CFG Generator SASD

Original By Class A – T1
권선일, 백인선, 이주희, 안혜수
Modified By Class A – T9
문윤주, 이인혁

Structured Analysis

- **Statement of Purpose**
- **System Context Diagram**
  - Event List
- **Data Flow Diagram**
  - Data Dictionary
  - Process Specification
  - Total DFD

Structured Design

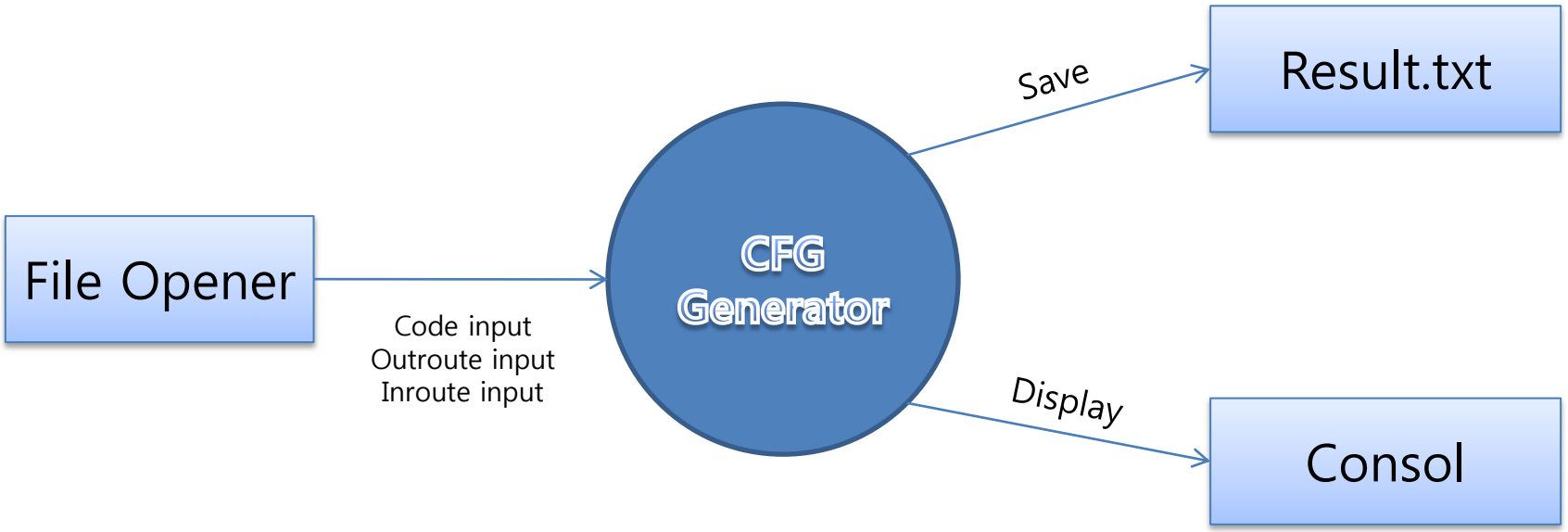
- **Structure Charts**
  - Transform Analysis
  - Basic
  - Advanced

SA Part.

Statement of Purpose

- Convert a C source code to CFG.
- This program converts only Main() Function parts.
- It is a single-file that doesn't have user defined header files and doesn't include pointers.
- The C source code has 100~200 lines which includes main function.
- The source code can not detect logical errors.
- When C source code inputted successfully, the program shows "Success" message. Or in error case, the program shows "Fail" and terminates the program.
- Report with a text(*.txt) file.
- The report show all blocks and edges of CFG.

System Context Diagram

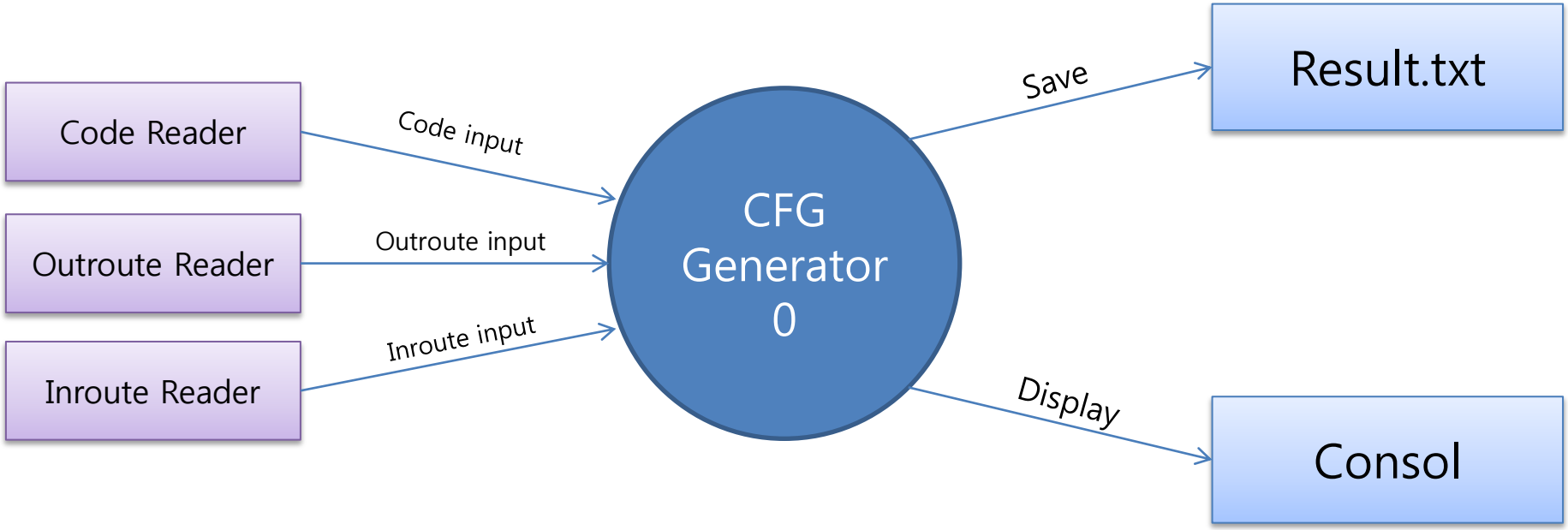


Event List

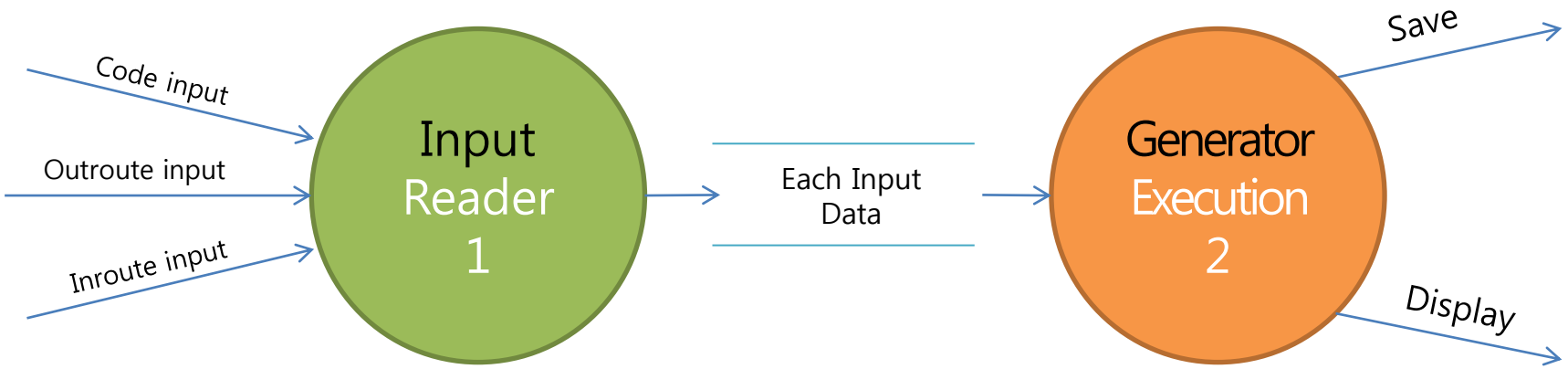
Input Event	Description
Code input	Converted to CFG .C Source Code of the file
Outroute input	The path to the file to be saved.
Inroute input	C Code the path of a file for reading.

Output Event	Description
Save	CUI forms stored on the path to save the converted CFG
Display	Processing the user to see what progress the resulting output to the consol screen.

Data Flow Diagram – Level 0.



Data Flow Diagram – Level 1.

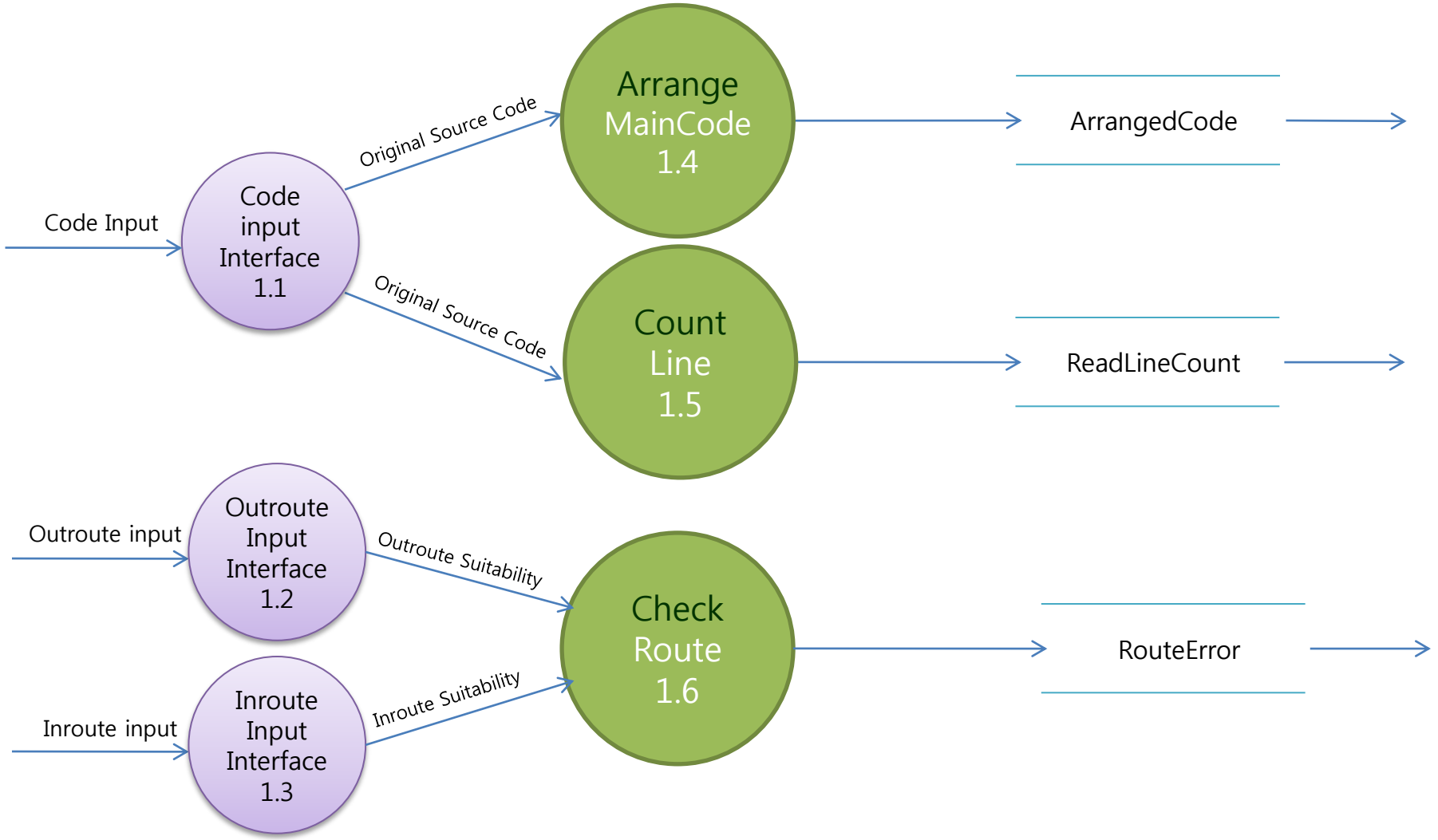


Data Dictionary (DFD Level 1)

Data 명	설 명	형식
Code Input	C Code written to be used for the conversion of the Source File.	File
Outroute Input	The path to the file to be saved.	String
Inroute Input	C Code the path of a file for reading.	String
Each Input Data	Reader Interface exported by each of the structures. ArrangedCode, ReadLine Count, Route Error has.	Data Structure
Save	Generator to create blocks and edges to send such information to the Saver Interface and Output to Result.txt	Integer/String
Display	Generator to create a situation or CFG shown by the success / failure to send such information to the Displayer Interface and Consol Output of the result value.	String

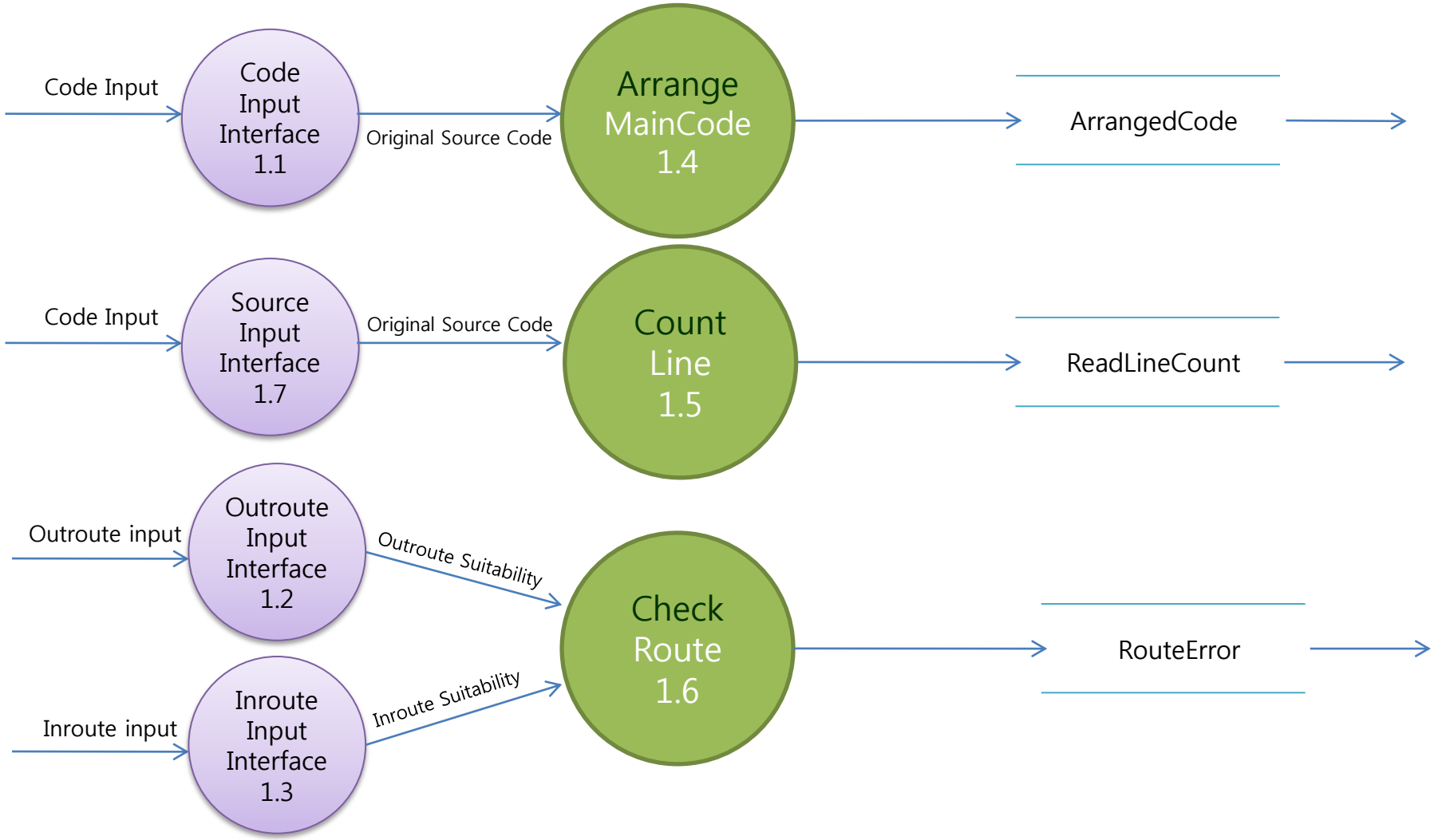
Data Flow Diagram – Level 2.

Original



Data Flow Diagram – Level 2.

Modified



Source Code – DFD Level 2.

```
int Arrange_MainCode(FILE *fp,Block *n)
{
        int a,b,number;
        char *str="main(";

        if(fp==NULL)
        {
                return 1;
        }
        while(!feof(fp))
        {
                fgets(n[idx].s, sizeof(n[idx].s), fp);
                n[idx].LineNum=idx+1;
                idx++;
        }
        do{
                for(b=0;b<idx-1;b++)
                        n[b]=n[b+1];
                idx--;
        }while(!strstr(n[0].s,str));
        for(b=0;b<idx-1;b++)
                n[b]=n[b+1];
        idx--;
        return 0;
}
```

```
int CheckRoute(char *buf,int fileCh)
{
        int i=0,j;
        int avail;
        char *ch;
        ch=(char *)malloc(sizeof(char)*4);
        while(buf[i]!='.'){
                i++;
        }
        ch[0]=buf[i+1];
        ch[1]=buf[i+2];
        ch[2]=buf[i+3];
        ch[3]='\0';

        avail=buf[i+4];
        if(!strcmp(ch, "txt") && fileCh==0 && avail==0)
                return 1;
        else if(fileCh==1)
        {
                return 2;
        }
        else
                return 3;
}
```

Process Specification

Name	1.1 Code Input Interface
Input	Code Input
Output	Original Source Code
Process Description	Entered by the user that convert the "Code Input" after conversion to a digital signal is passed.

Name	1.2 Outroute Input Interface
Input	Outroute Input
Output	Outroute Suitability
Process Description	File for output if the path entered is a valid path to True / False to convert the digital signal is passed.

Name	1.3 Inroute Input Interface
Input	Inroute Input
Output	Inroute Suitability
Process Description	File for input if the path entered is a valid path to True / False to convert the digital signal is passed.

Name	1.4 Arrange MainCode
Input	Original Sourc Code
Output	Arranged Code
Process Description	Original Source Code of the main () function of the first part of the code in the Source Code, except for variable declarations, and Arranged Code to extract the exported.

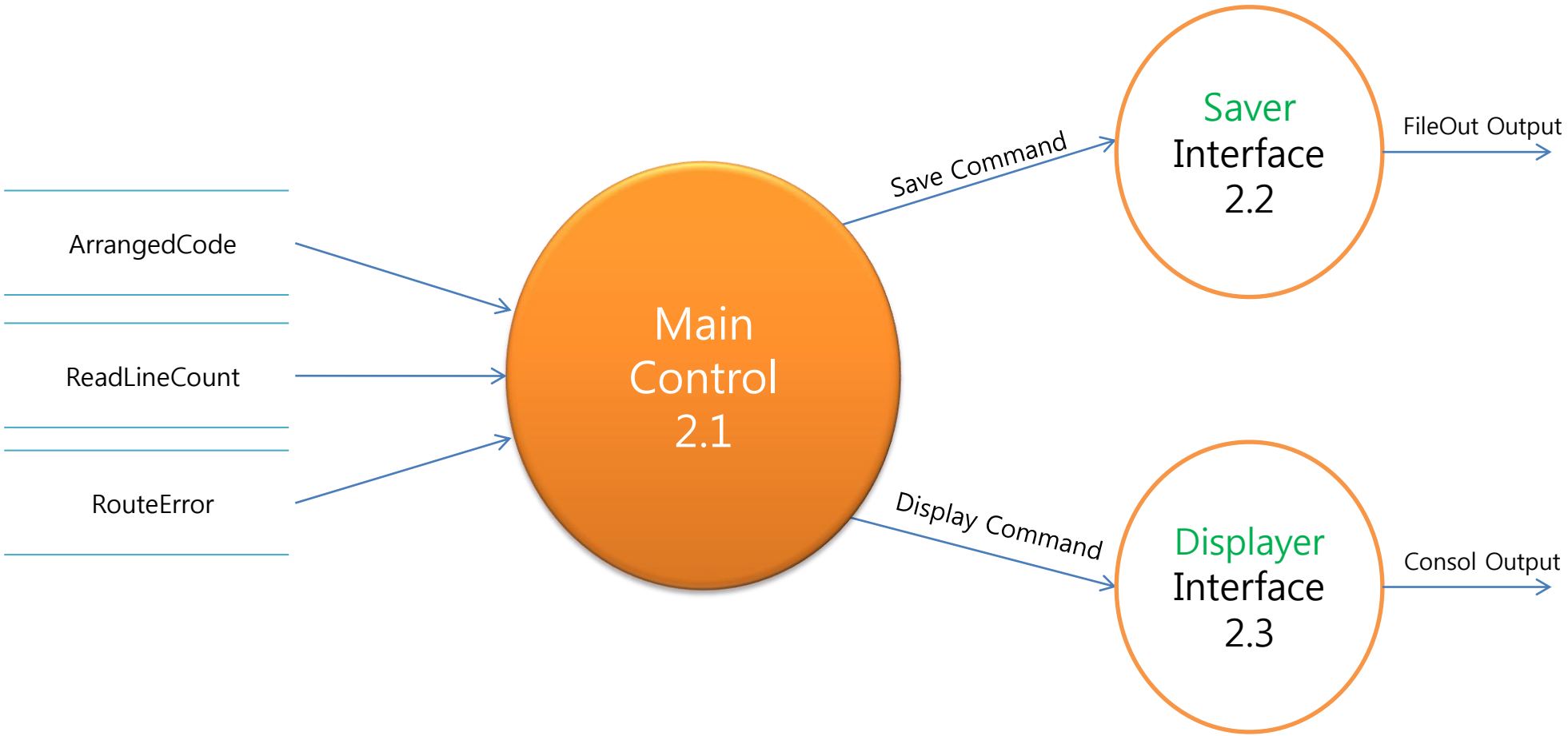
Process Specification (cont.)

Name	1.5 Count Line
Input	Original Source Code
Output	ReadLineCount
Process Description	';' To determine the Line as a unit within Main () '; 'and are counted. Printf function () in the '; 'are excluded from the count.

Name	1.6 Check Route
Input	Outroute Input, Inroute Input
Output	RouteError
Process Description	Enter the Run Command received In, Out File path to validate. If false, input any one of the two prints out false.

Name	1.7 Source Input Interface
Input	Code Input
Output	Original Source Code
Process Description	Entered by the user to convert the "Code Input" after conversion to a digital signal is passed. Interface to divide and ReadLineCount ArrangedCode the allocation of two said.

Data Flow Diagram – Level 2.



Process Specification (cont.)

Name	2.1 Main Control
Input	ArrangedCode, ReadLineCount, RouteError
Output	Save Command, Display Command
Process Description	Data for each to be delivered to judge RoutError and ReadLineCount, CFG's Block by creating a structure to the list, with respect to the final save Save Command and Display Command for outputting the result of the conversion is calculated.

Name	2.2 Saver Interface
Input	Save Command
Output	FileOut Output
Process Description	Save Command Result.txt who enter through the final output to override any TotalCFG Data (CUI).

Name	2.3 Displayer Interface
Input	Display Command
Output	Consol Output
Process Description	Display Command who enter through the Consol window outputs the result of the conversion will. Converted on success and a success message Outfile path, a failure message on failure, and prints the cause of the error information.

Data Dictionary (DFD Level 2)

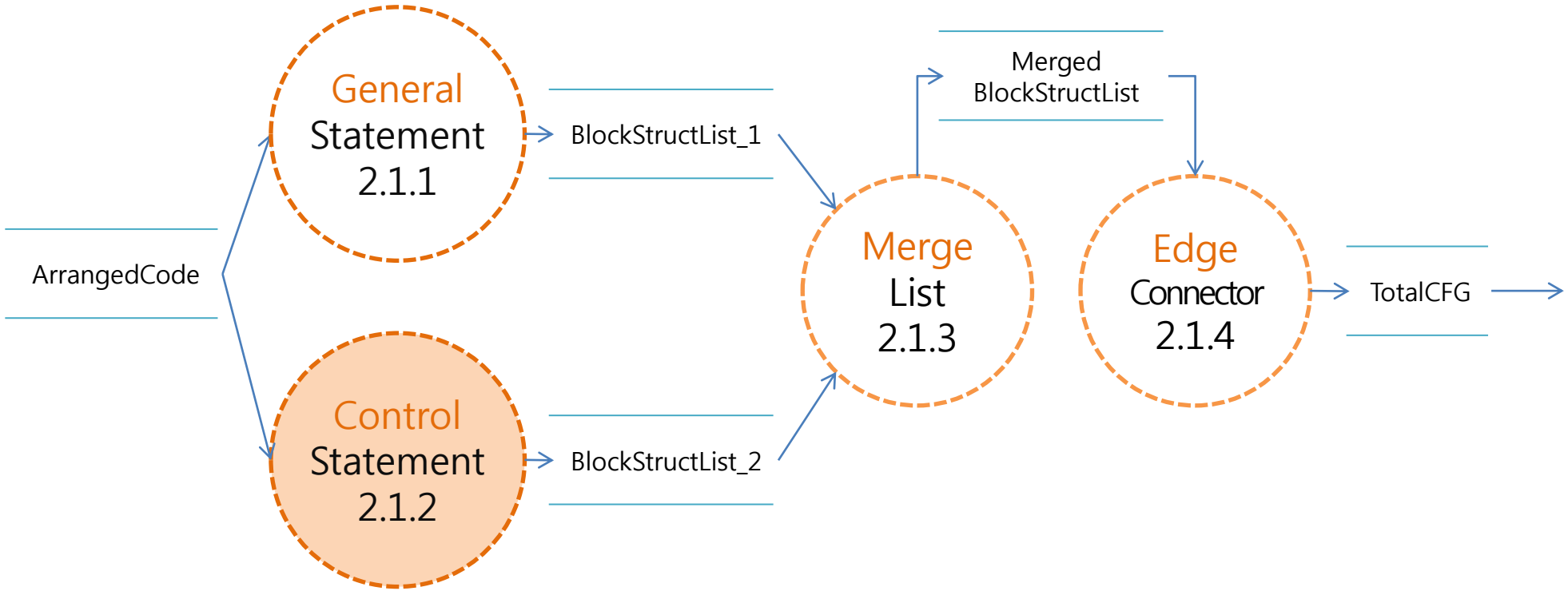
Data 명	설 명	형식
Code Input	C Code written to be used for the conversion of the Source File.	File
Outroute Input	The path to the file to be saved.	String
Inroute Input	C Code the path of a file for reading.	String
Original Source Code	C Code Source itself represents the input received.	String
ArrangedCode	Source code in the main () CFG can be converted in part identified as part of Data.	File
ReadLineCount	Read the number of lines from Main code.	Integer

Data Dictionary (DFD Level 2) (Cont.)

Data 명	설 명	형식
RouteError	File I / O required for the path of the file determines the suitability Data.	True/False
Save Command	Generated by the Generator Block and Edge sends information about the Saver Interface.	Integer/String
Display Command	Generator to create a situation or CFG shown by the success / failure, and sends the information Displayer Interface.	String
FileOut Output	Information came from Save Command in Saver Interface Result.txt Gather information about Total CFG.	File
Consol Output	Displayer Interface to console from Display Command Message.	String

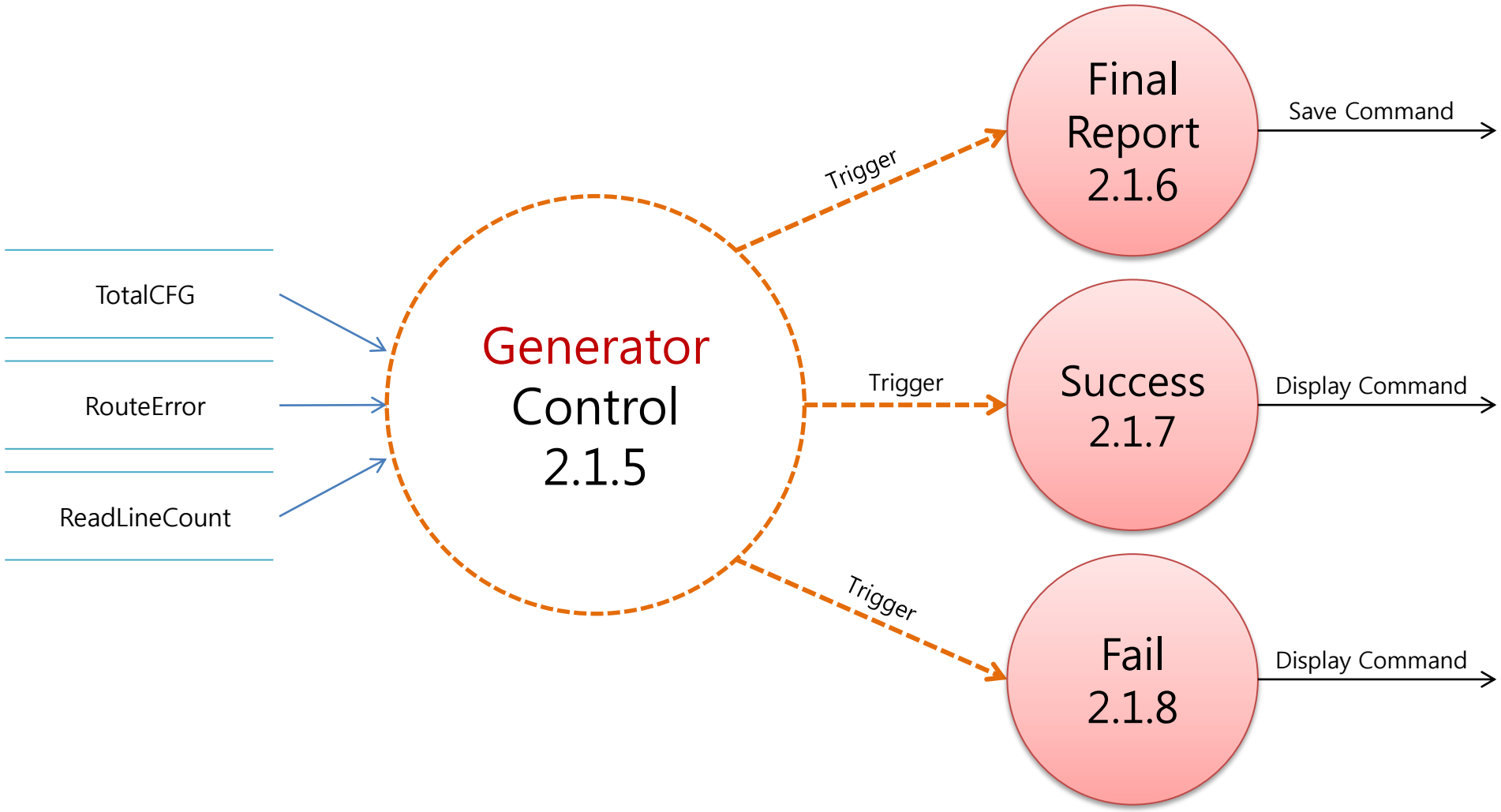
Data Flow Diagram – Level 3.

Original



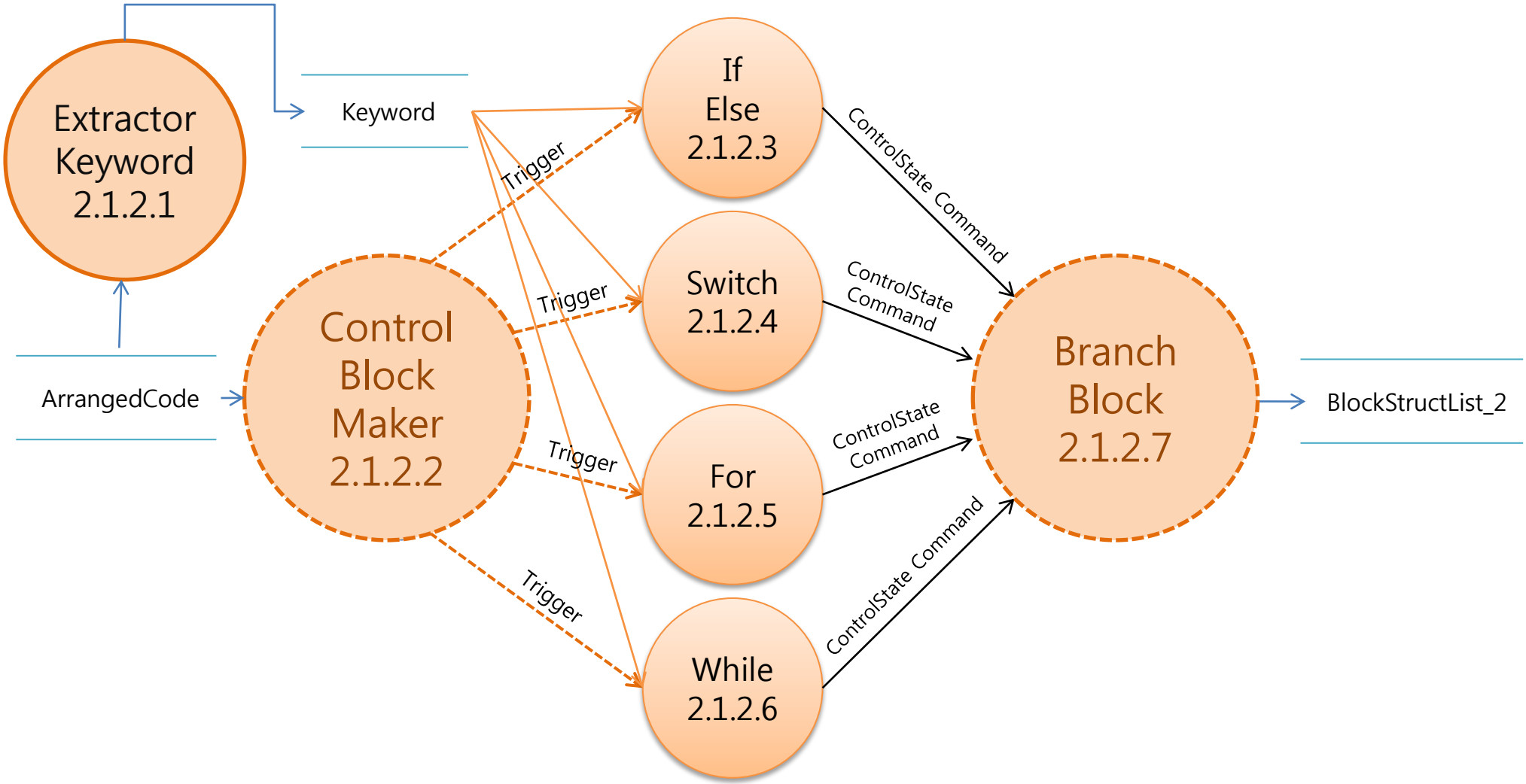
Data Flow Diagram – Level 3.

Original



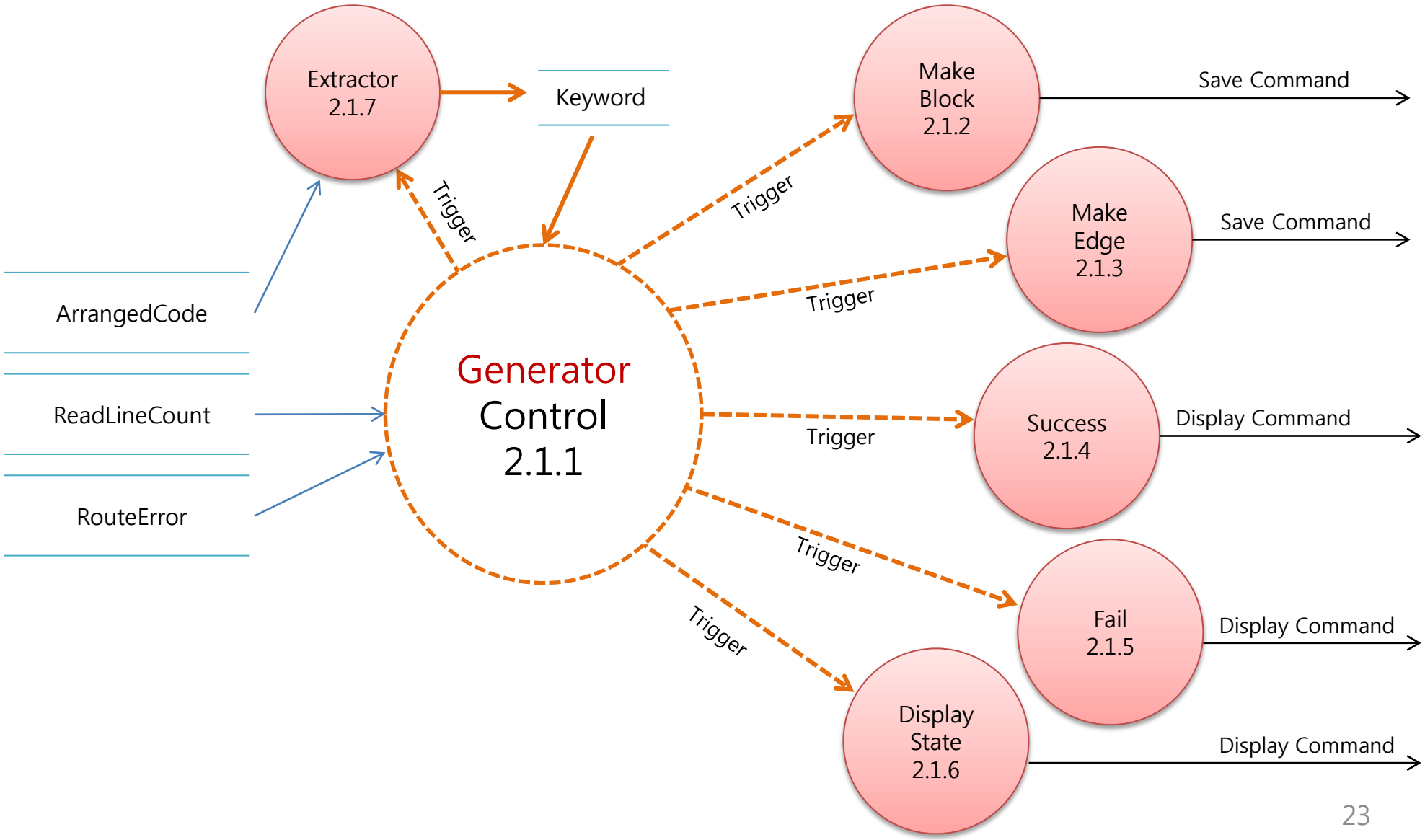
Data Flow Diagram – Level 4.

Original



Data Flow Diagram – Level 3.

Modified



Source Code – DFD Level 3.

```
void Extractor(Block *n,Edge *e,int start)
{
        char prev[100] = {0,};
        char pres[100] = {0,};
        char fut[100] = {0,};
        char Blockcase[100][100];
        int i=0,l=0,j,on=0;
        int number,returnLine;
```

```
else if(strstr(n[i].s,"for("))
{
        sprintf(prev,"%d",n[i].LineNum);
        i++;
        while(!strstr(n[i].s,"}")){
                sprintf(pres,"%d",n[i-1].LineNum);
                sprintf(fut,"%d",n[i].LineNum);
                strcat(pres,"----->");
                strcat(pres,fut);
                strcpy(e[edgenum].s,pres);
                edgenum++;
                i++;
        }
        strcpy(pres,prev);
        sprintf(fut,"%d",n[i-1].LineNum);
        strcat(fut,">>Back----->");
        strcat(fut,pres);
        strcpy(e[edgenum].s,fut);
        edgenum++;
        sprintf(prev,"%d",n[i-1].LineNum);
        sprintf(pres,"%d",n[i+1].LineNum);
        strcat(prev,"----->");
        strcat(prev,pres);
        strcpy(e[edgenum].s,prev);
        edgenum++;
}
```

```
void DisplayerInterface(char *ment)
{
        printf("%s\n",ment);
}
```

Source Code – DFD Level 3. (cont.)

```
void Generator(char **argv,int argc)
{
        FILE *fp=NULL;
        FILE *ofp=NULL;
        Block n[200];
        Edge e[200];
        int checker=0;
        int RouteError,i;
        char buf[100]={0,};
```

```
if(RouteError==1)
{
        Success();
        Save_Start();

        Extractor(n,e,0);
}
else if(RouteError==2)
{
        Fail_Inroute();
        exit(0);
}
else
{
        Fail_Outroute();
        exit(0);
}
for(i=0;i<idx;i++){
        printf("Block[%d] = %s",i+1,n[i].s);
        printf("_____\n");
}
for(i=0; i<edgenum; i++){
        printf("edge[%d] = %s\n",i+1,e[i].s);
}
for(i=0; i<idx; i++)
{
        strtok(n[i].s,"\n");
}
SaverInterface(n,e,ofp,argv[2]);
```

```
void Save_Start(void)
{
        char *ment="Start Convert !!";
        DisplayerInterface(ment);
}
void Success(void)
{
        char *ment="Success Convert !!";
        DisplayerInterface(ment);
}
void Fail_Inroute(void)
{
        char *ment="Inroute Input Error.";
        DisplayerInterface(ment);
}
void Fail_Outroute(void)
{
        char *ment="Outroute Input Error.";
        DisplayerInterface(ment);
}
```

Data Dictionary (DFD Level 3)

Data 명	설 명	형식
ArrangedCode	Source code in the main () CFG can be converted in part identified as part of Data.	File
ReadLineCount	Read the number of lines from Main code.	Integer
RouteError	File I / O required for the path of the file determines the suitability Data.	True/False

Keyword	Through the Block and Edge Extractor for Data is converted.
	
	If you believe in Block Data with Type 1 information on Block Number, Block Type, Start Line Number, End Line Number has.
	
	If you believe in Edge Data taken with Type 2 Information Edge Number, Edge Type, Source Block Number, Destination Block Number has.
		Struct
		{
		int/double,
		Int,
		String
		}
		List[]

Process Specification (cont.)

Name	2.1.1 General Control
Input	ArrangedCode, RouteError, ReadLineCount
Output	Trigger
Process Description	ArrangedCode, RouteError, ReadLineCount Data Extractor converts those inputs and sends the Trigger to start working. Depending on the value received from the Keyword Extractor for Block or Make Block or Make Edge to Edge Information as to Trigger Output.

Name	2.1.2 Make Block
Input	Trigger
Output	Save Command
Process Description	Trigger receives information from the Main Control Block information to generate and send Saver Interface.

Name	2.1.3 Make Edge
Input	Trigger
Output	Save Command
Process Description	Trigger receives information from the Main Control Edge information to generate and send Saver Interface.

Process Specification (cont.)

Name	2.1.4 Success
Input	Trigger
Output	Display Command
Process Description	Received from the Main Control Trigger for the success of the operation sends information Displayer Interface.

Name	2.1.5 Fail
Input	Trigger
Output	Display Command
Process Description	Received from the Main Control Trigger Success/failure of the operation sends information about the Displayer Interface.

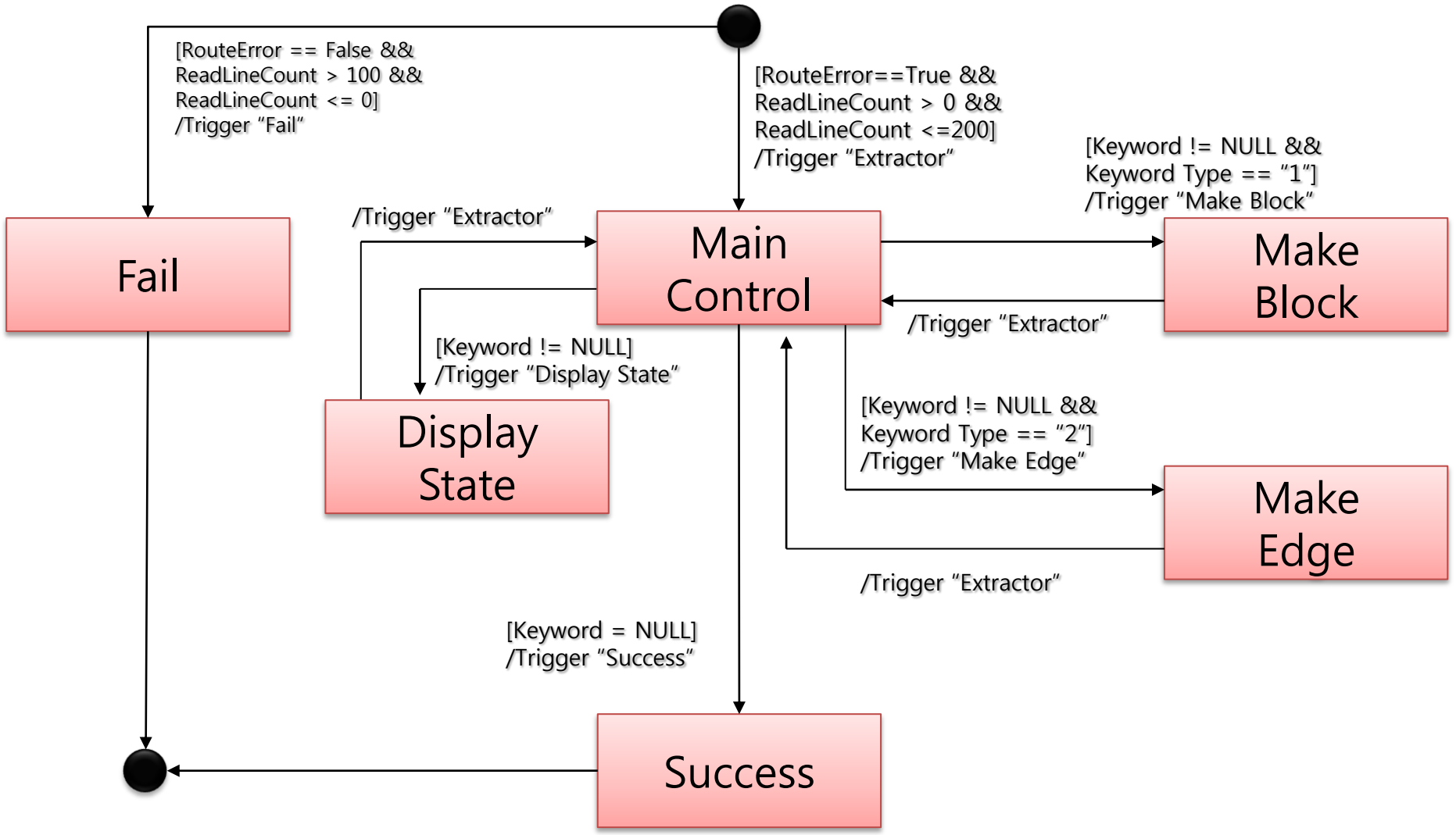
Name	2.1.6 Display State
Input	Trigger
Output	Display Command
Process Description	Received from the Main Control Trigger was recently working on the State for the information and sends it to Displayer Interface.

Name	2.1.7 Extractor
Input	ArrangedCode, Trigger
Output	Keyword
Process Description	Main Control receives a Trigger ArrangedCode Keyword information for them line by line in the Data Structure for CFG put back pass in the Main Control. This information is based on the Main Control Make Block or Make Edge sends information to the Output Trigger.

Data Flow Diagram – Level 4.

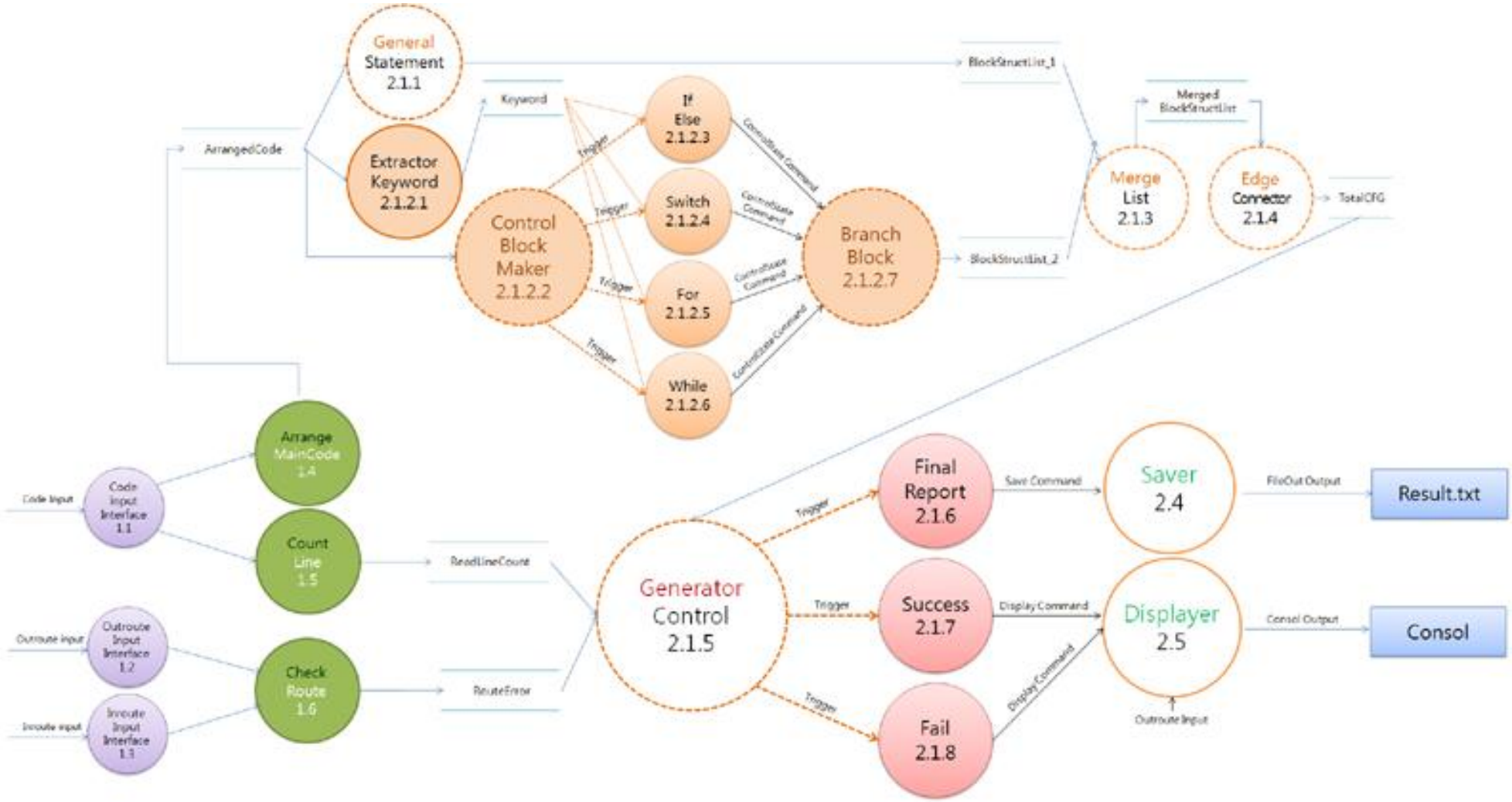
- State Transition Diagram for Controller 2.1.1

Modified



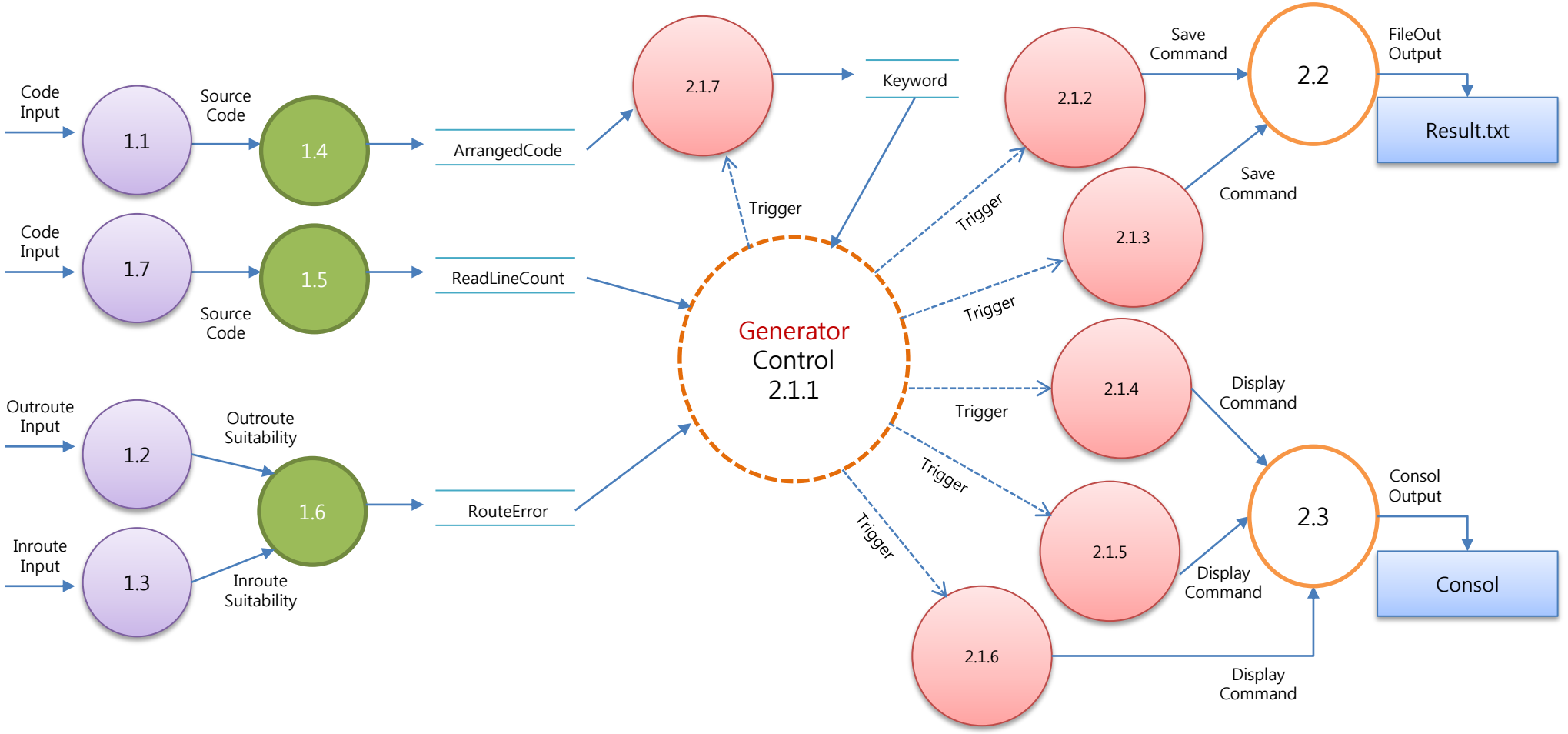
Total DFD

Original



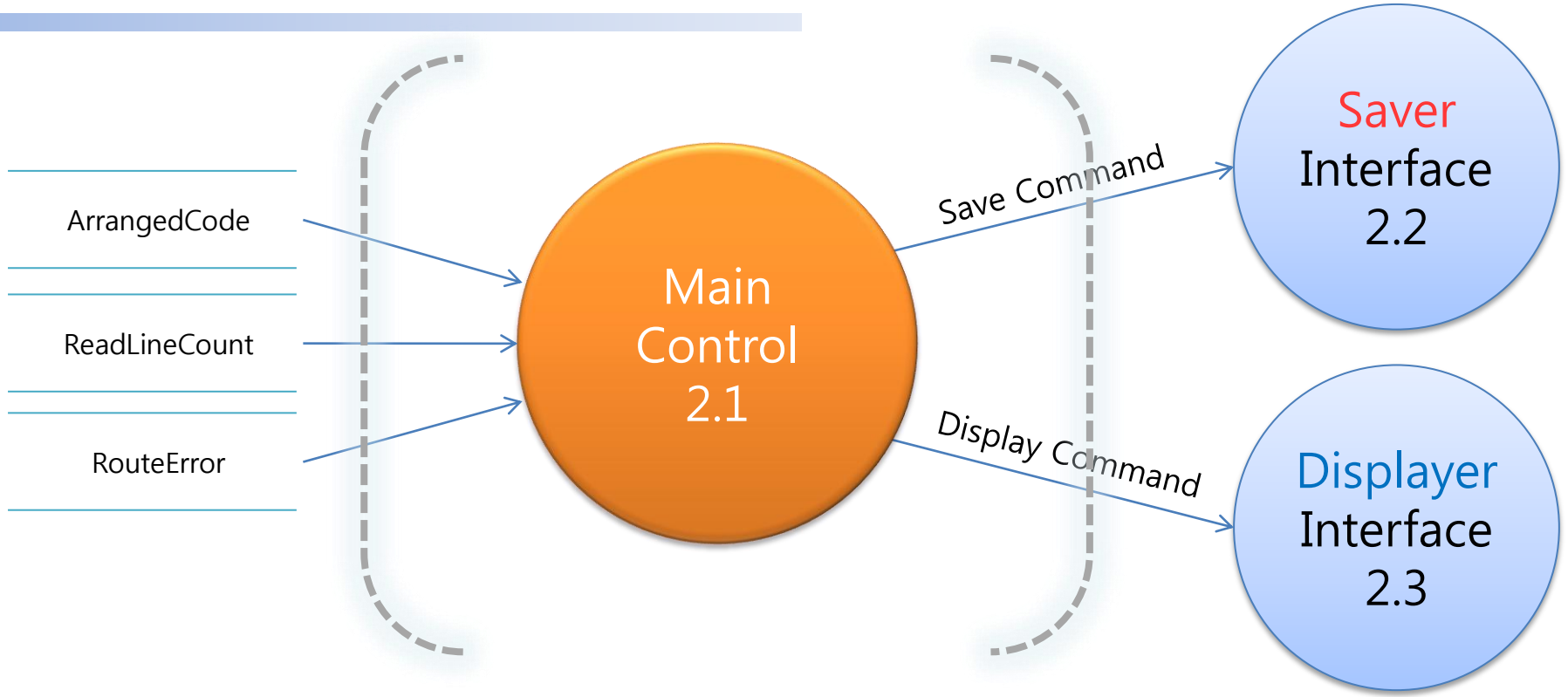
Total DFD

Modified



SD Part.

Transform Anlysis

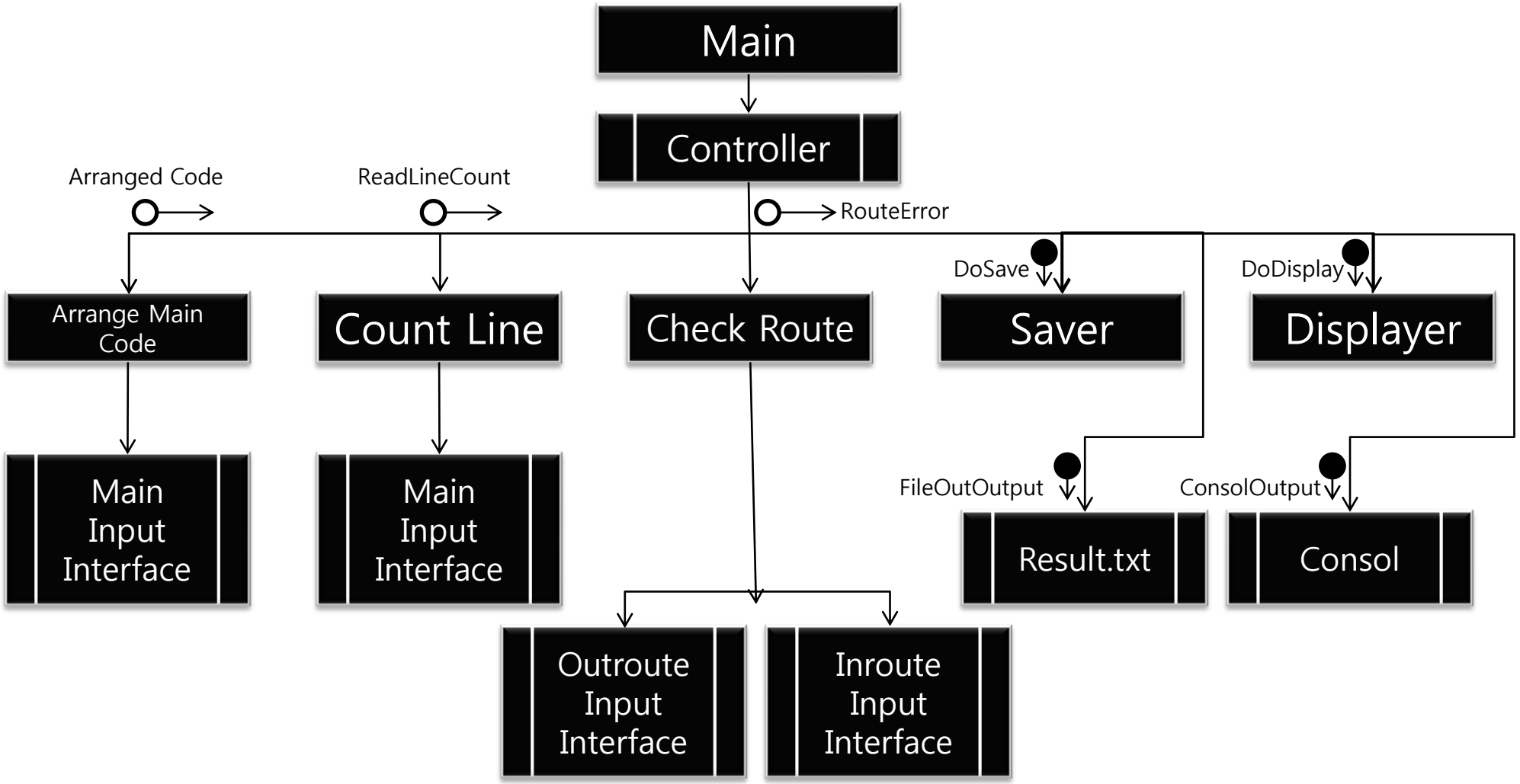


Afferent Flow (Input)

Central Transformation (Control)

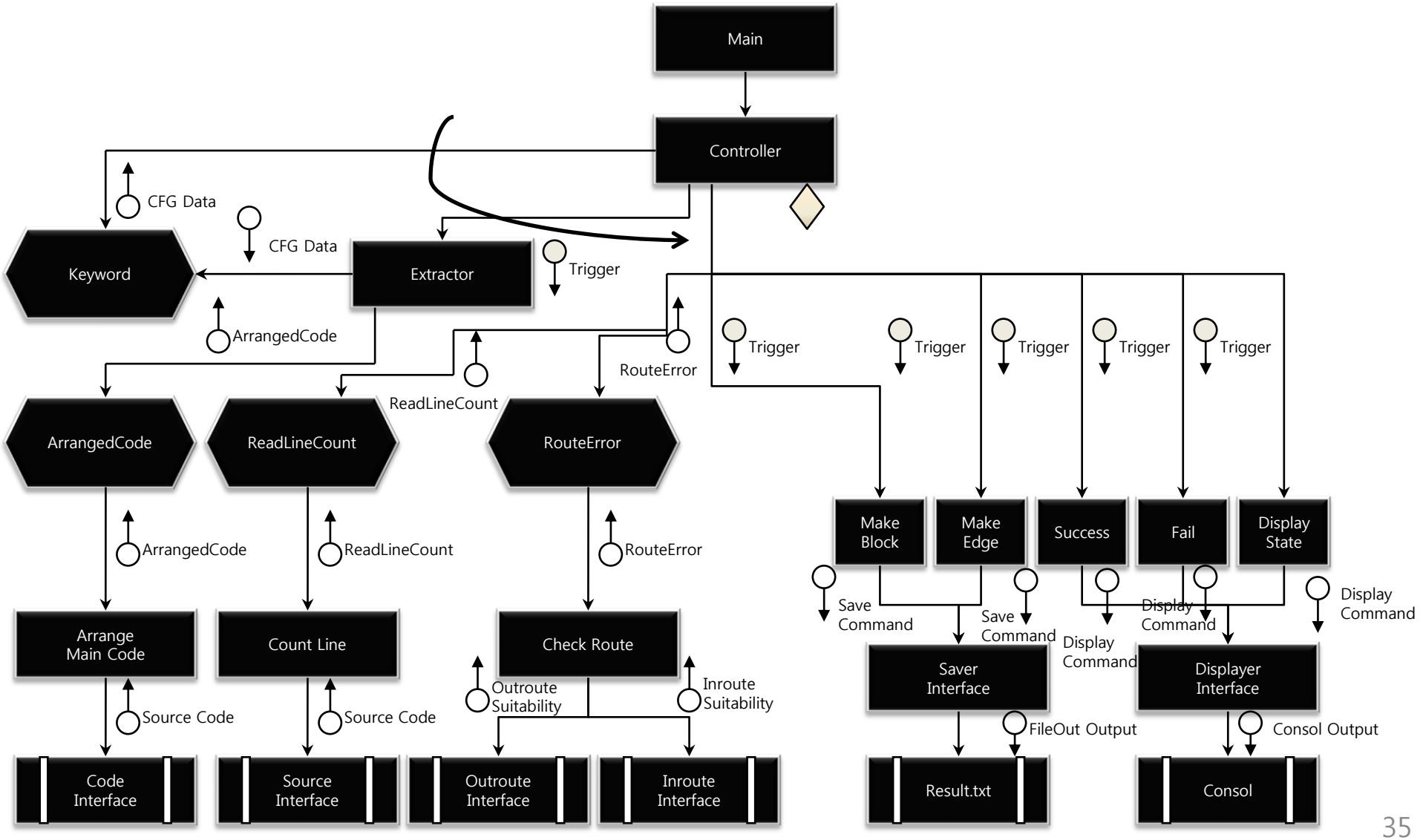
Efferent Flow (Output)

Basic


Main
Controller
Arranged Code
ReadLineCount
RouteError
DoSave
DoDisplay
Arrange Main Code
Count Line
Check Route
Saver
Displayer
Main Input Interface
Main Input Interface
FileOutOutput
ConsolOutput
Result.txt
Consol
Outroute Input Interface
Inroute Input Interface

Advanced

Modified



Demonstration.

The End

감 사 합 니 다 !