2==NULL)
33       {
34          printf("file %s openfail\n",foutname);
35          exit(1);
36       }
```

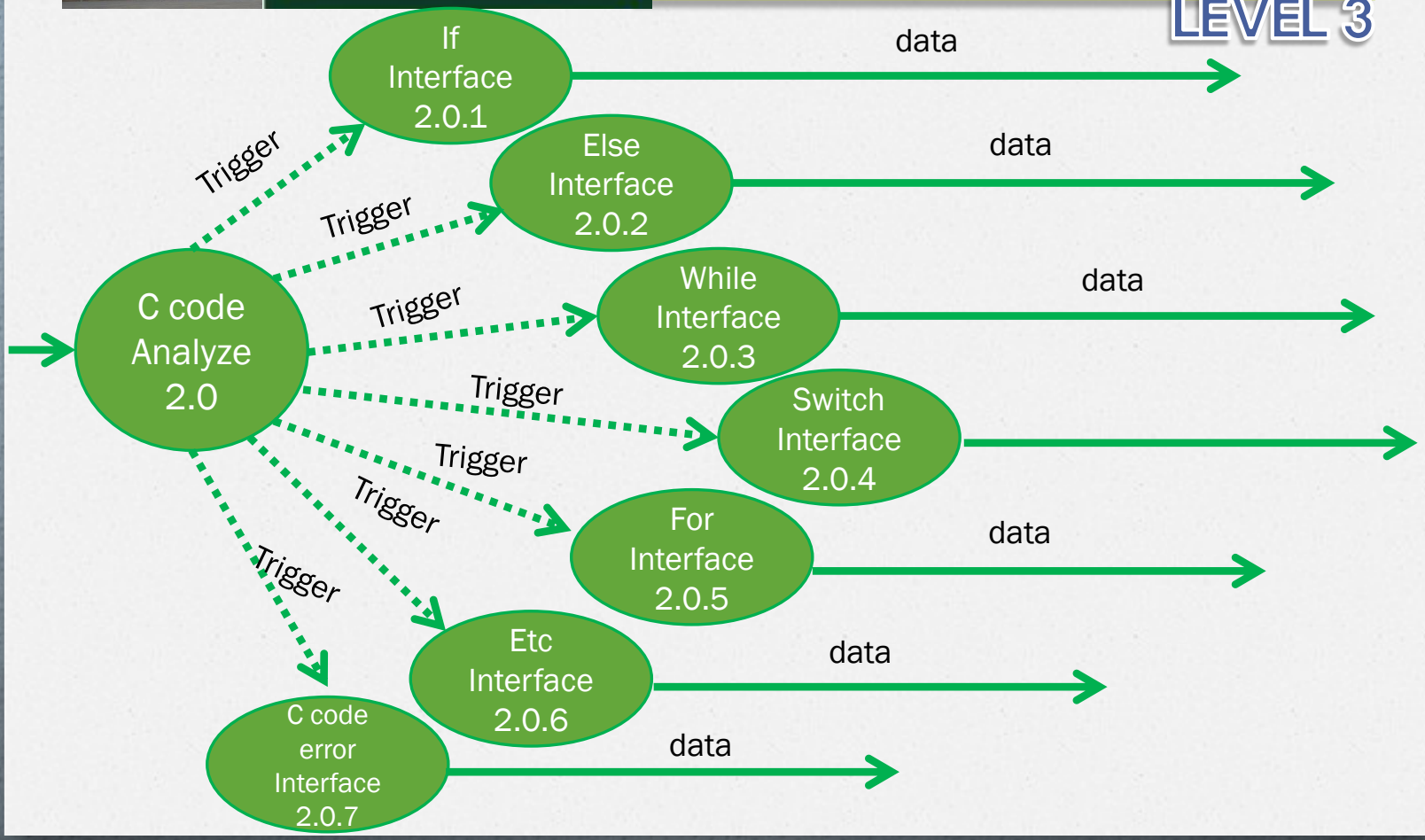
KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM LEVEL 2



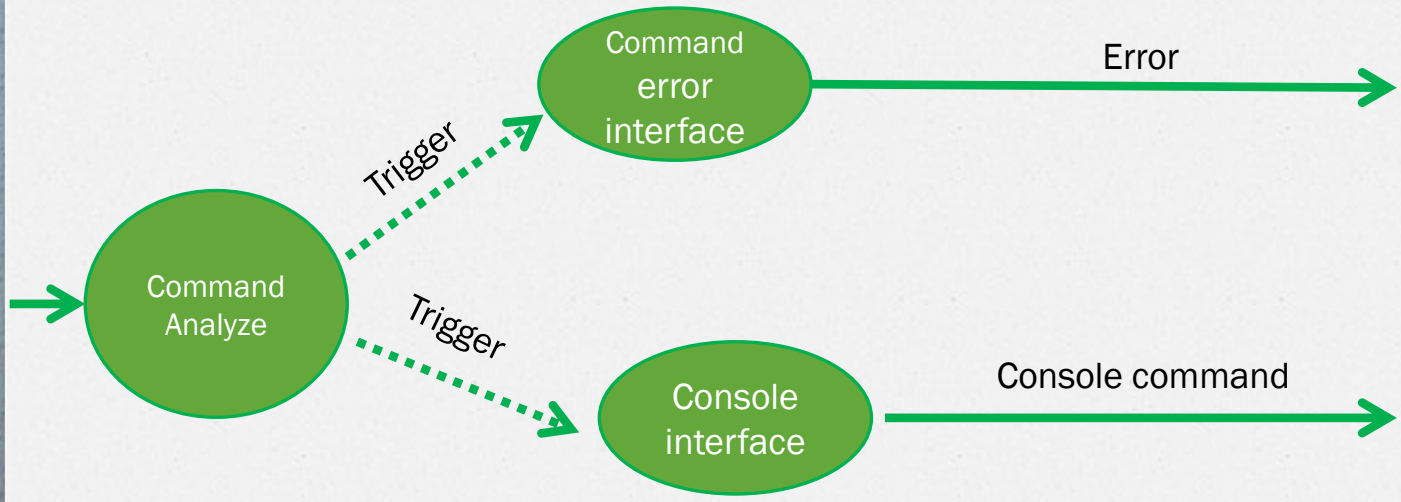
KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM LEVEL 3



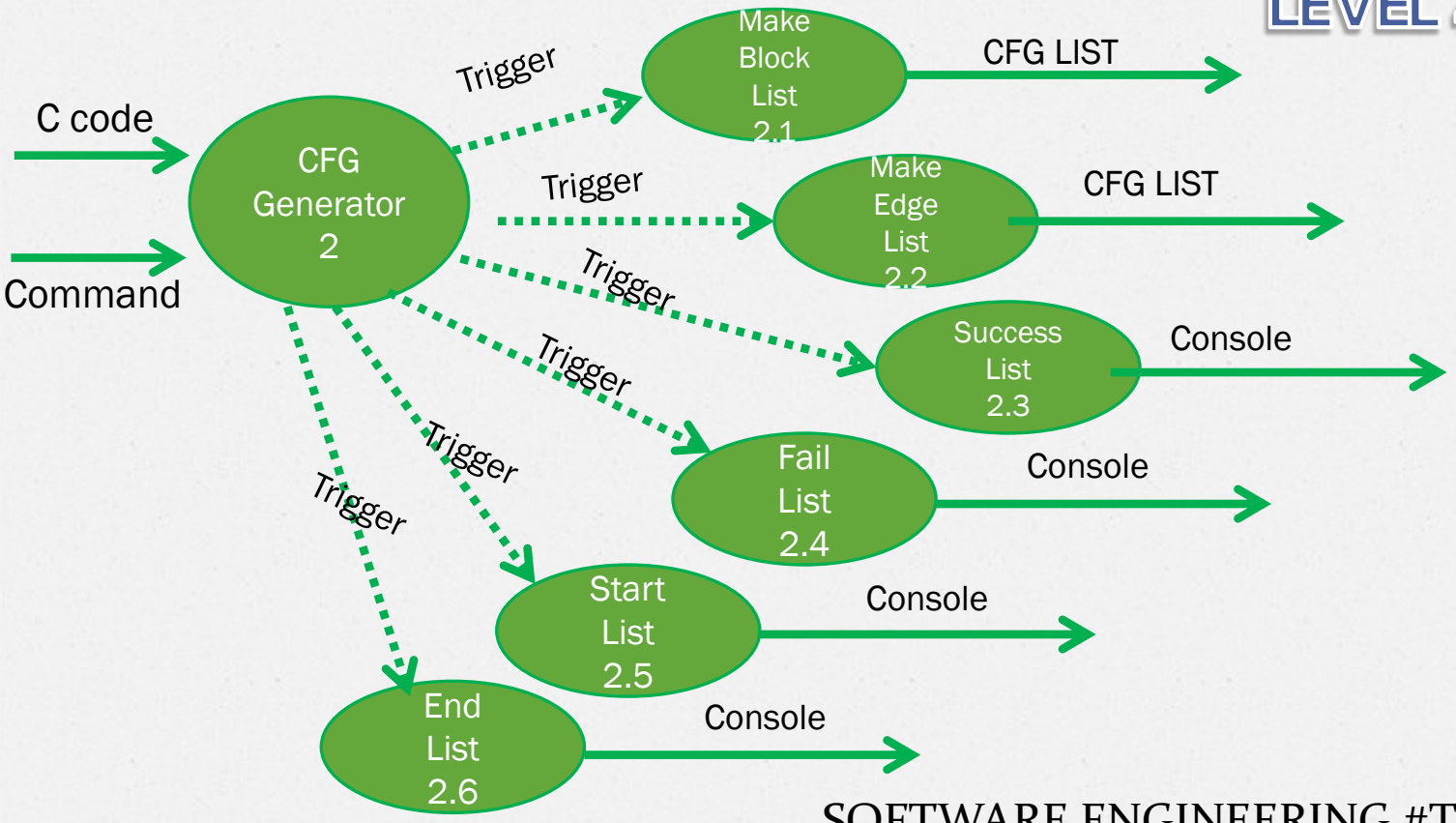
KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM LEVEL 3



KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM LEVEL 4



SOFTWARE ENGINEERING #T8

KU 건국대학교 KONKUK UNIV.

EVENT LIST

Input / output Event	description
C code Input	Check C code
Command Input	Check Command
C code	C code content
Command	Command content
Console	Handle a Command
CFG List	Print command CFG List

KU 건국대학교 KONKUK UNIV.

capture

```
37        else
38        {
39              while(fgets(buffer,256,fp1))
40              {
41                 line_num++;
42                 if(strstr(buffer,word1))
43                 {
44                    printf("*****************\n");
45                    printf(" %d line- block [%s]\n",line_num, word1);
46                    printf("*****************\n");
47                    fprintf(fp2,"/*/");
48                    fprintf(fp2," %d line- block [%s]",line_num, word1, fp2);
49                    fprintf(fp2,"/*/");
50                 }
51                 else if(strstr(buffer,word2))
52                 {
53                    printf("*****************\n");
54                    printf(" %d line- block [%s]\n",line_num, word2);
55                    printf("*****************\n");
56                    fprintf(fp2,"/*/");
57                    fprintf(fp2," %d line- block [%s]",line_num, word1, fp2);
58                    fprintf(fp2,"/*/");
59                 }
60                 else if(strstr(buffer,word3))
61                 {
62                    printf("*****************\n");
63                    printf(" %d line- block [%s]\n",line_num, word3);
64                    printf("*****************\n");
65                    fprintf(fp2,"/*/");
66                    fprintf(fp2," %d line- block[%s]\n",line_num,word3);
67                    fprintf(fp2,"/*/");
68                 }
```

KU 건국대학교 KONKUK UNIV.

capture

```
69                  else if(strstr(buffer,word4))
70                  {
71                     printf("*****************\n");
72                     printf(" %d line- block [%s]\n",line_num, word4);
73                     printf("*****************\n");
74                     fprintf(fp2,"/*/");
75                     fprintf(fp2," %d line- block [%s]\n",line_num,word4);
76                     fprintf(fp2,"/*/");
77                  }
78                  else if(strstr(buffer,word5))
79                  {
80                     printf("*****************\n");
81                     printf(" %d line- block [%s]\n",line_num, word5);
82                     printf("*****************\n");
83                     fprintf(fp2,"/*/");
84                     fprintf(fp2," %d line- block [%s]\n",line_num,word5);
85                     fprintf(fp2,"/*/");
86                  }
87                  else
88                  {
89                     printf(" %d line- edge\n",line_num);
90                     fprintf(fp2,"/");
91                     fprintf(fp2," %d line- edge",line_num);
92                  }
```

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	1.1
Name	C code Interface
Input	C code Input
Output	C code
Description	C code will be sent to the receiving CFG Generator
Reference No.	1.2
Name	Command Interface
Input	Command interface
Output	Command input
Description	Command received will be sent to the CFG Generator

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2
Name	CFG Generator
Input	C code, Command
Output	Trigger
Description	After accepting the basic information and EdgeList BlockList haejum and deliver information to users

Reference No.	2.0
Name	C code Analyze
Input	C code data
Output	Trigger
Description	Appropriate for the received code to execute Command

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.0.1
Name	If interface
Input	trigger
Output	data
Description	If statement is divided into block and Edge

Reference No.	2.0.2
Name	Else interface
Input	trigger
Output	data
Description	Else statement is divided into block and Edge

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.0.3
Name	While interface
Input	trigger
Output	data
Description	while statement is divided into block and Edge
Reference No.	2.0.4
Name	Switch interface
Input	trigger
Output	data
Description	Switch statement is divided into block and Edge

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.0.5
Name	For interface
Input	trigger
Output	data
Description	For statement is divided into block and Edge

Reference No.	2.0.6
Name	Etc interface
Input	trigger
Output	data
Description	Divide the rest of the code block and the Edge

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.0.7
Name	Error interface
Input	Trigger
Output	data
Description	Error is passed

Reference No.	2.1
Name	Make Block List
Input	Trigger
Output	CFG List
Description	Block List is created.

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.2
Name	Make Edge List
Input	trigger
Output	CFG List
Description	Edge List is created.

Reference No.	2.3
Name	Success List
Input	trigger
Output	Console
Description	Input successfully enters, tells the result to the user.

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.4
Name	Fail List
Input	trigger
Output	Console
Description	Input when not successfully enter, tell the result to the user.

Reference No.	2.5
Name	Start List
Input	trigger
Output	Console
Description	Input occurs when successfully entered, the user tells you that to start.

KU 건국대학교 KONKUK UNIV.

PROCESS SPECIFICATION

Reference No.	2.6
Name	End List
Input	trigger
Output	Console
Description	The completion of the CFG output to the user tells you that completion.

Reference No.	3.1
Name	Command
Input	Console
Output	Message Display
Description	Message to show the user.

KU 건국대학교 KONKUK UNIV.

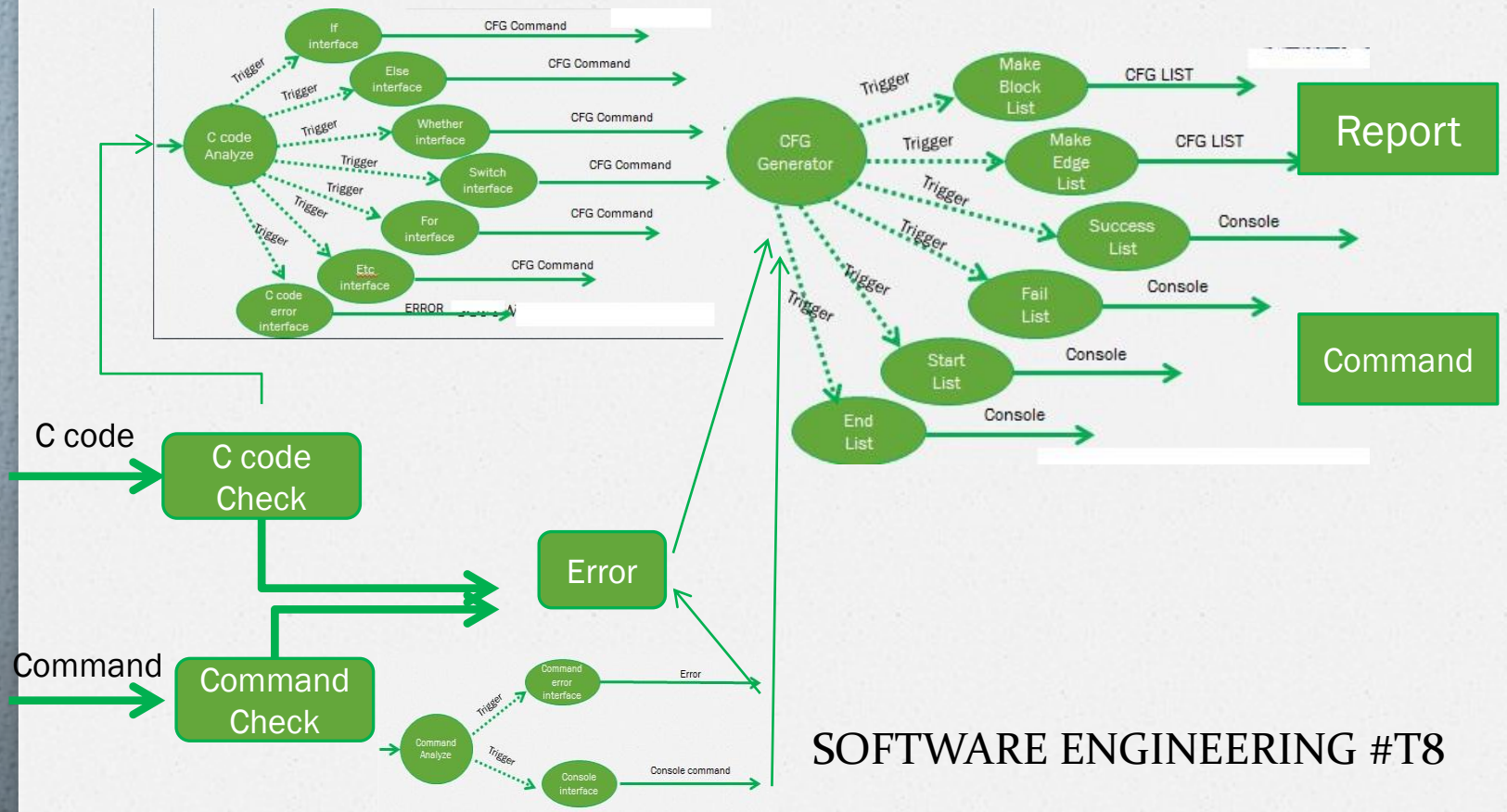
PROCESS SPECIFICATION

Reference No.	3.2
Name	Report
Input	CFG List
Output	FIle
Description	CFG List receives, txt file format stores.

KU 건국대학교 KONKUK UNIV.

DATA FLOW DIAGRAM

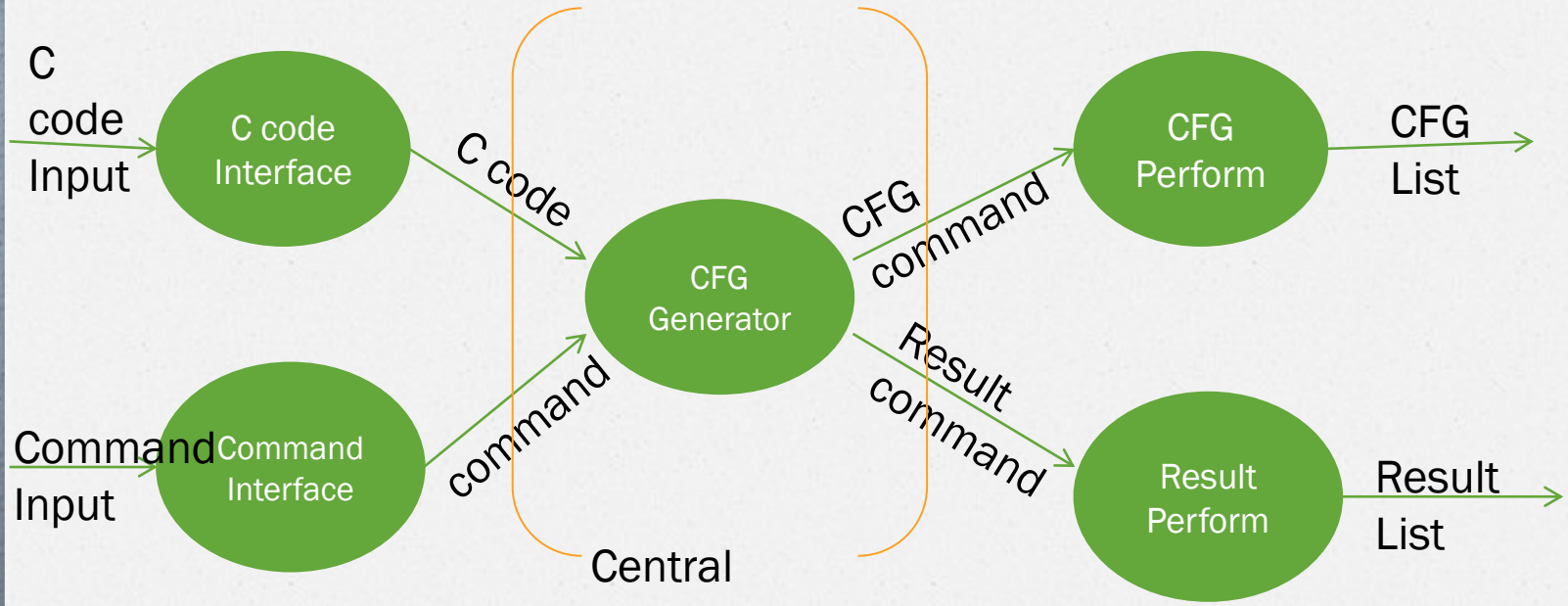
FINAL



SOFTWARE ENGINEERING #T8

KU 건국대학교 KONKUK UNIV.

TRANSFORM ANALYSIS



KU 건국대학교 KONKUK UNIV.

MODULE DEFINITION

Input / output Event	description
C code Interface	Enter the code are built on Cygwin environment.
Command Interface	Enter the command takes.
CFG perform	Create a CFG.
Command perform	To output the results to the user.

KU 건국대학교 KONKUK UNIV.

SD BASIC

Main

Controller

Code

C code Interface

Command Interface

command

Trigger

Trigger

Make block/edge

Trigger

Success/fail /start/end result

Trigger

Trigger

Trigger

Etc Interface

If/else/whether/switch/for Interface

Error Interface

Console Command

CFG Command

CFG Command

CFG Perform

Result Perform

KU 건국대학교 KONKUK UNIV.

SD ADVANCED

Main

Controller

Code

C code Interface

Command Interface

command

Trigger

Trigger

Make block/edge

Trigger

Success/fail /start/end result

Trigger

Trigger

Trigger

Etc Interface

If/else/whether/switch/for Interface

Error Interface

Console Command

CFG Command

CFG Command

CFG Perform

Result Perform

KU 건국대학교 KONKUK UNIV.

DEMO
capture

SOFTWARE ENGINEERING #T8

KU 건국대학교 KONKUK UNIV.

DEMO(test.txt)

test.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
#include<stdio.h>
int main(void)
{
        int a=1;
        printf("hello world");

        if(a==2)
        {
                exit(1);
        }
        else
        {
                printf("wow");
        }
        while(0)
        {
         switch
        }

        return 0;
}
```

KU 건국대학교 KONKUK UNIV.

DEMO(performing)

```
VAIO@VAIO-VAIO ~
$ ./main
write the input file:test.txt
write the output file:cg.txt
 1 line- edge
 2 line- edge
 3 line- edge
 4 line- edge
 5 line- edge
 6 line- edge
*****************
 7 line- block [if]
*****************
 8 line- edge
 9 line- edge
 10 line- edge
*****************
 11 line- block [else]
*****************
 12 line- edge
 13 line- edge
 14 line- edge
*****************
 15 line- block [while]
*****************
 16 line- edge
*****************
 17 line- block [switch]
*****************
 18 line- edge
 19 line- edge
 20 line- edge
 21 line- edge
```

KU 건국대학교 KONKUK UNIV.

DEMO(result.txt)

cg.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
/ 1 line- edge
/ 2 line- edge
/ 3 line- edge
/ 4 line- edge
/ 5 line- edge
/ 6 line- edge
/*
/ 7 line- block
[if]/*/
/ 8 line- edge
/ 9 line- edge
/ 10 line- edge
/*
/ 11 line- block
[if]/*/
/ 12 line- edge
/ 13 line- edge
/ 14 line- edge
/*
/ 15 line- block
[while]□/*/
/ 16 line- edge
/*
/ 17 line- block
[switch]□
18 line- edge
/ 19 line- edge
/ 20 line- edge
/ 21 line- edge
```