

SASD of CFG Generator

Team 6

발표 : 200811425 김평석
200811435 신성호
200811451 이형열
200811454 전인서

Structured Analysis

Contents of SA

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

Statement of Purpose

- Entire Statement of Purpose
- Specific Statement of Purpose

Entire Statement of Purpose

Entire Statement of Purpose

Program은 CUI(Command User Interface)기반의 Cygwin에서 구동된다.

C language Program을 입력 받아 CFG(Control Flow Graph)의 State목록과 Edge의 목록을 List형태로 출력한다.

C program의 유효성을 검사한 후 중간언어를 생성한다.

생성된 중간언어를 통하여 CFG정보를 도출한다.

도출된 CFG정보는 사용자 옵션에 의하여 File 과 Console에 각각 출력된다.

System Condition에 따라 Message를 Console에 출력한다.

Specific Statement of Purpose

Input Statement of Purpose

Command Line 명령어 형태로 Source File Name, Output File Name과 Output Option을 입력 받는다.

입력되는 C Program File은 200줄 이내의 단일 파일이며 *.c의 확장자를 가져야 한다.

Main function을 반드시 포함하는 Code이어야 한다.

Pointer를 사용하지 않는 Code를 대상으로 한다.

사용자가 정의한 헤더와 Library를 사용한 파일에 대해서는 작동하지 않는다.

입력 받는 C Code파일은 문법적 오류가 없는 파일로 한정한다.

Specific Statement of Purpose

Parsing Statement of Purpose

입력 받은 C Code를 지정된 토큰을 기준으로 나눈다.

나뉜 부분의 동작을 파악하여 중간언어로 변환한다.

Macro에서 생성되는 분기는 고려하지 않는다.

삼항연산자($x ? y : z$)는 분기를 생성하지 않는다.

Library와 Function안의 분기는 CFG에 나타내지 않는다.

Specific Statement of Purpose

Constructing CFG Statement of Purpose

중간언어를 분석하여 CFG의 Block List를 생성한다.

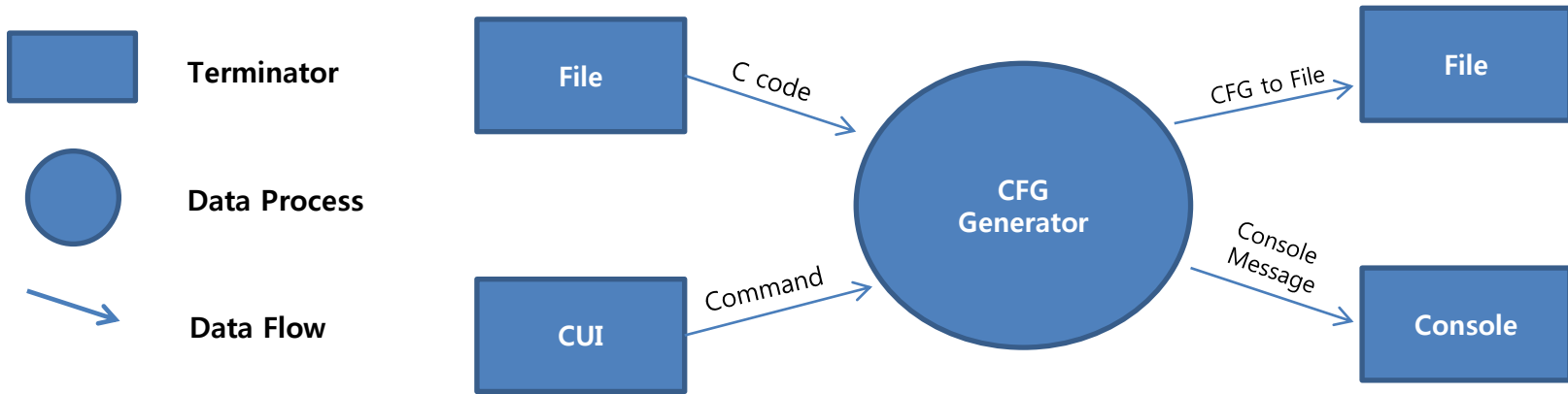
생성된 Block List를 이용하여 Edge List를 생성한다.

Output Statement of Purpose

Command Error 또는 File Error가 발생하면 Console에 Error Message와 Help Message를 출력한다.

출력옵션에 따라서 CFG component List를 이용하여 File과 Console에 각각 구조적으로 출력한다.

System Context Diagram & Event List



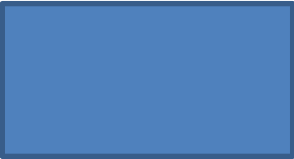
Input/Output Event	Description	Format/Type
Command	Input File Name, Output File Name 그리고 Output Option을 포함한 문자열 Ex) ./cfg [option] [input file name] [output file name]	string
C Code	CFG로 변환될 C code로 *.c의 확장자 명을 가지는 파일. C 표준에 의해 생성된 소스코드이다.	*.c
CFG to File	File에 출력될 CFG로서 string형식으로 파일에 쓰여지게 된다.	string
Console Message	콘솔에 출력할 메시지들과 CFG로 각 상태에 맞는 메시지들을 콘솔에 출력하도록 해준다.	CmdError / FileError / HelpMessage / GenerateMessage / CFG / OutFileName

Data Flow Diagram

Data Dictionary

Process Specification

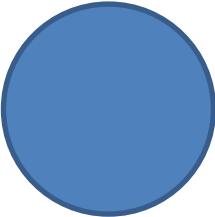
Notation of Data Flow Diagram



Terminator



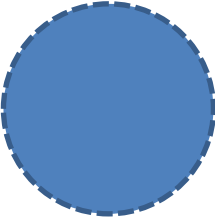
Data Store



Data Process



Data Flow

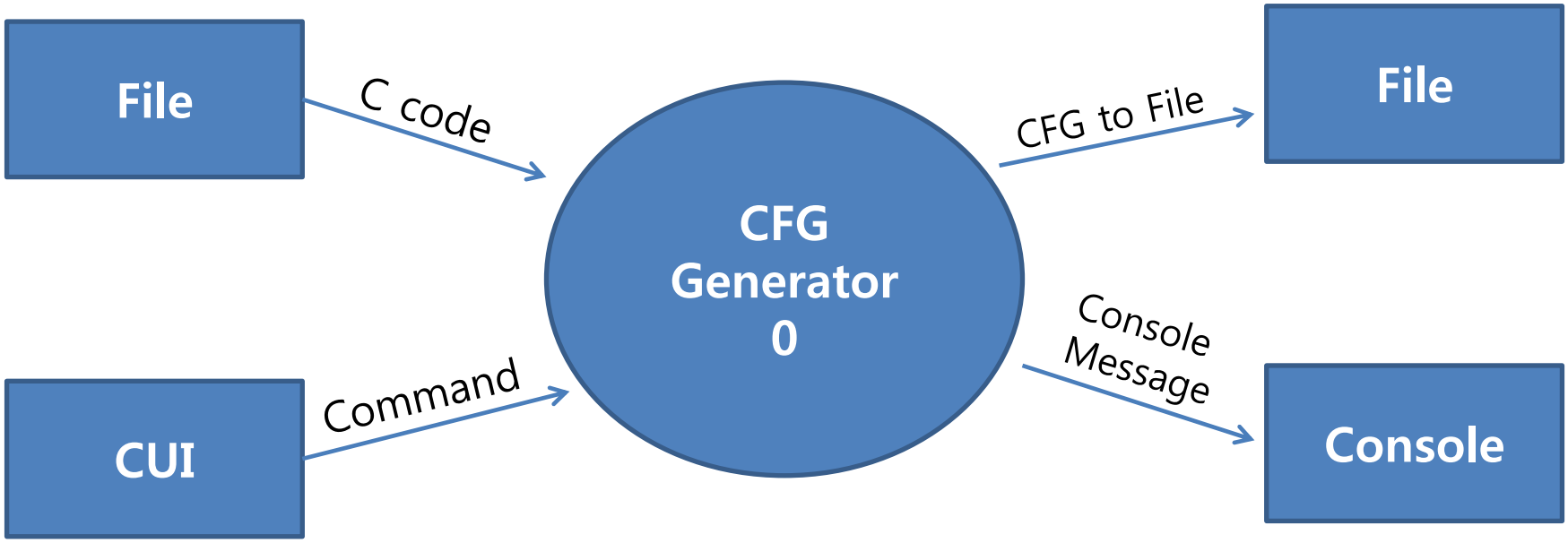


Control Process



Control Flow

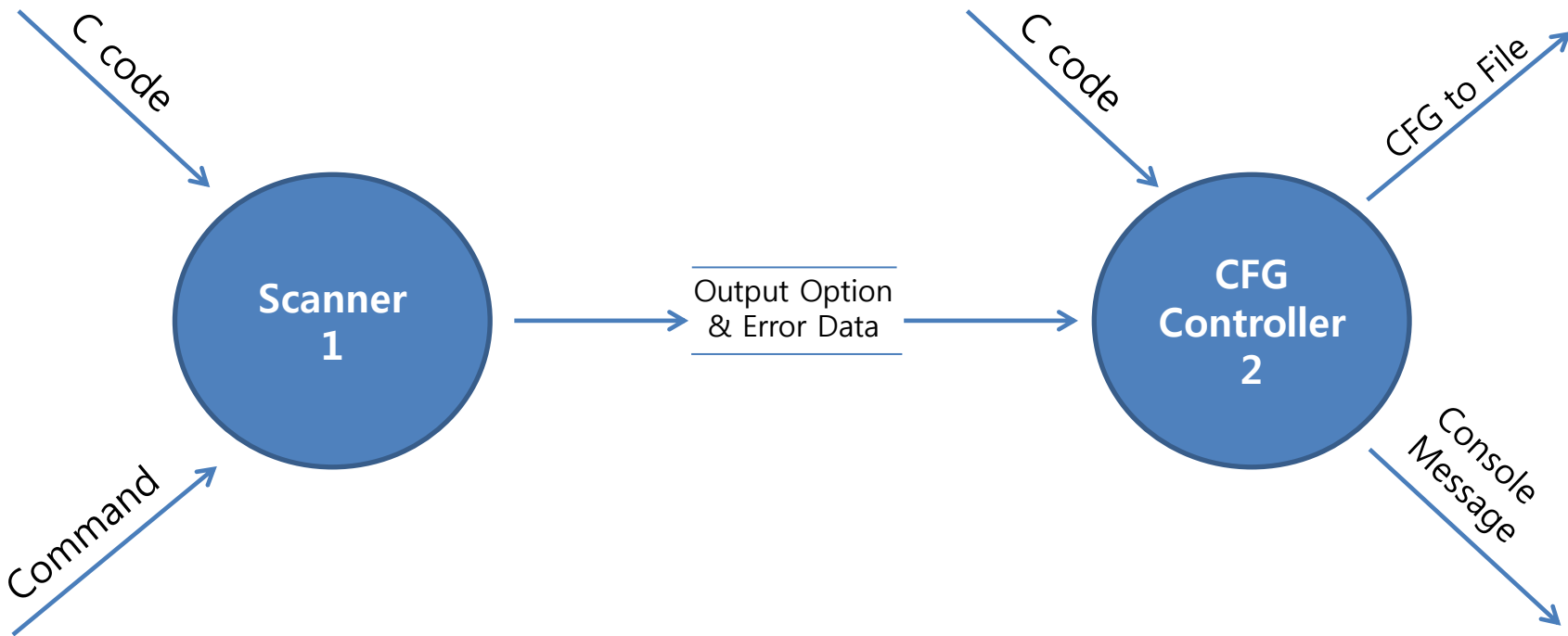
DFD Level 0



Data Dictionary Level 0

Input/Output Event	Description	Format/Type
Command	Input File Name, Output File Name 그리고 Output Option을 포함한 문자열 Ex) ./cfg [option] [input file name] [output file name]	string
C Code	CFG로 변환될 C code로 *.c의 확장자 명을 가지는 파일. C 표준에 의해 생성된 소스코드이다.	*.c
CFG to File	File에 출력될 CFG로서 string형식으로 파일에 쓰여지게 된다.	string
Console Message	콘솔에 출력할 메시지들과 CFG로 각 상태에 맞는 메시지들을 콘솔에 출력하도록 해준다.	CmdError / FileError / HelpMessage / GenerateMessage / CFG / OutFileName

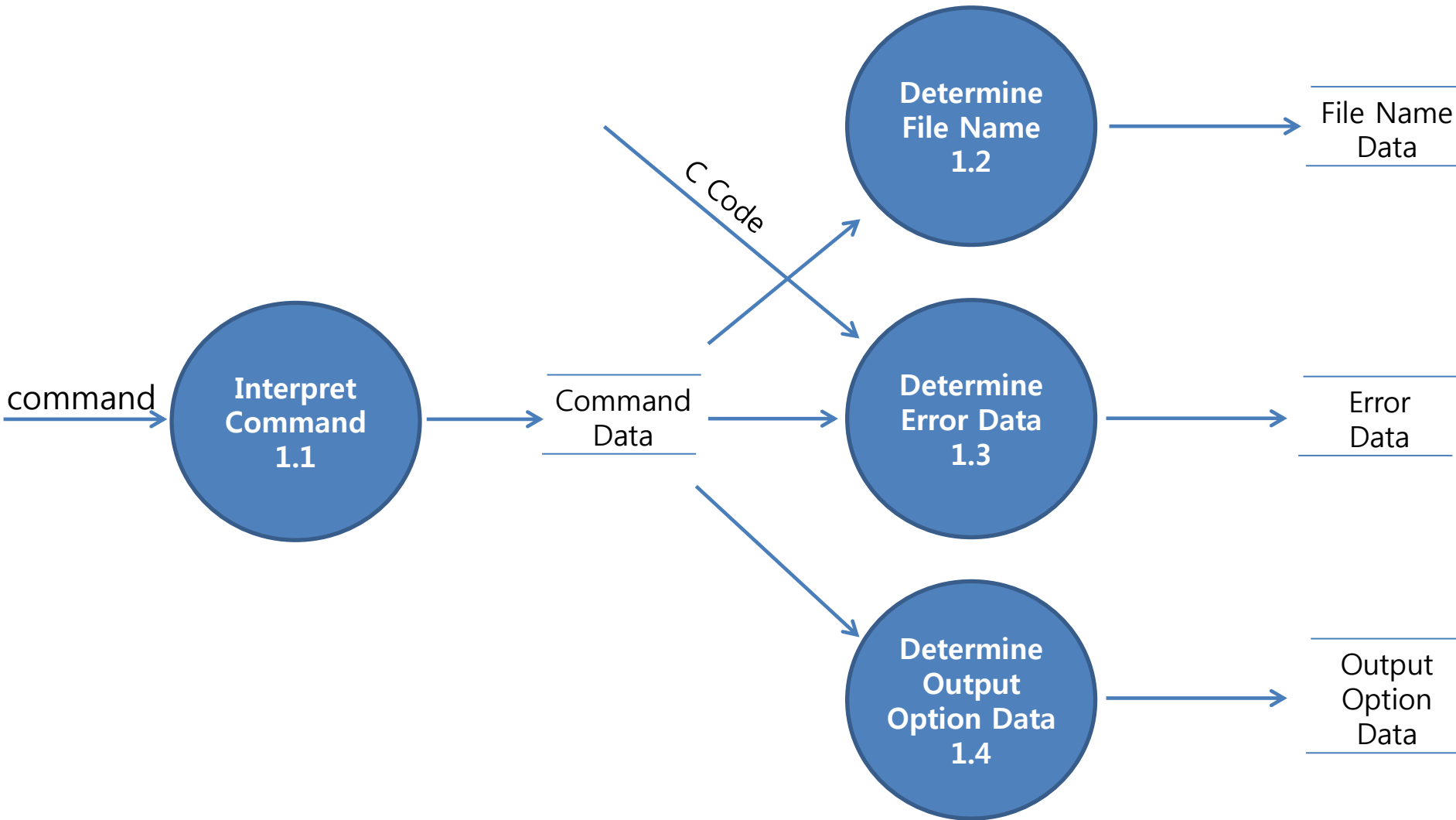
DFD Level 1



Data Dictionary Level 1

Input/Output Event	Description	Format / Type
Output option & Error Data	<p>Output Option과 Error Data를 저장 Error Data의 경우 ED 의 변수명을 갖는 string타입으로 저장되며, 각 상황에 따라 나뉜다.</p> <p>*CE (Command Error : command line으로 부터 입력받은 명령어의 문법& 형식), FE (File Error : 파일의 포맷과, 파일의 존재유무), NE (None Error : 에러가 없을경우)로 각각 표현되게 된다. Ex) string ED = "NE";</p> <p>Output Option Data의 경우는 OO의 변수명을 갖는 string타입으로 저장되며, 각 상황에 따라 나뉜다.</p> <p>*PF(print only file), PC(print console with file)을 표현하게 된다. Ex string OO = "PC";</p>	string

DFD Level 2



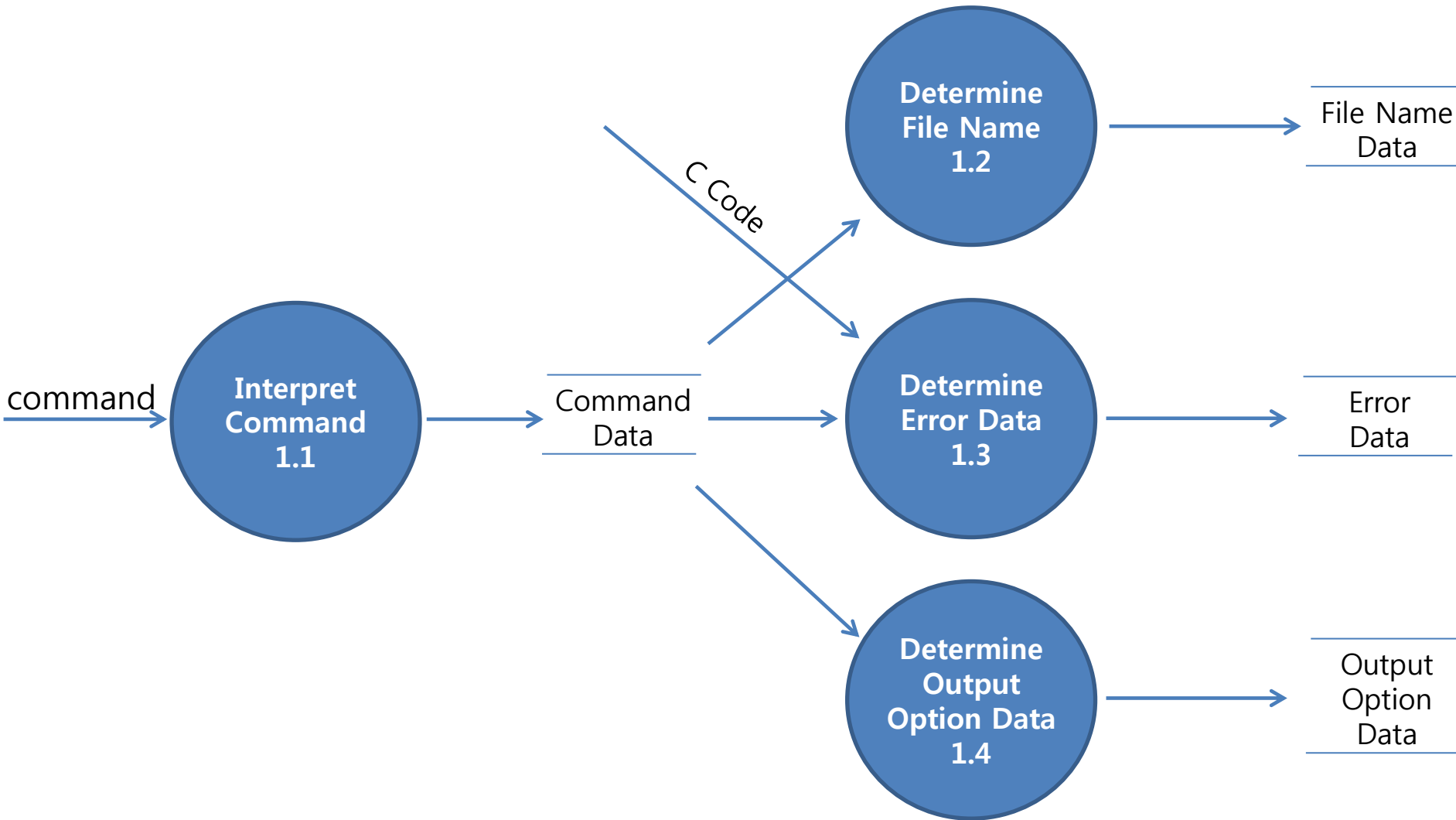
Data Dictionary Level 2

Input/Output Event	Description	Format/Type
Command Data	<p>Command Data를 저장한다.</p> <p>입력 받은 명령어가 올바를 경우 미리 정해진 형태로 분석하여 struct형태로 Data들을 분리하여 저장하게 된다.</p> <p>Ex) ./cfg [option] [input file name] [output file name] 형태로 입력 받으면 각각 정보들을 string형태로 나누어 구조체에 저장한다.</p> <p>만약 적절하지 않은 명령어가 들어오면 모두 NULL값이 저장된다.</p>	Struct
File Name Data	Command Data의 정보를 이용하여 Input File Name과 Output File Name을 구조체에 저장한다	Struct

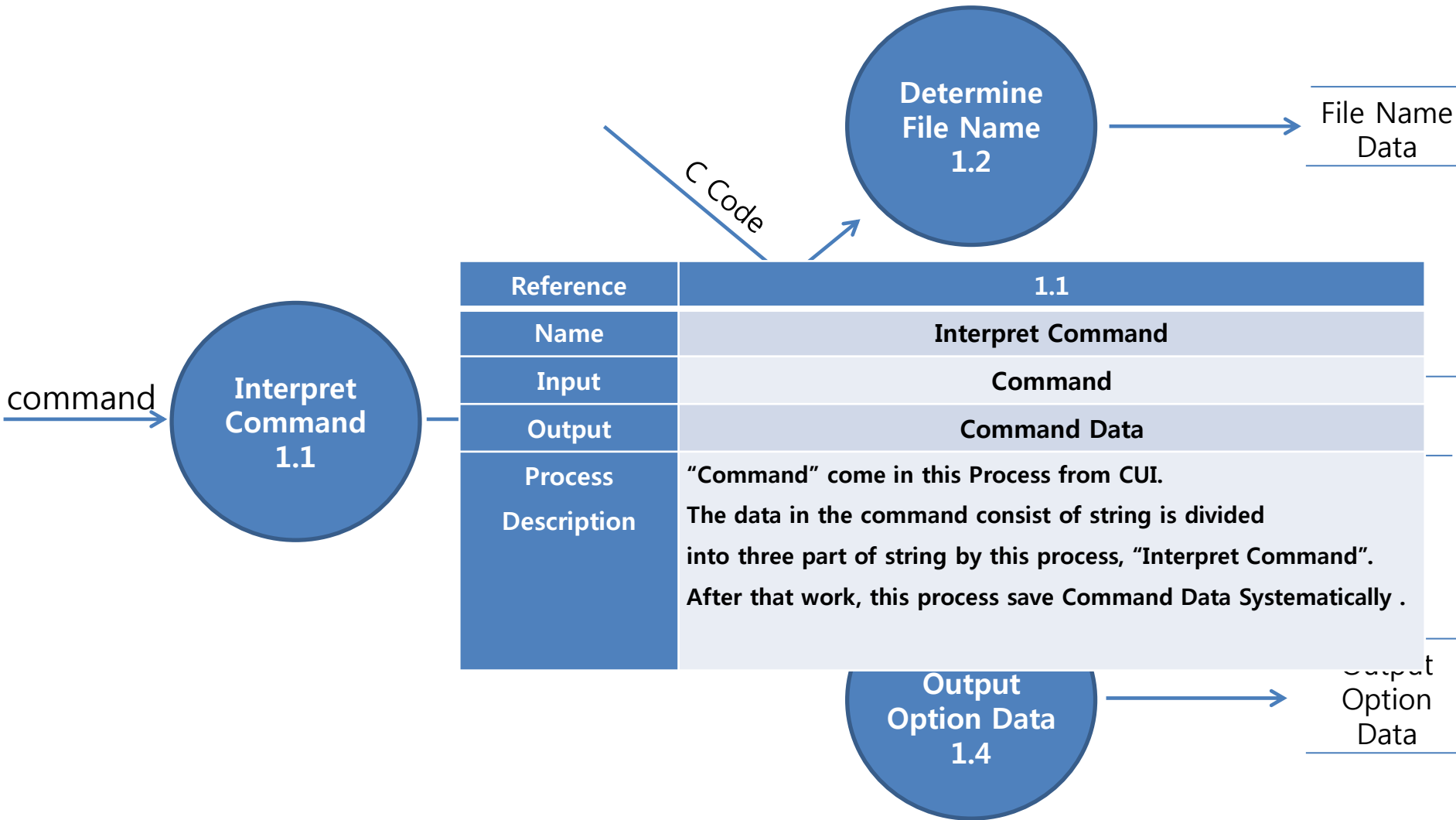
Data Dictionary Level 2

Input/Output Event	Description	Format/Type
Error Data	<p>Command Data를 분석하여 명령어의 오류와 File Error를 체크하여 그에 상응하는 값을 string 형태로 저장하게 된다. ED 의 변수명을 갖는 string타입으로 저장되며, 각 상황에 따라</p> <p>CE (Command Error : command line으로 부터 입력 받은 명령어의 문법& 형식),</p> <p>FE (File Error : 파일의 포맷과, 파일의 존재유무)</p> <p>NE (None Error : 에러가 없을 경우) 중 하나의 문자열을 저장한다.</p> <p>Ex) string ED = "NE";</p>	string
Option Data	<p>Command Data로 부터 옵션을 받아와 OO의 변수명을 갖는 string타입으로 저장되며, 각 상황에 따라 PF(print only file), PC(print console with file)중 하나의 문자열을 저장한다.</p> <p>Ex) string OO = "PC";</p>	string

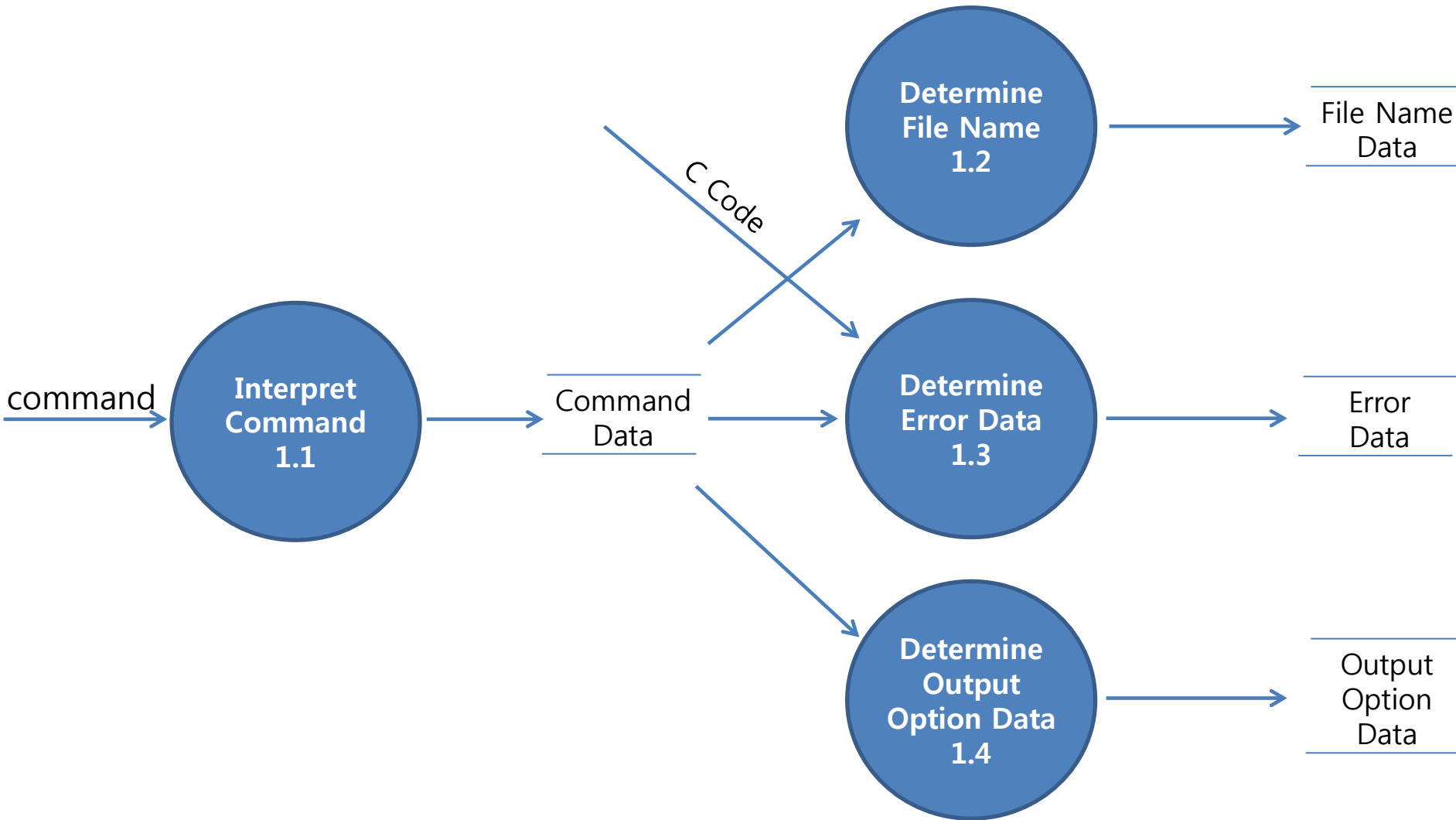
DFD Level 2



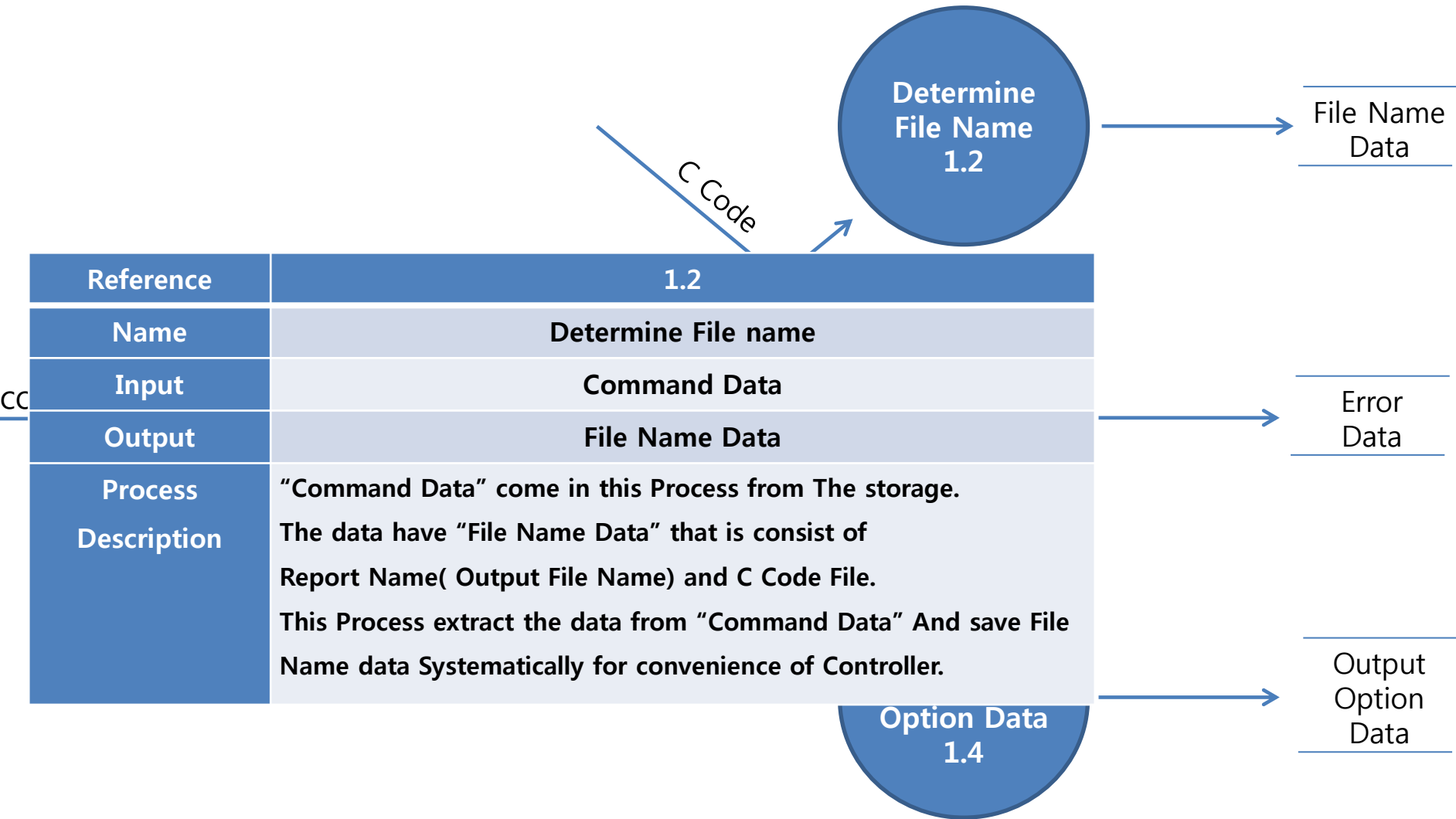
DFD Level 2



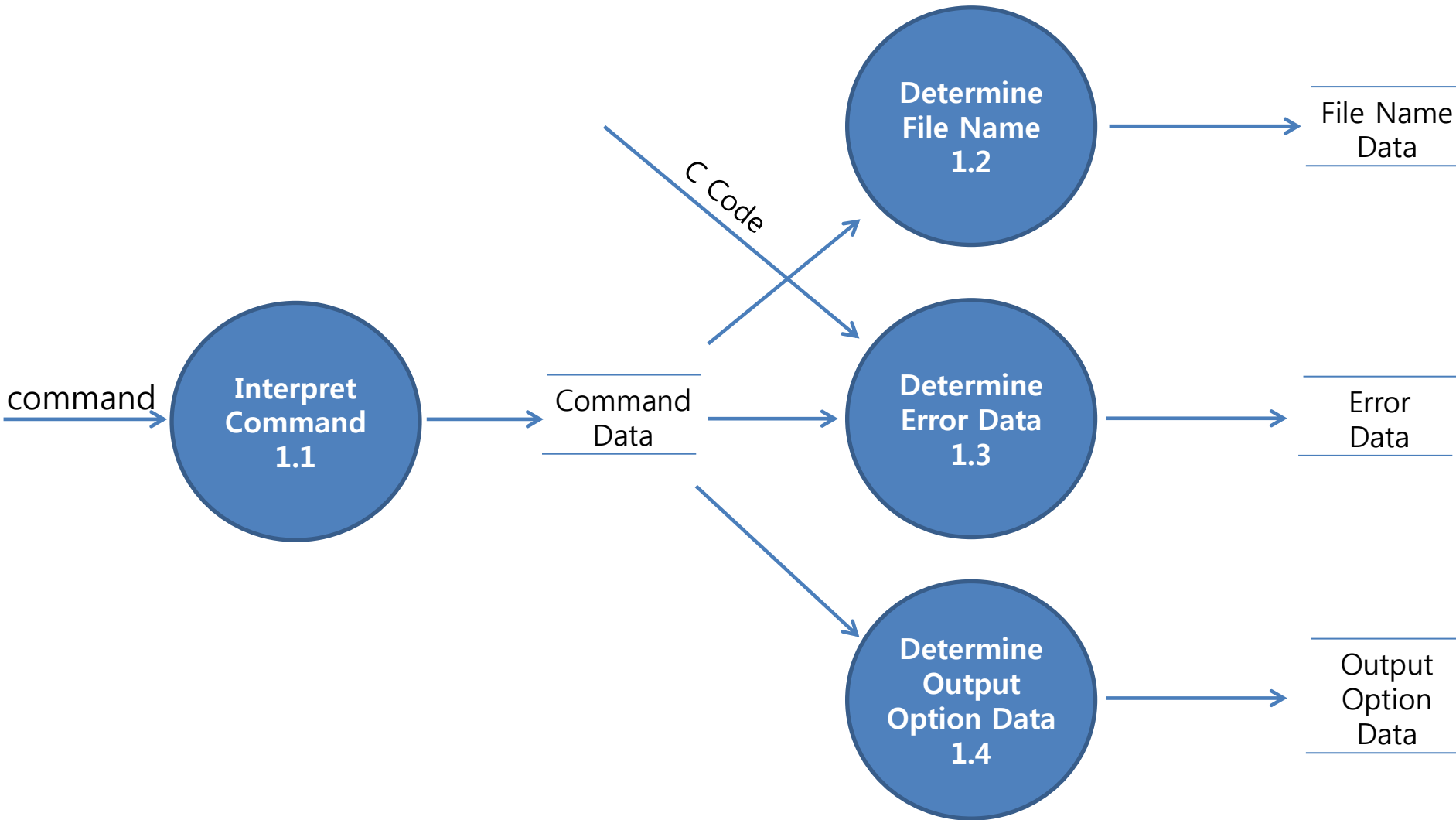
DFD Level 2



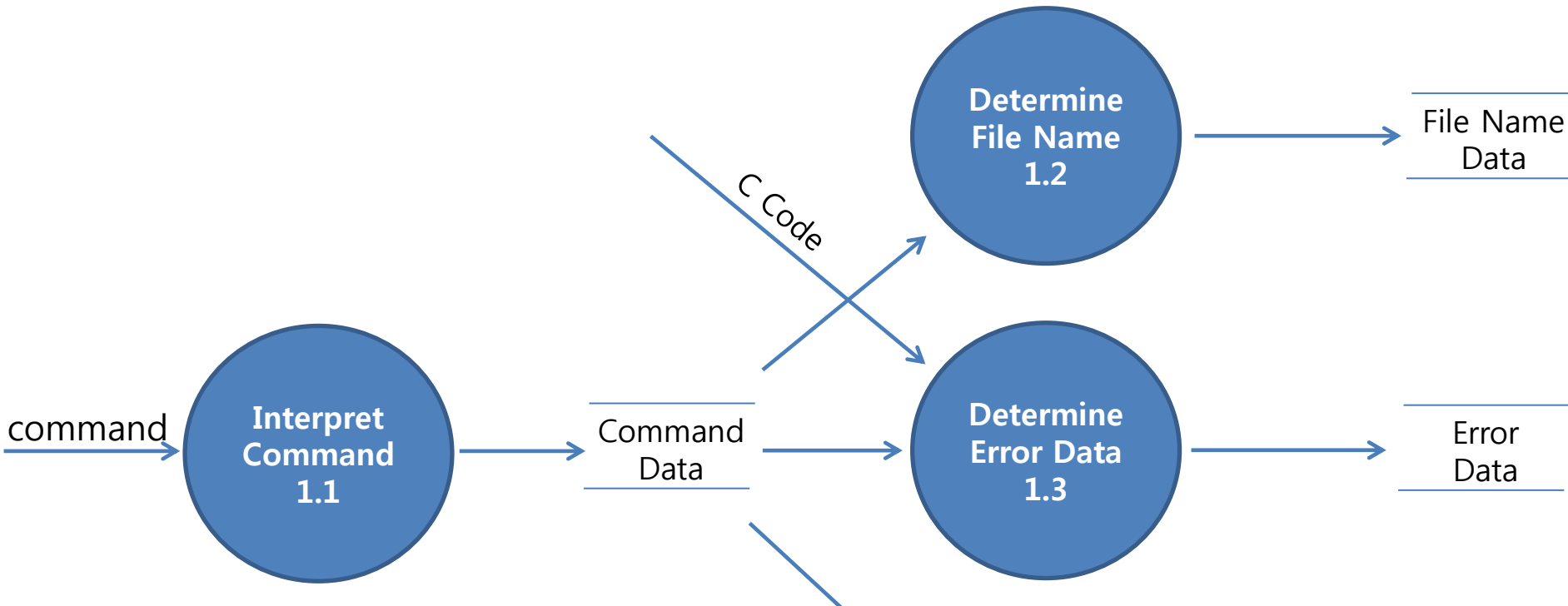
DFD Level 2



DFD Level 2

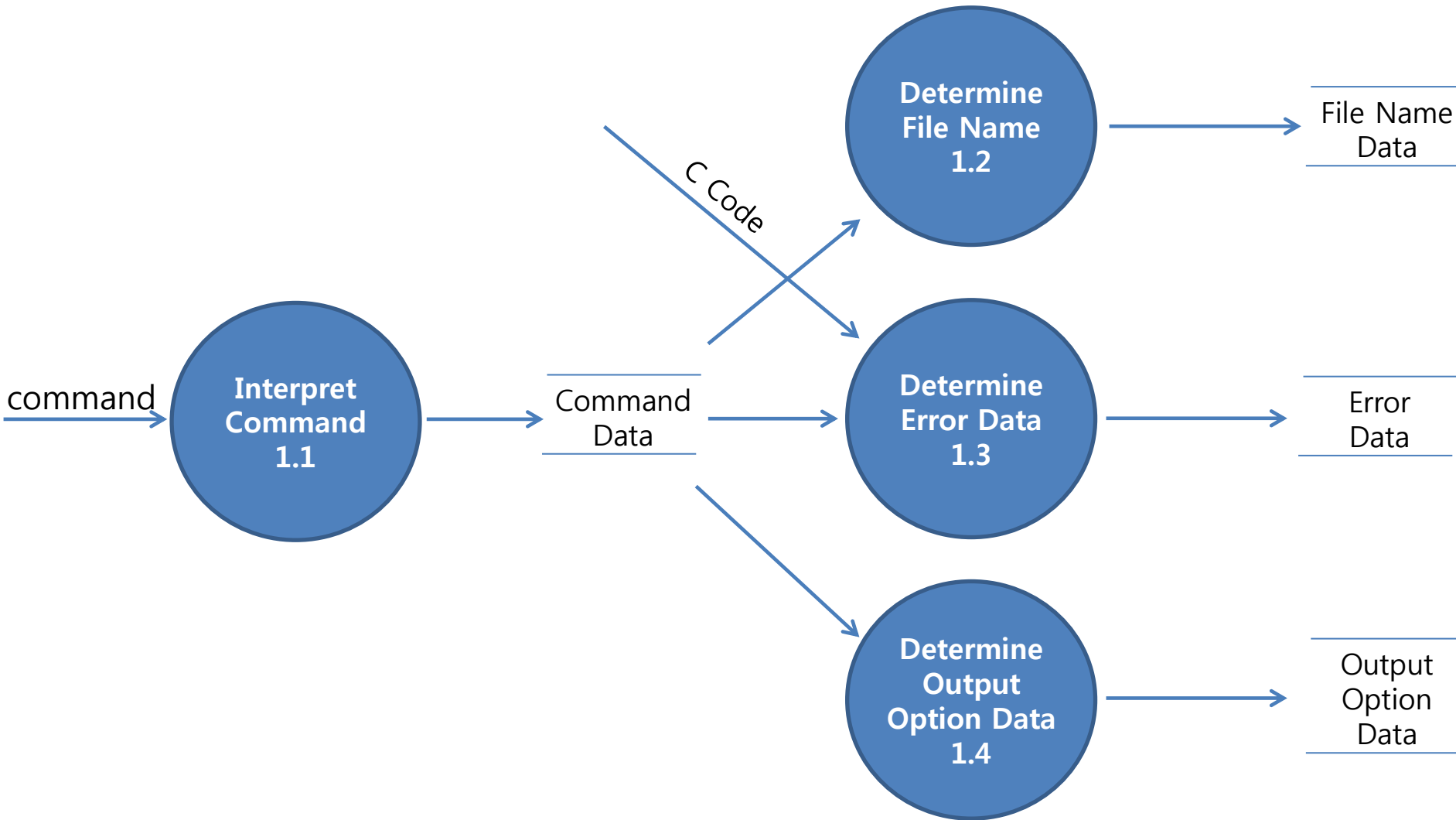


DFD Level 2

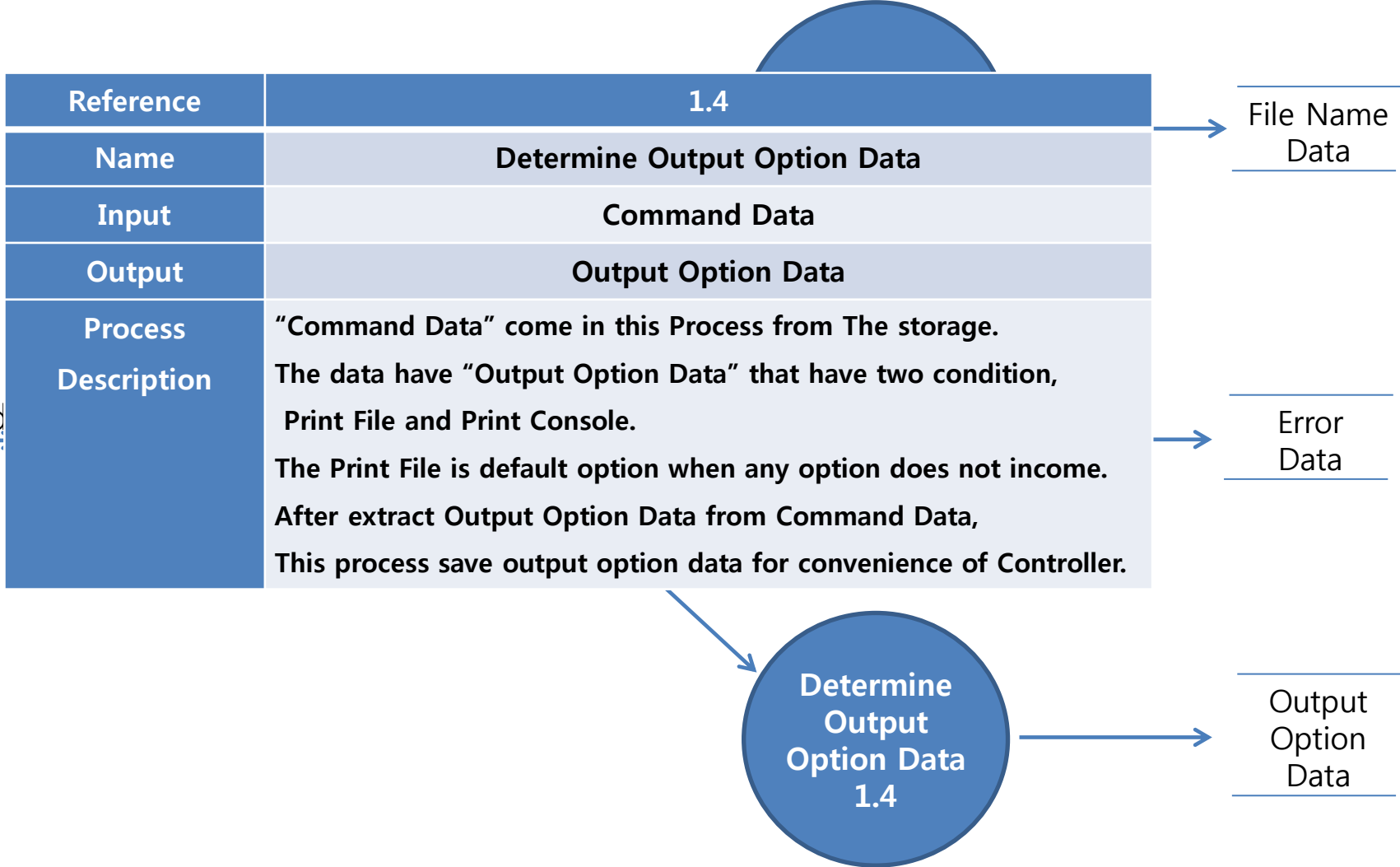


Reference	1.3
Name	Determine Error Data
Input	Command Data
Output	Error Data
Process Description	<p>"Command Data" come in this Process from The storage.</p> <p>The data have "Error Data" that is consist of File Error Data and Cmd Error Data.</p> <p>This Process try to open the C Code File to check the File's validity.</p>

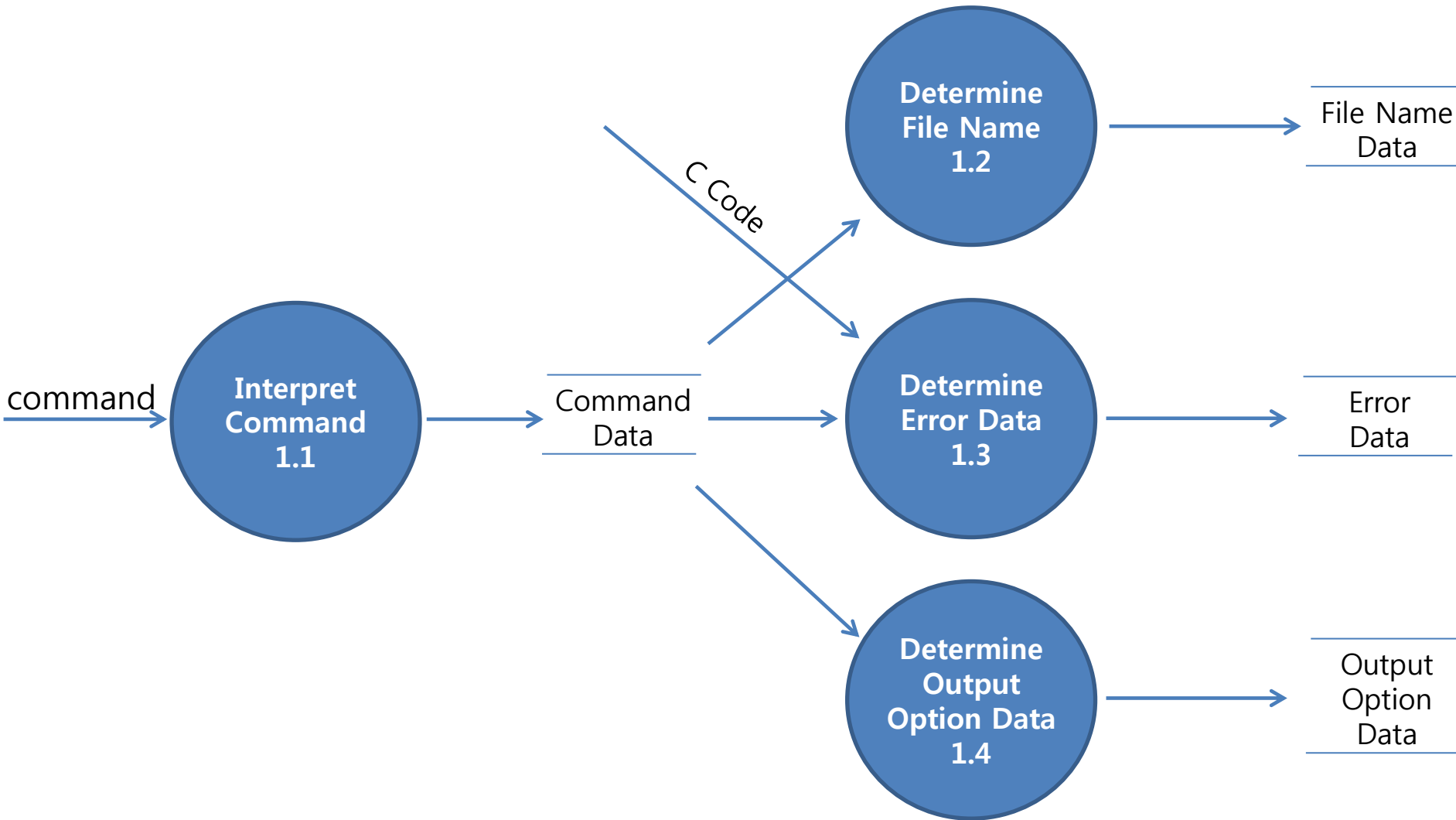
DFD Level 2



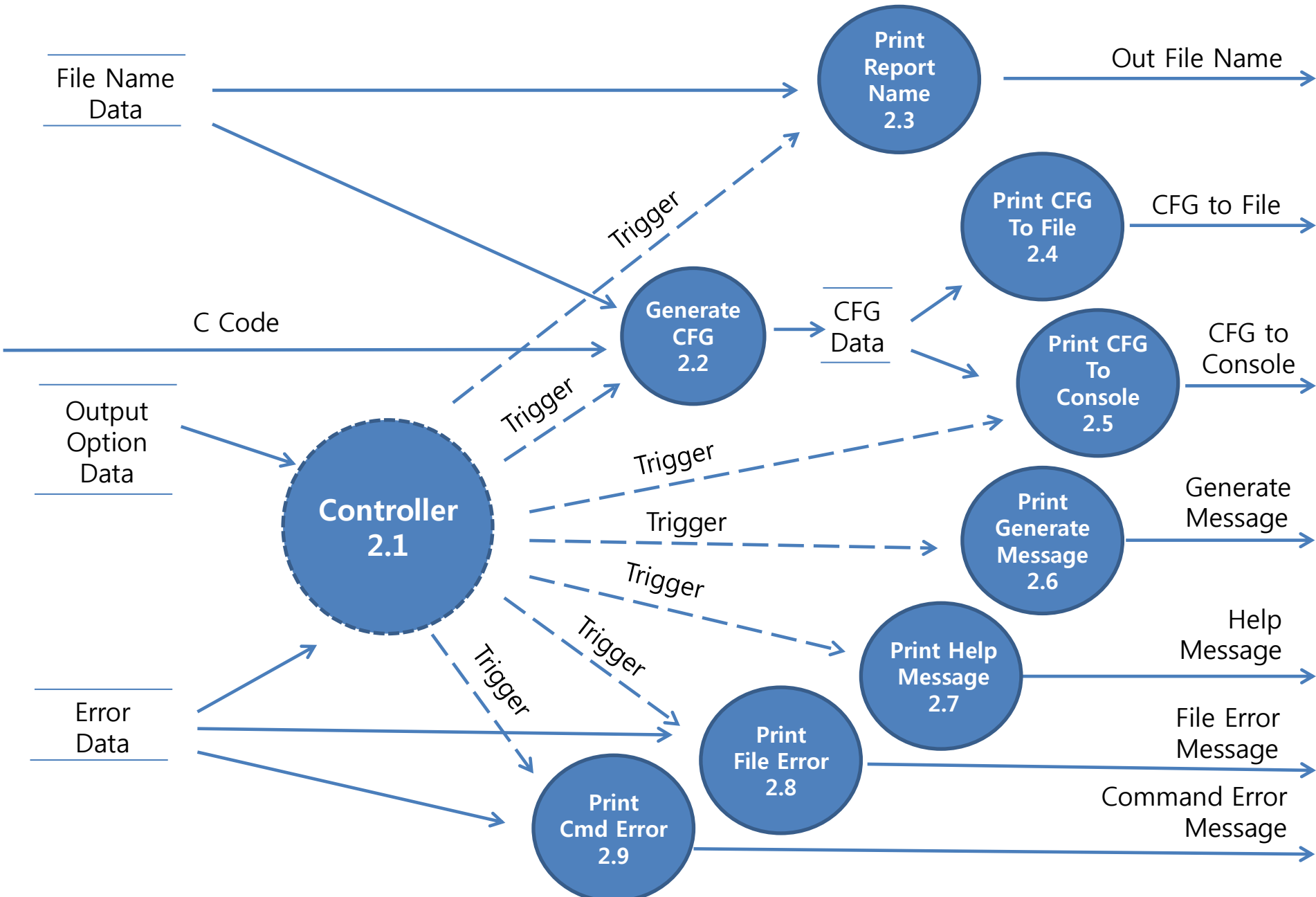
DFD Level 2



DFD Level 2



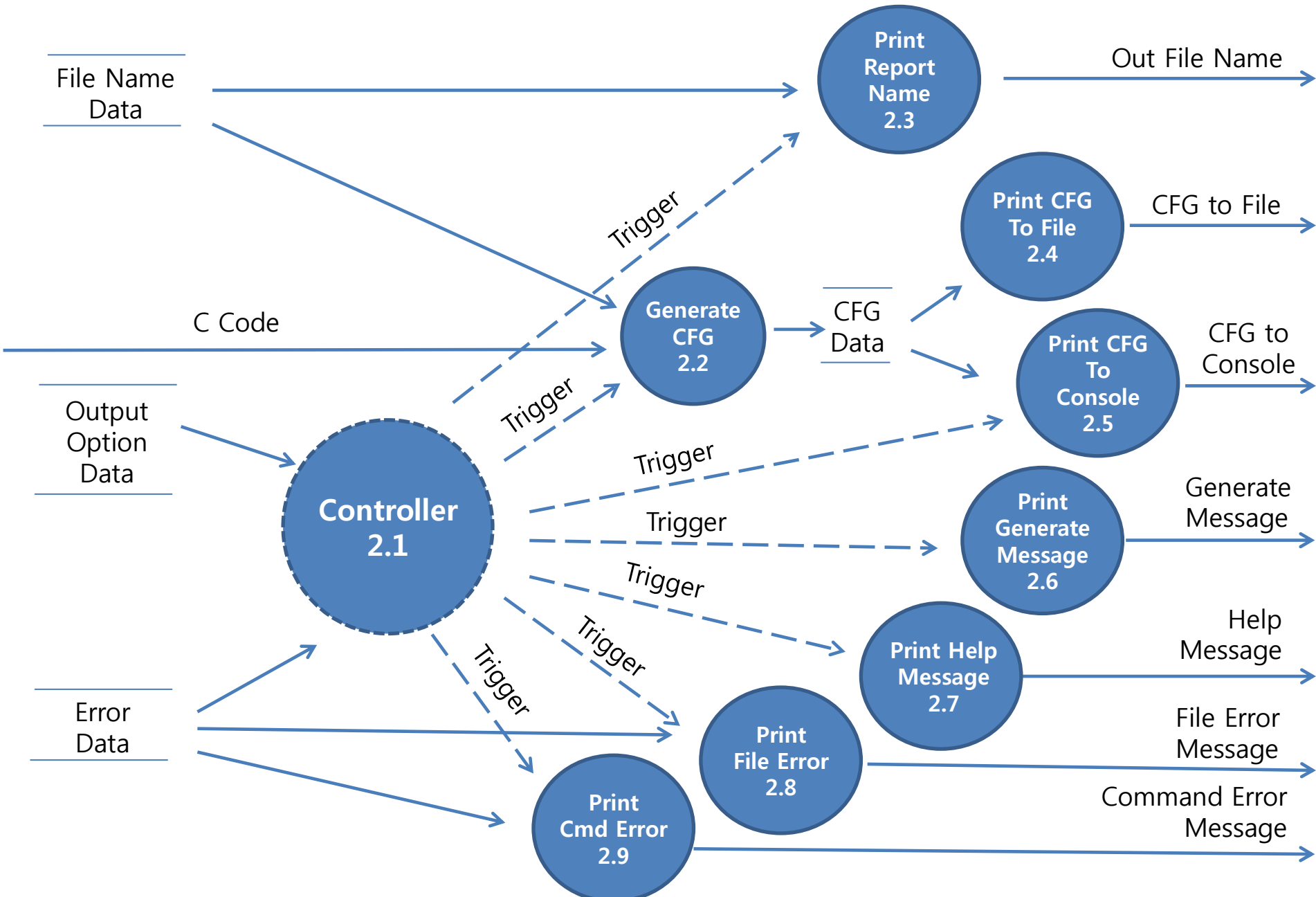
DFD Level 2



Data Dictionary Level 2

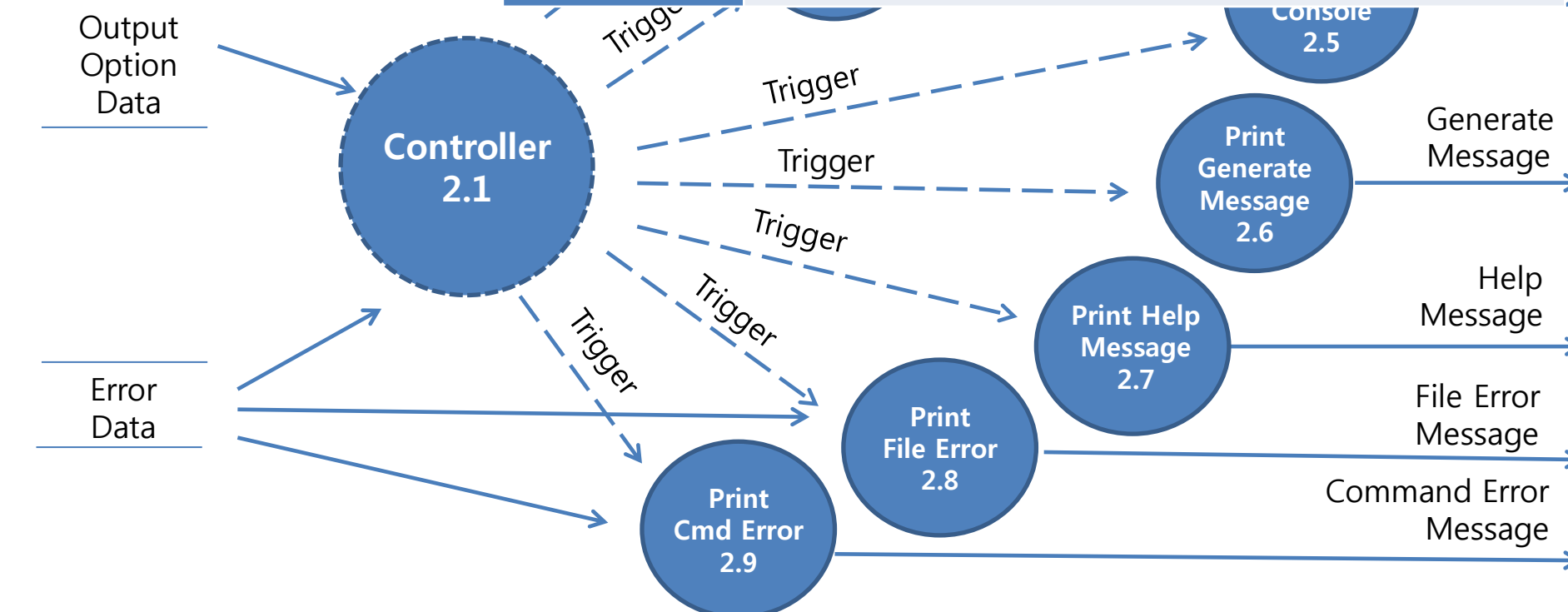
Input/Output Event	Description	Format/Type
CFG Data	CFG Data 를 저장한다. 입력된 C code로 부터 생성된 Data로 struct형태의 Node리스트 Edge리스트를 포함하는 struct로 생성되며, 이 Data를 토대로 File과 Console로 CFG를 출력하게 된다.	Struct
Command Error	Whether the cmd error occurred or not	String
Out File Name	Output file name that will convert to CFG and the file name that report will be saved in	string
File Error	Whether File exist or not	String
Help Message	<i>Help Message</i> that will outputs to Console	String
Generate Message	CFG generater's complete message	String
CFG (Console)	CFG that will outputs to console	string

DFD Level 2

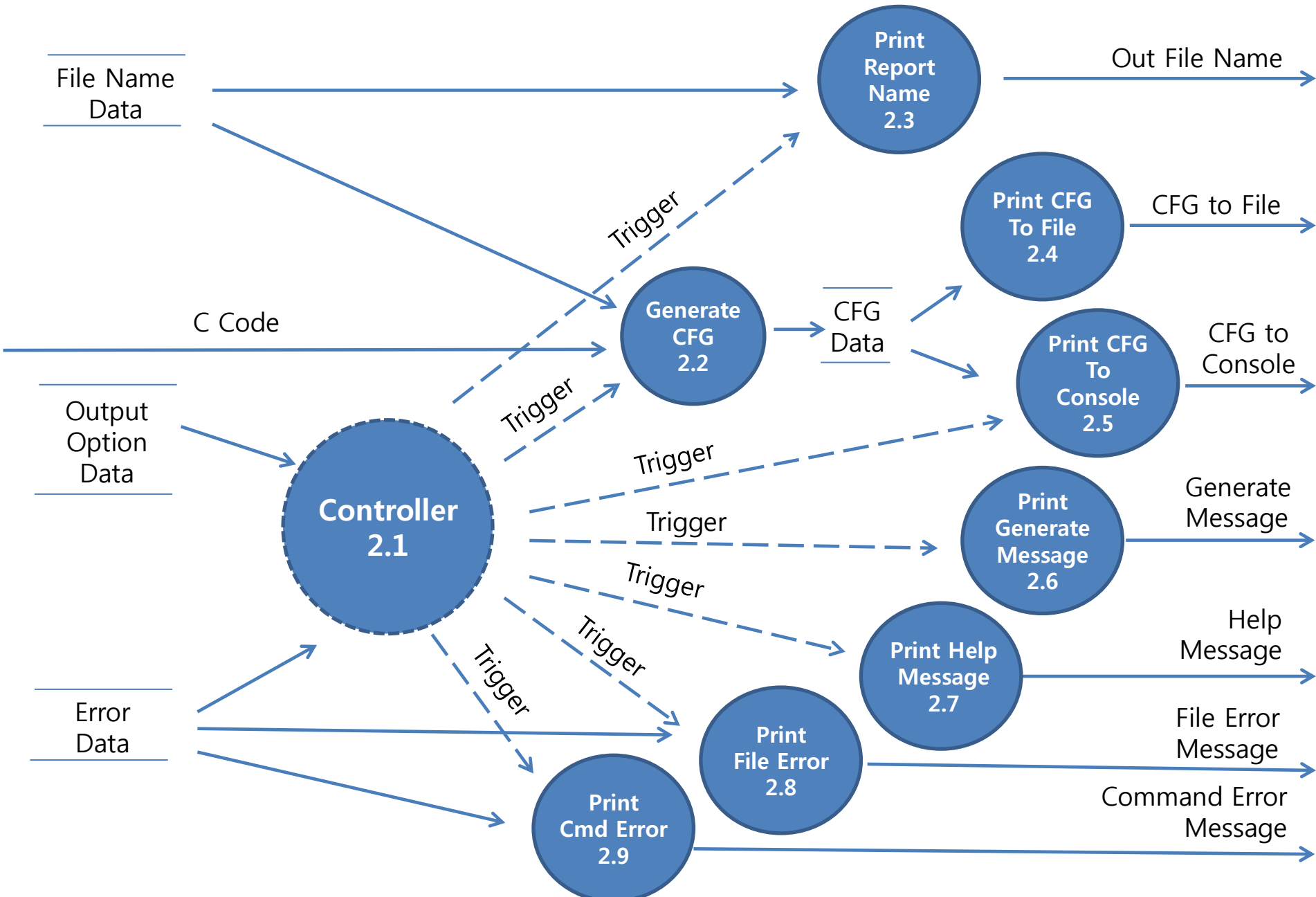


DFD Level 2

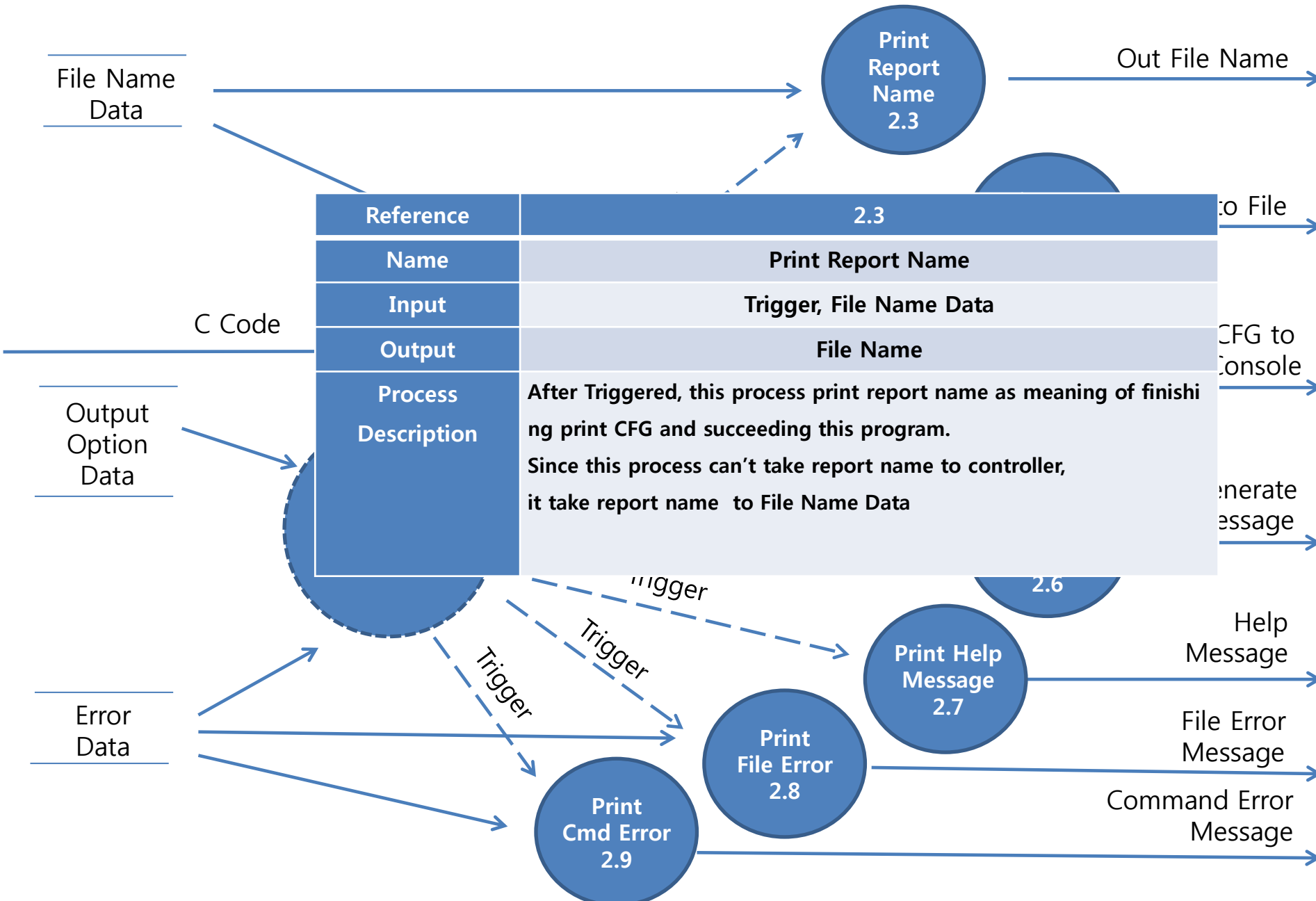
Reference	2.1
Name	Controller
Input	Error Data, Output Option Data
Output	Trigger
Process Description	"Error Data" and "Output Option data" come in this process for divergence. Controller can make a decision with that data for process of Program. The Operation of this Process is represented to "state machine" specifically.



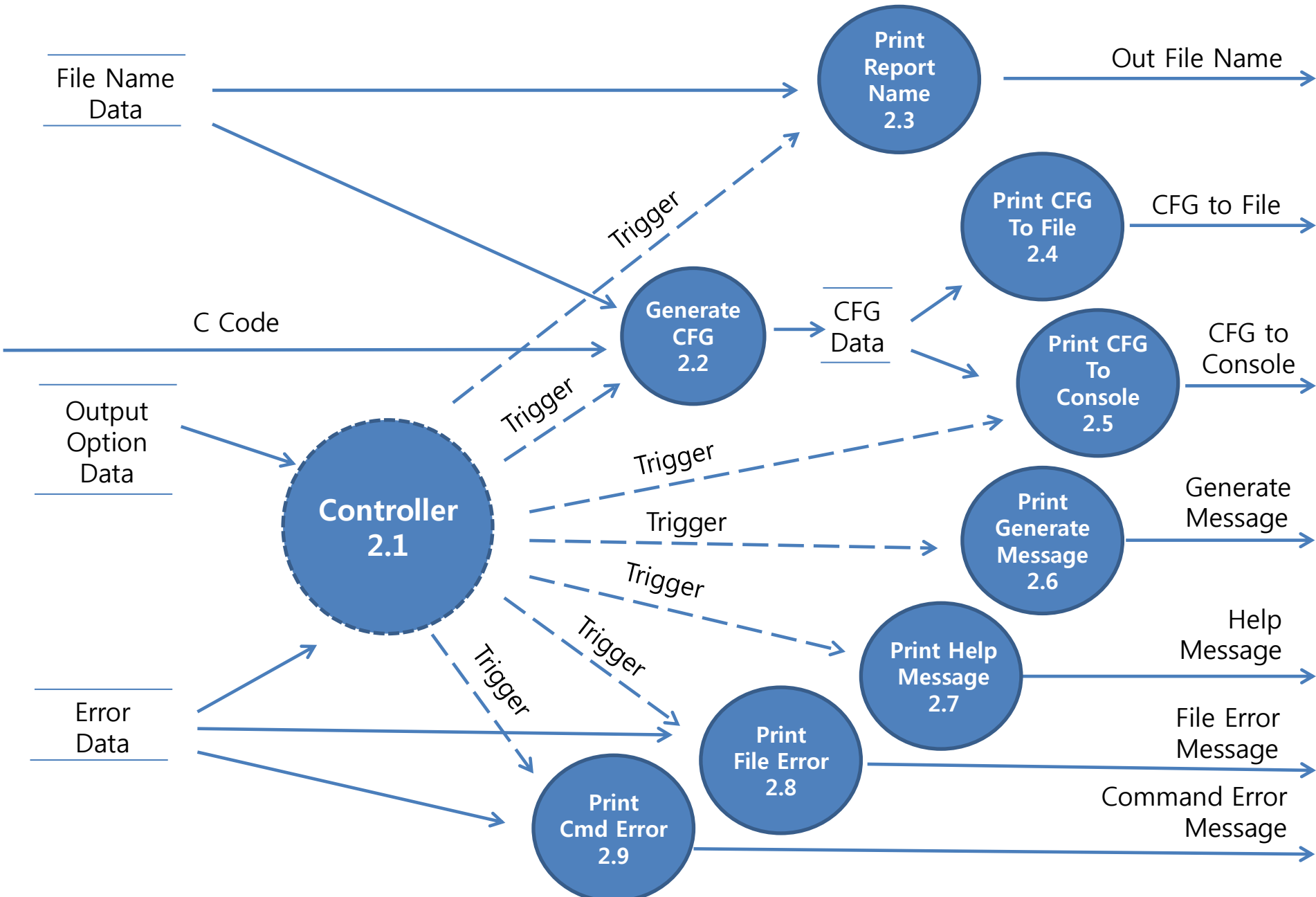
DFD Level 2



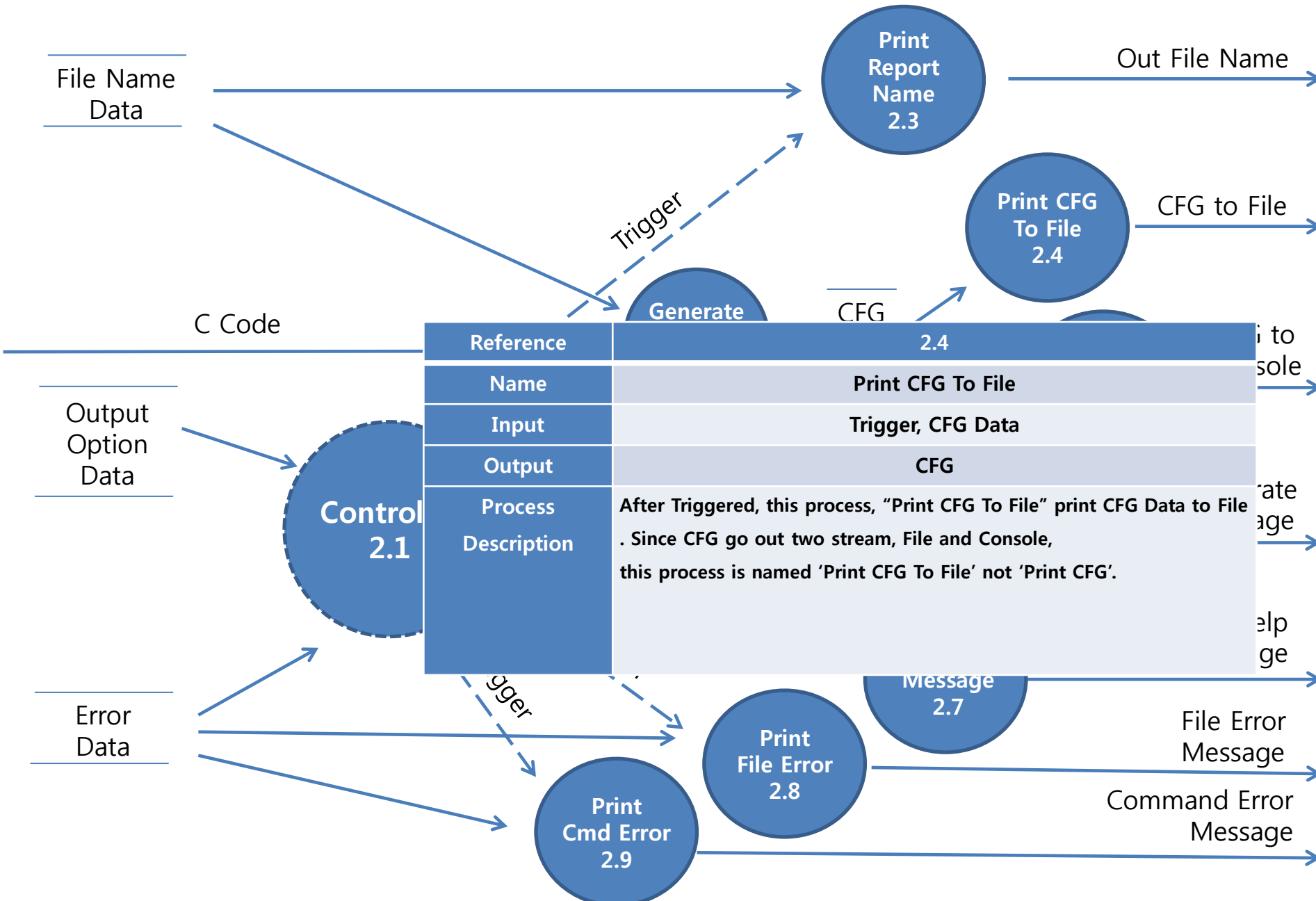
DFD Level 2



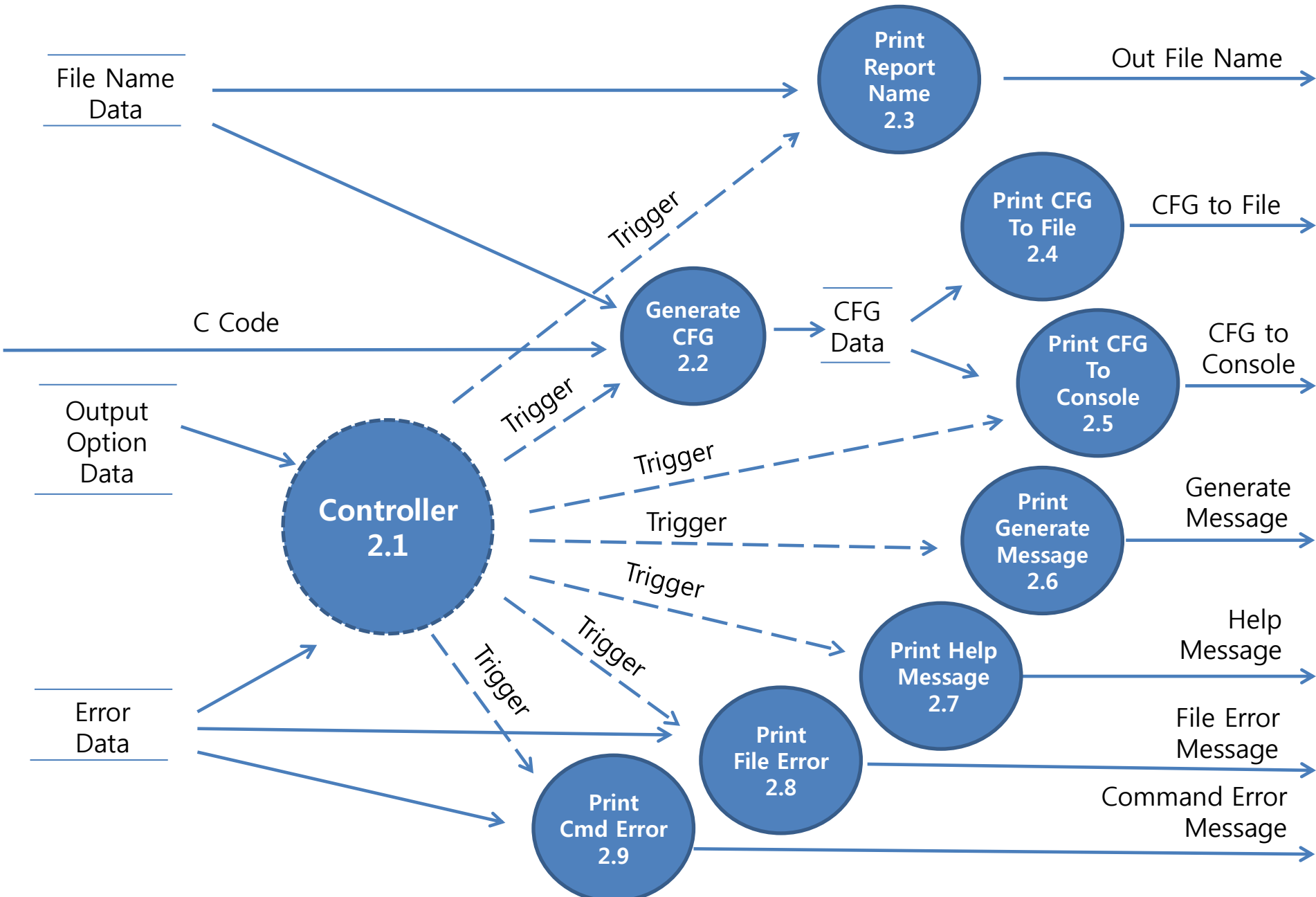
DFD Level 2



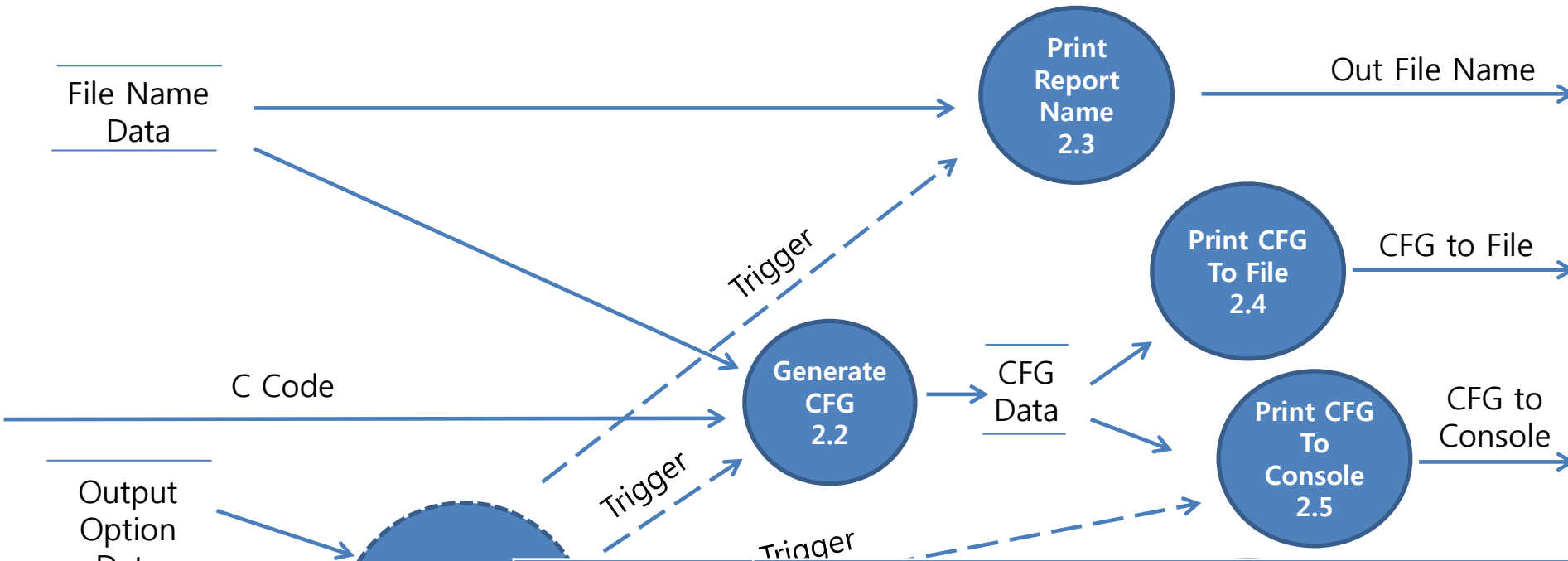
DFD Level 2



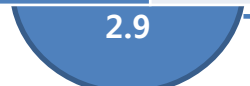
DFD Level 2



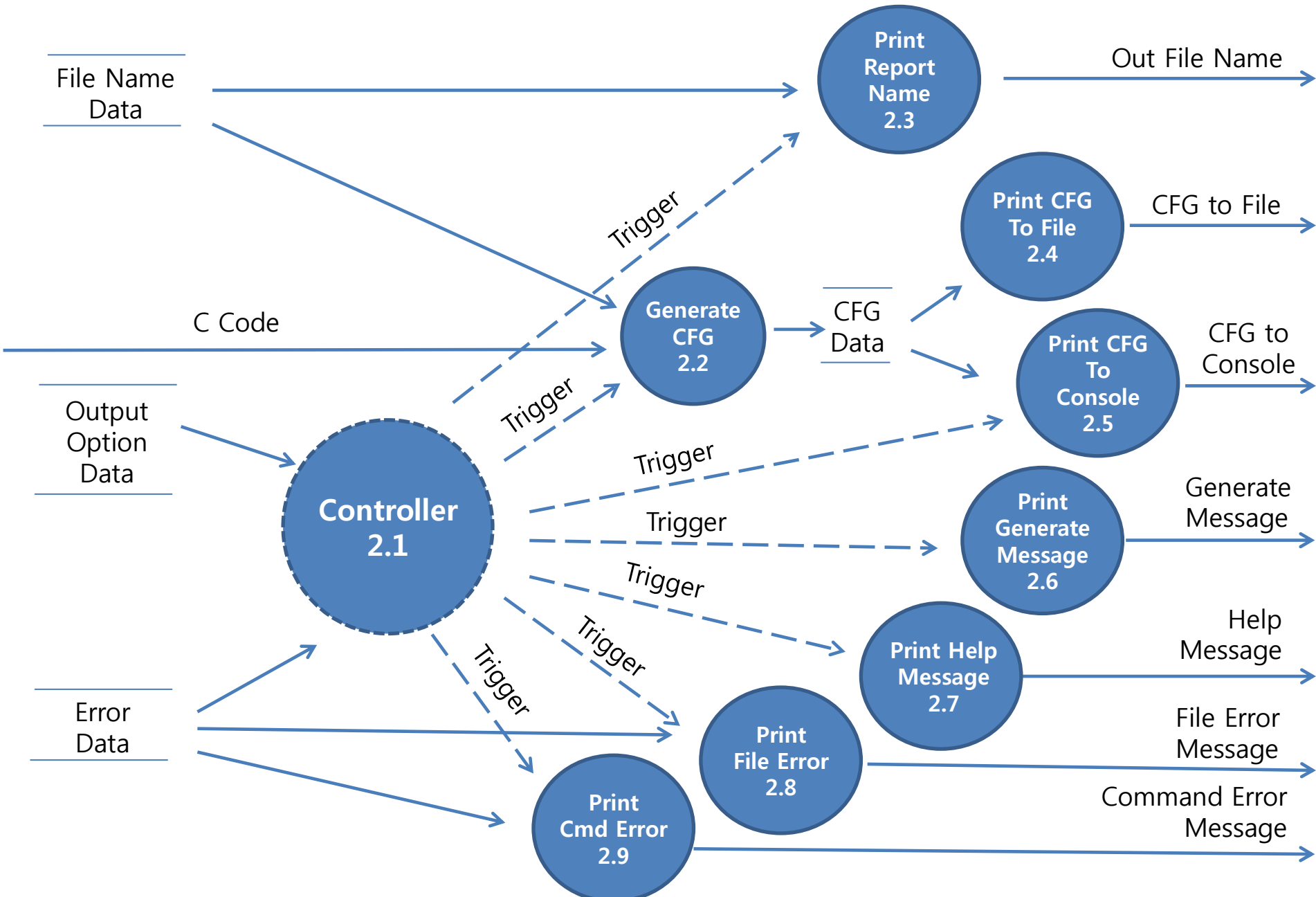
DFD Level 2



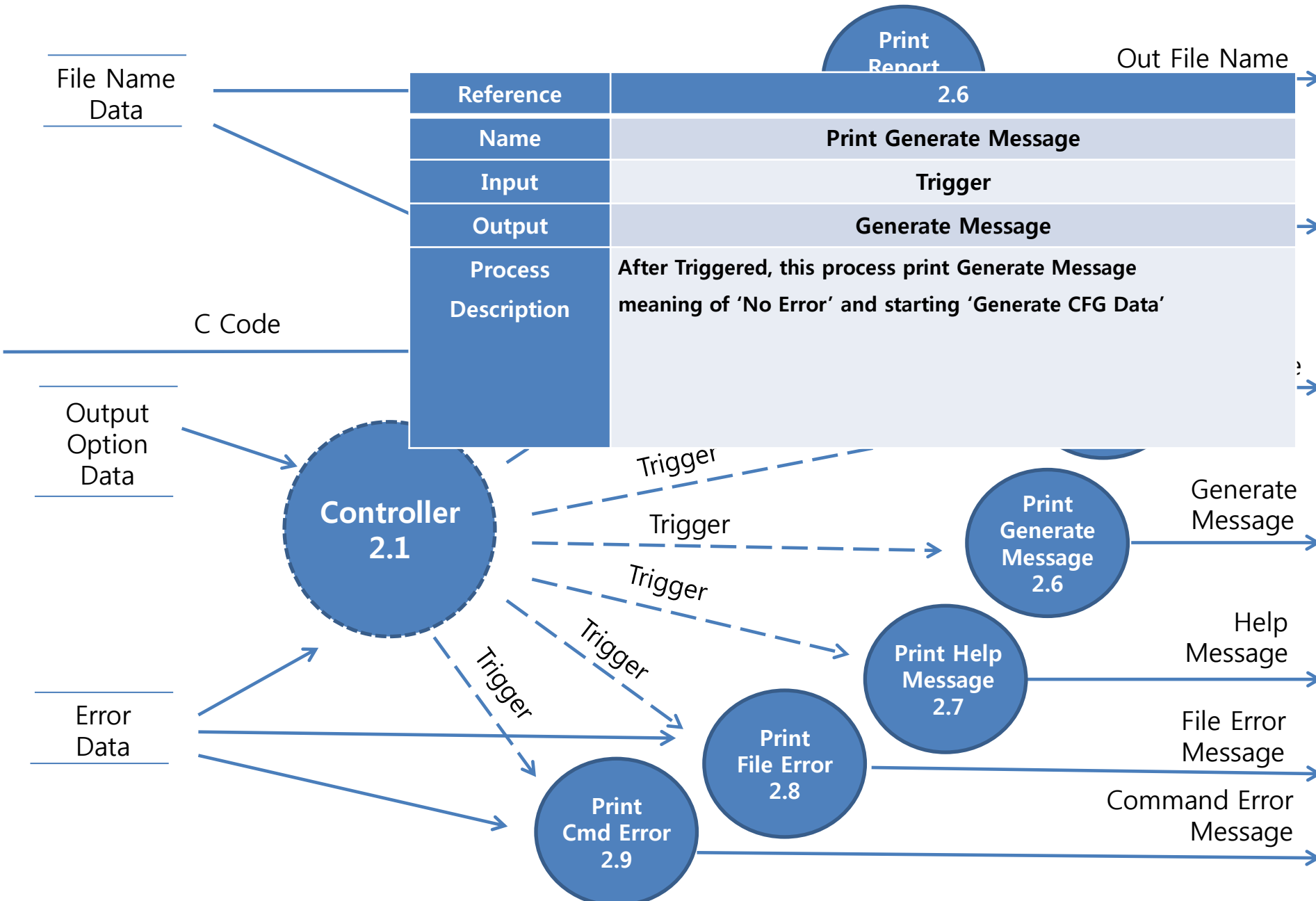
Reference	2.5
Name	Print CFG To Console
Input	Trigger , CFG Data
Output	CFG
Process Description	After Triggered, this process ,'Print CFG To Console', print CFG Data to Console. This process operate selectively, the contrary "Print CFG To File" operate unconditionally after generate CFG.



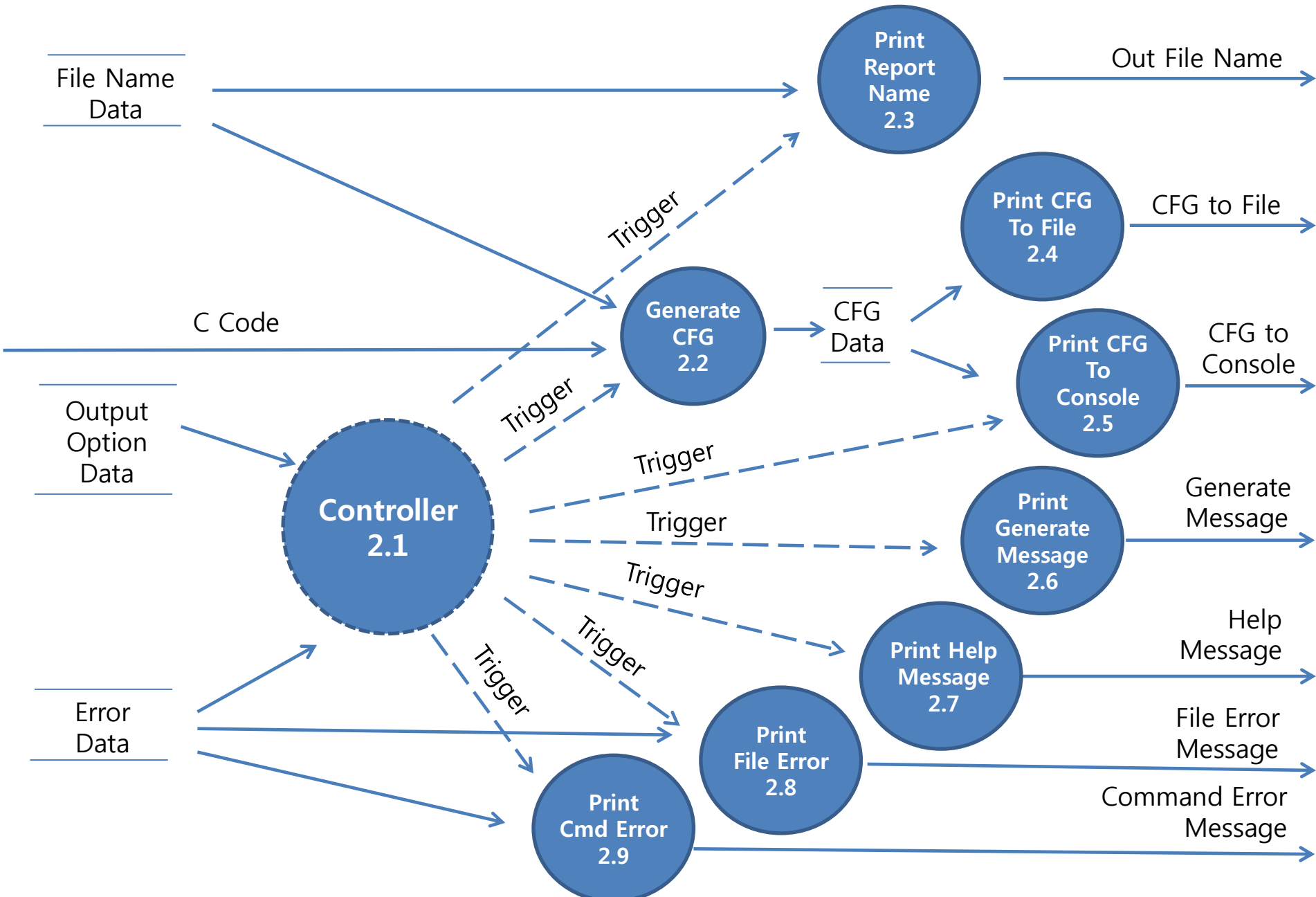
DFD Level 2



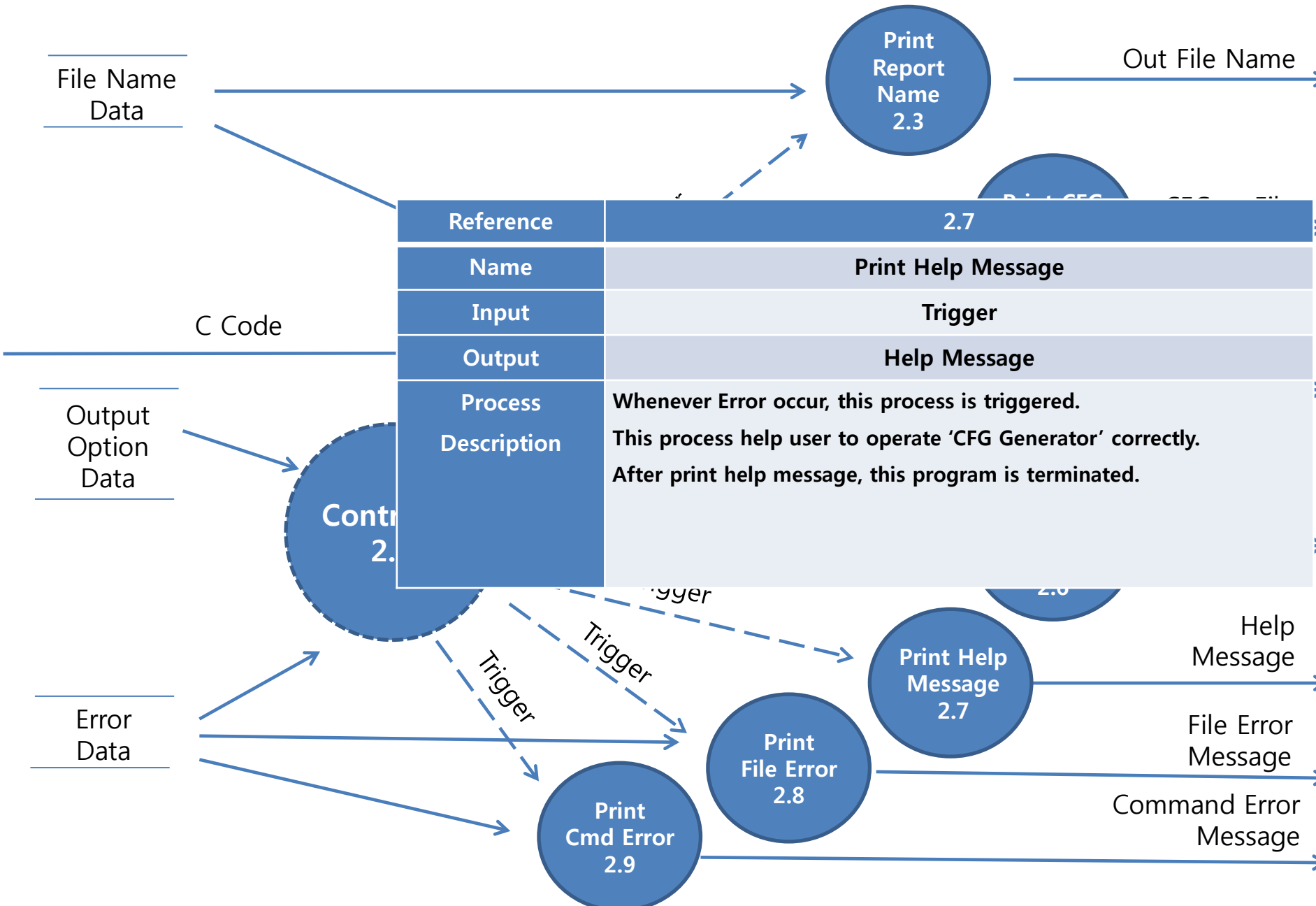
DFD Level 2



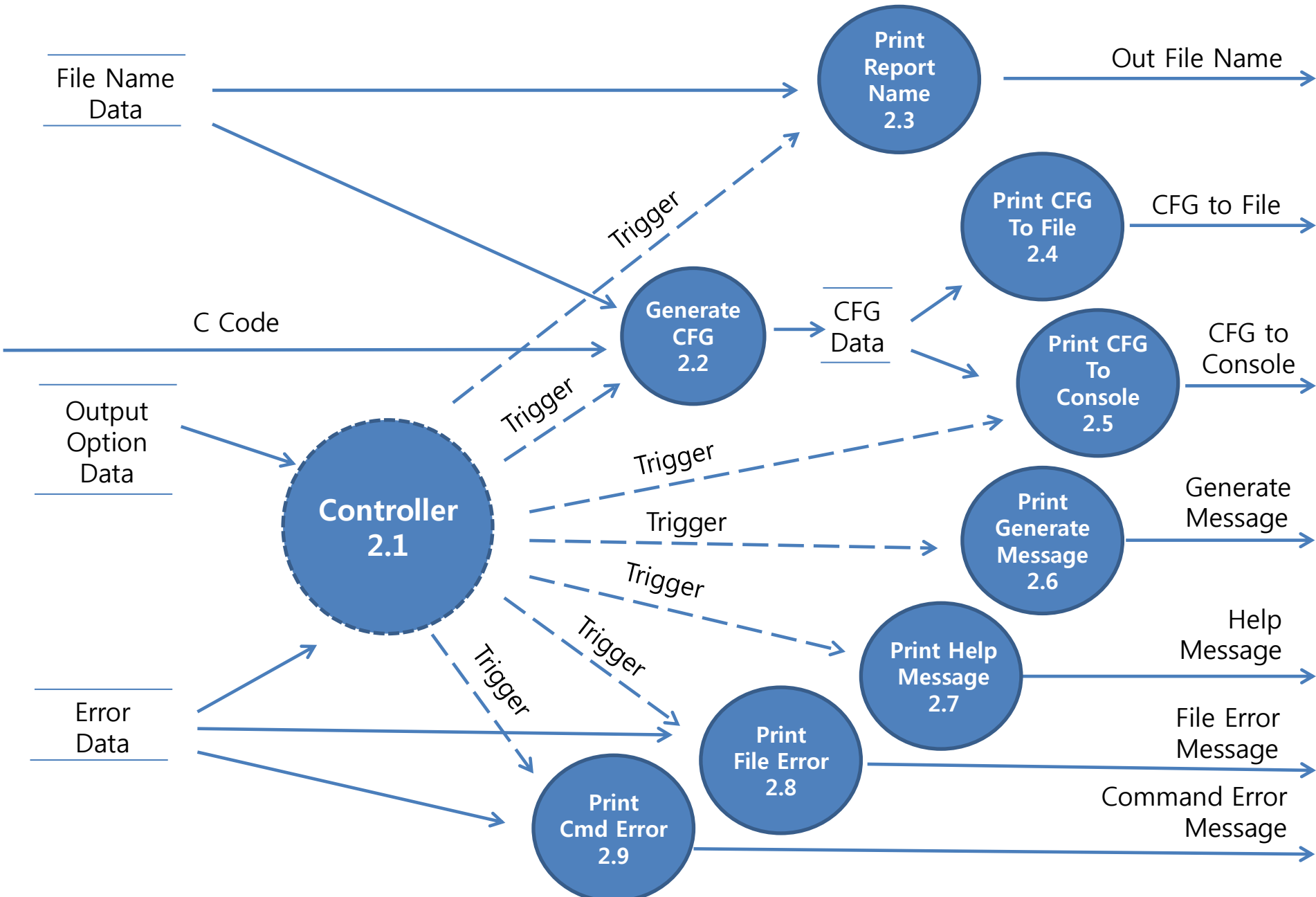
DFD Level 2



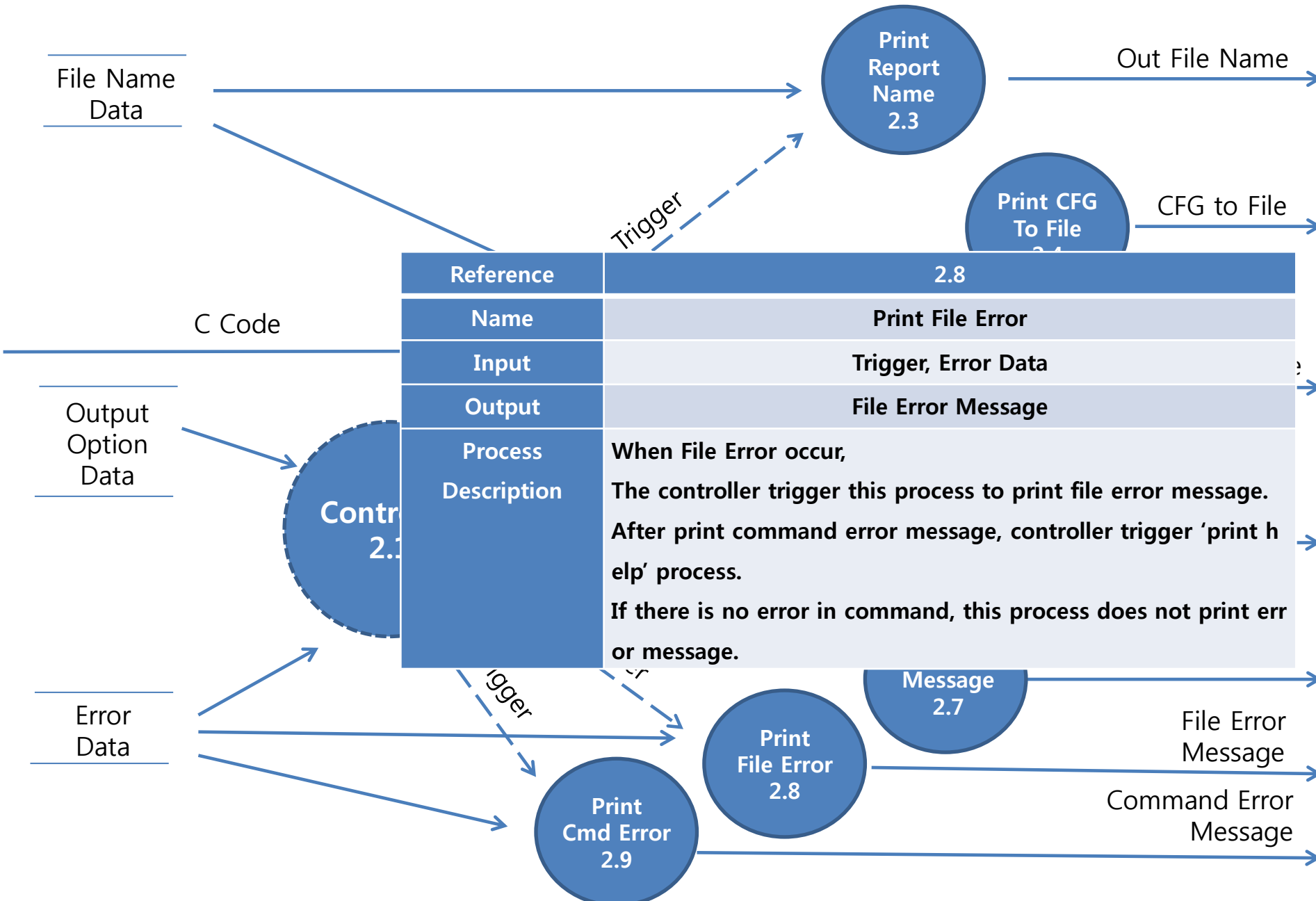
DFD Level 2



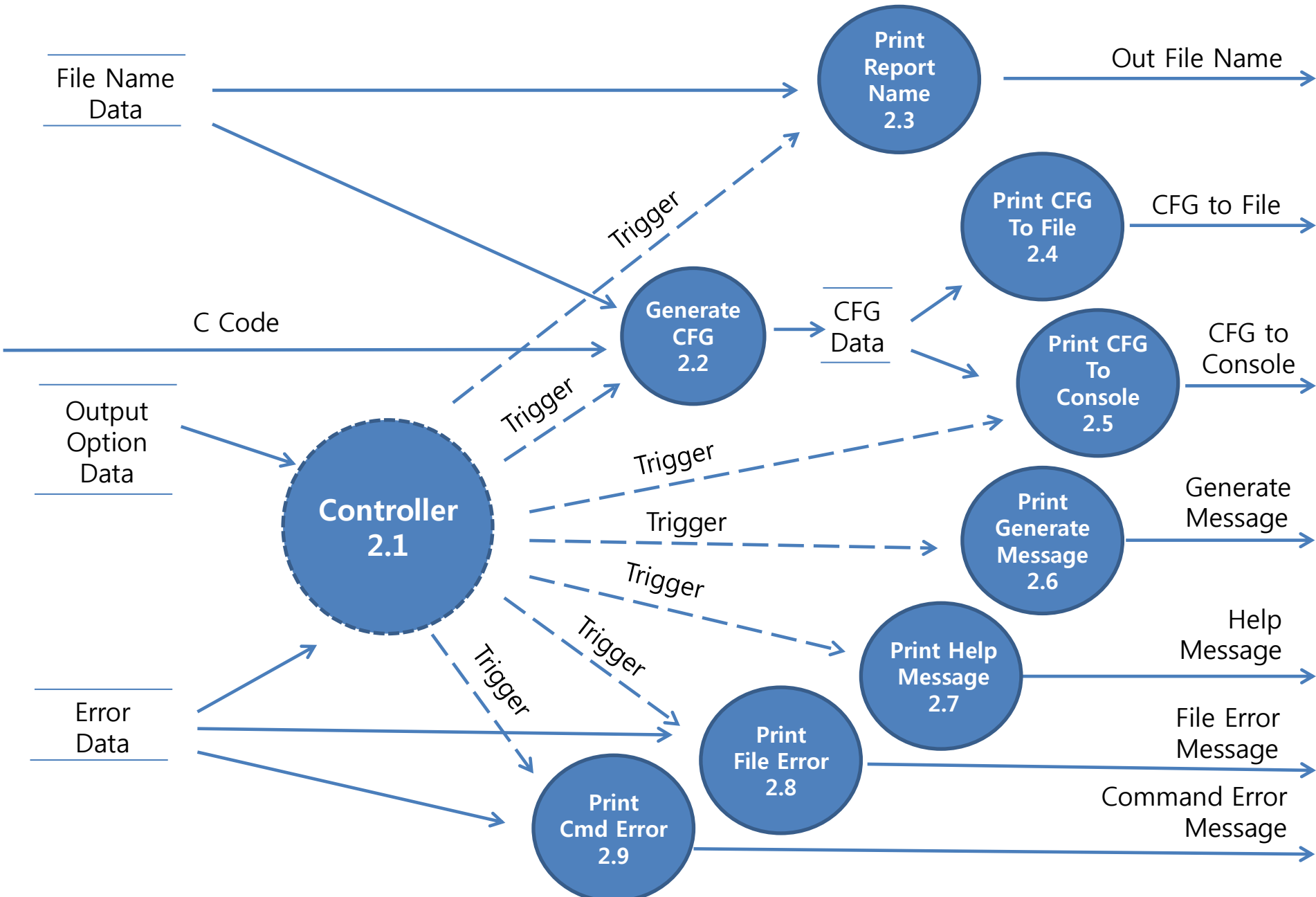
DFD Level 2



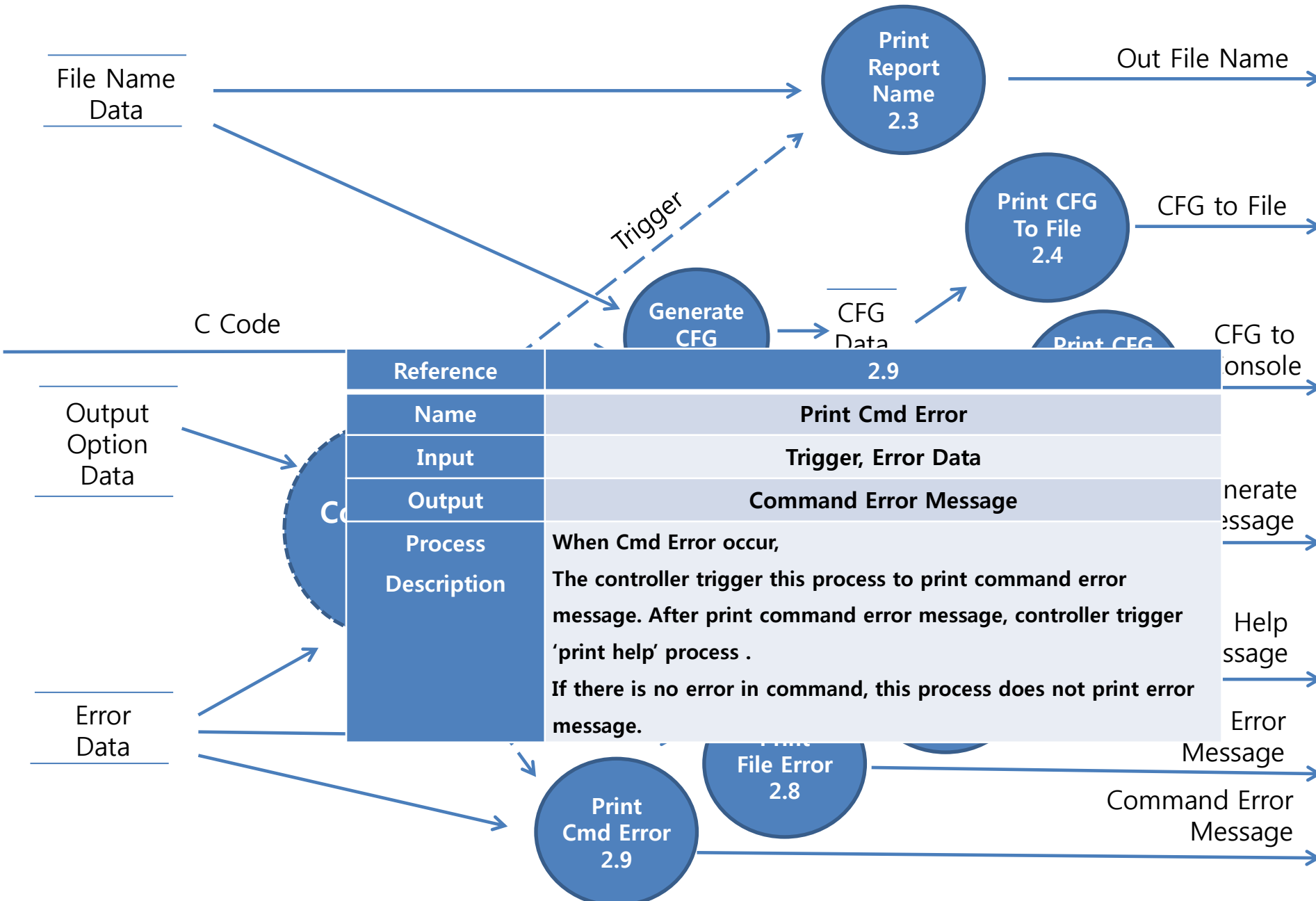
DFD Level 2



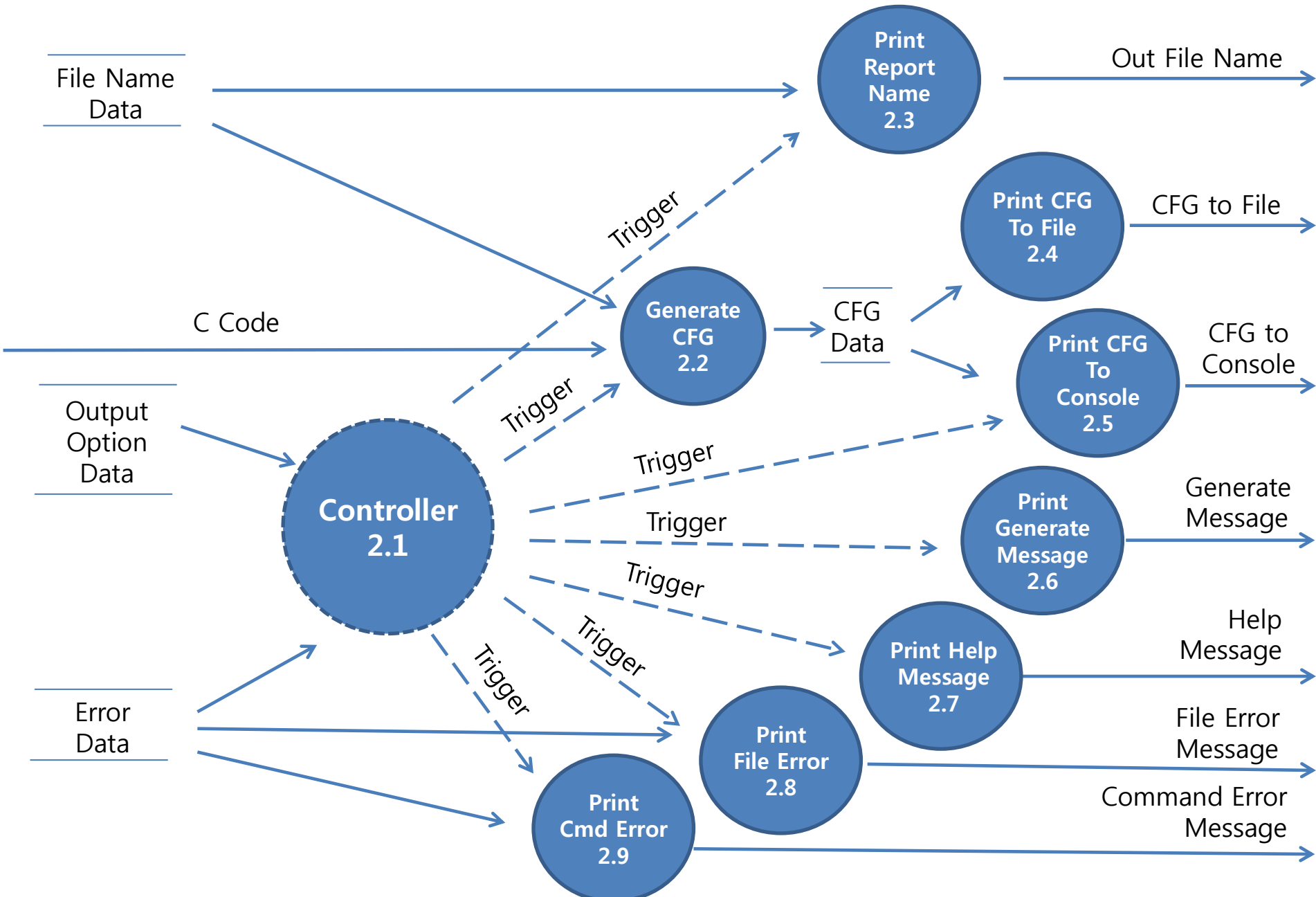
DFD Level 2



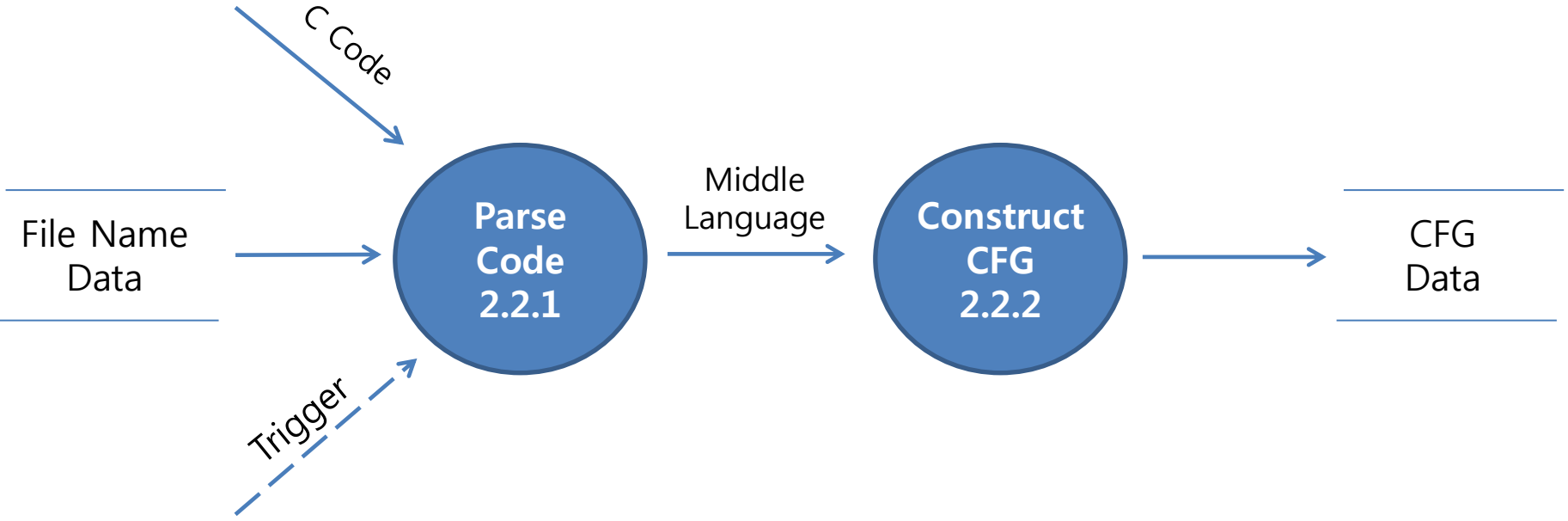
DFD Level 2



DFD Level 2



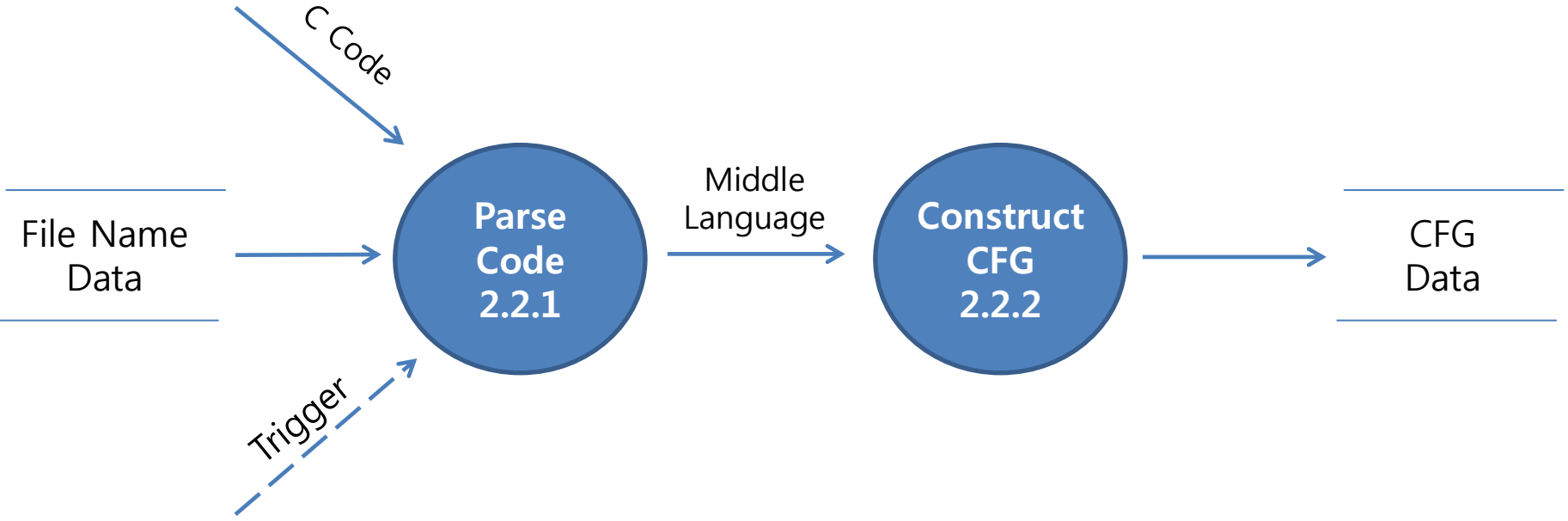
DFD Level 3



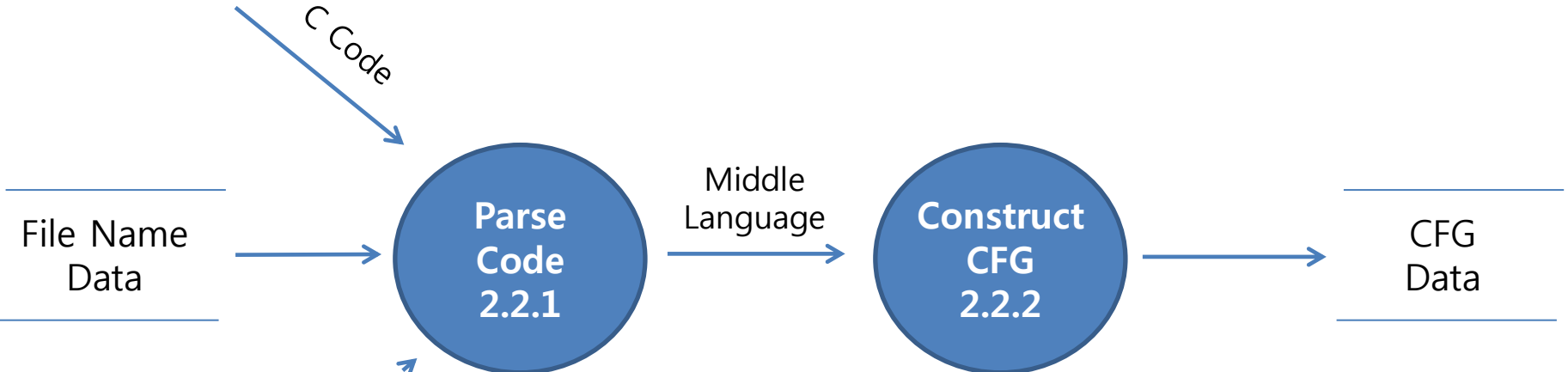
Data Dictionary Level 3

Input/Output Event	Description	Format/Type
Middle Language	<p>Parsing과정을 거쳐 생성된 Mid-Language 를 저장한다.</p> <p>파싱과정에서 생긴 중간적인 언어로서 소스코드를 분석하여 CFG component를 생성하기 위한 Data이다. 각 분기별 일정 정보를 struct형태의 list또는 배열로 저장</p>	Struct List / Array

DFD Level 3



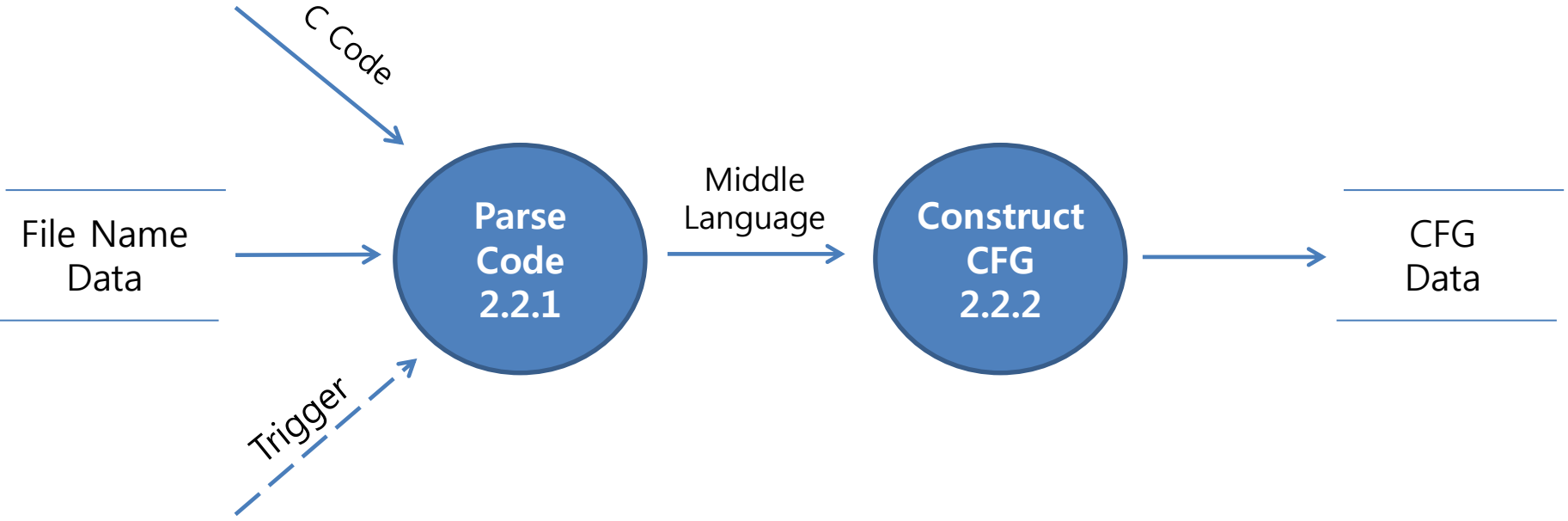
DFD Level 3



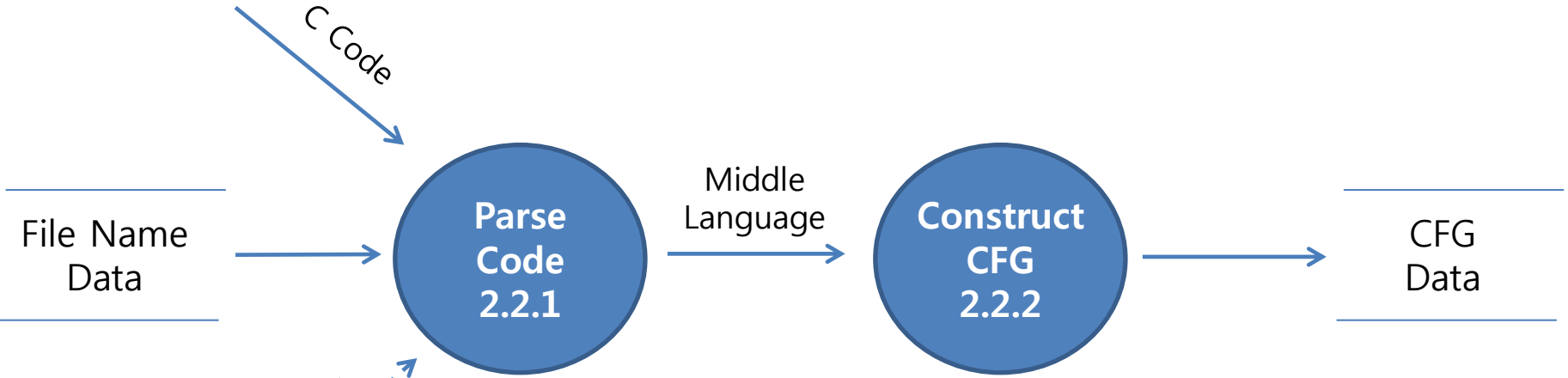
Trigger

Reference	2.2.1
Name	Parse Code
Input	Trigger, File Name data, C Code
Output	Middle Language
Process Description	When Error isn't occurred, this process read 'C code' from File. This process translates C code to middle language, reading each line of c code.

DFD Level 3

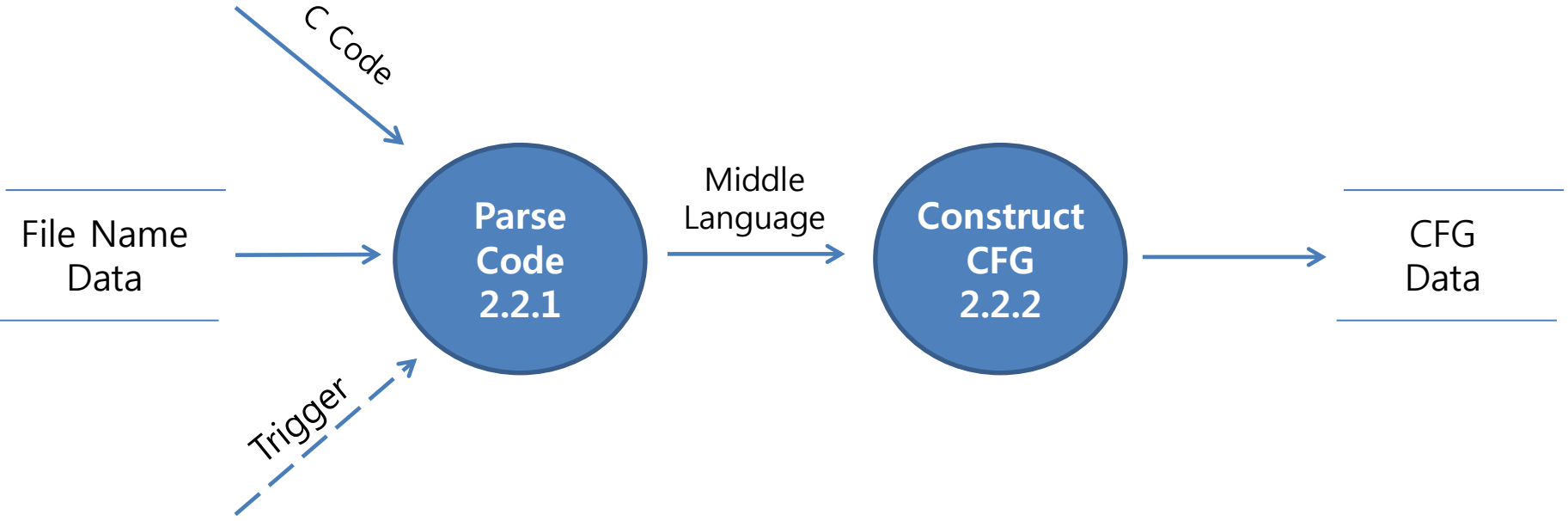


DFD Level 3

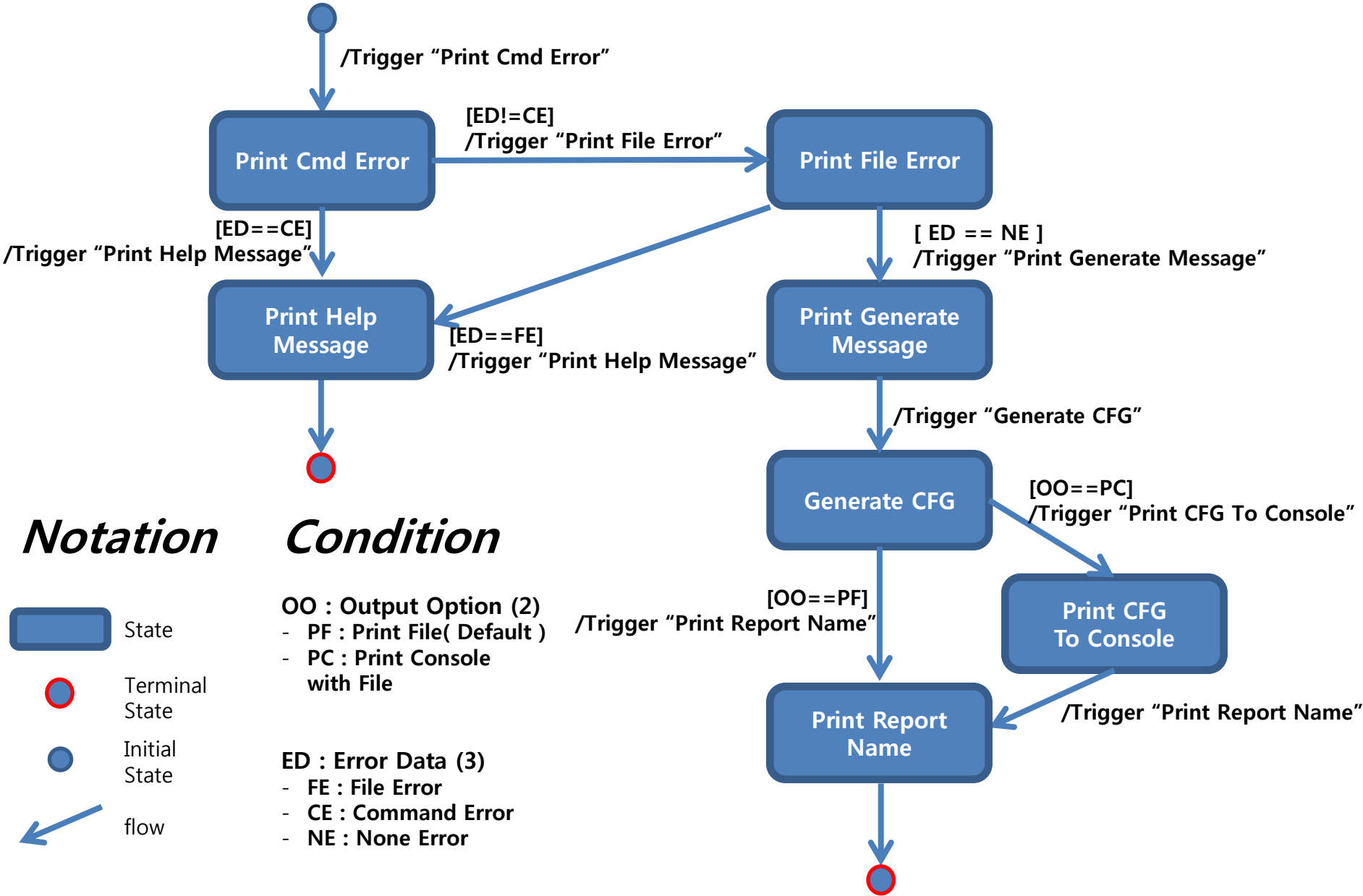


Reference	2.2.2
Name	Construct CFG
Input	Middle Language
Output	CFG Data
Process Description	<p>After Middle Language is generated, This process construct CFG with middle language. Finishing the constructing, this process save CFG Data for print report and console.</p>

DFD Level 3







DFD Level 4



Notation

Condition

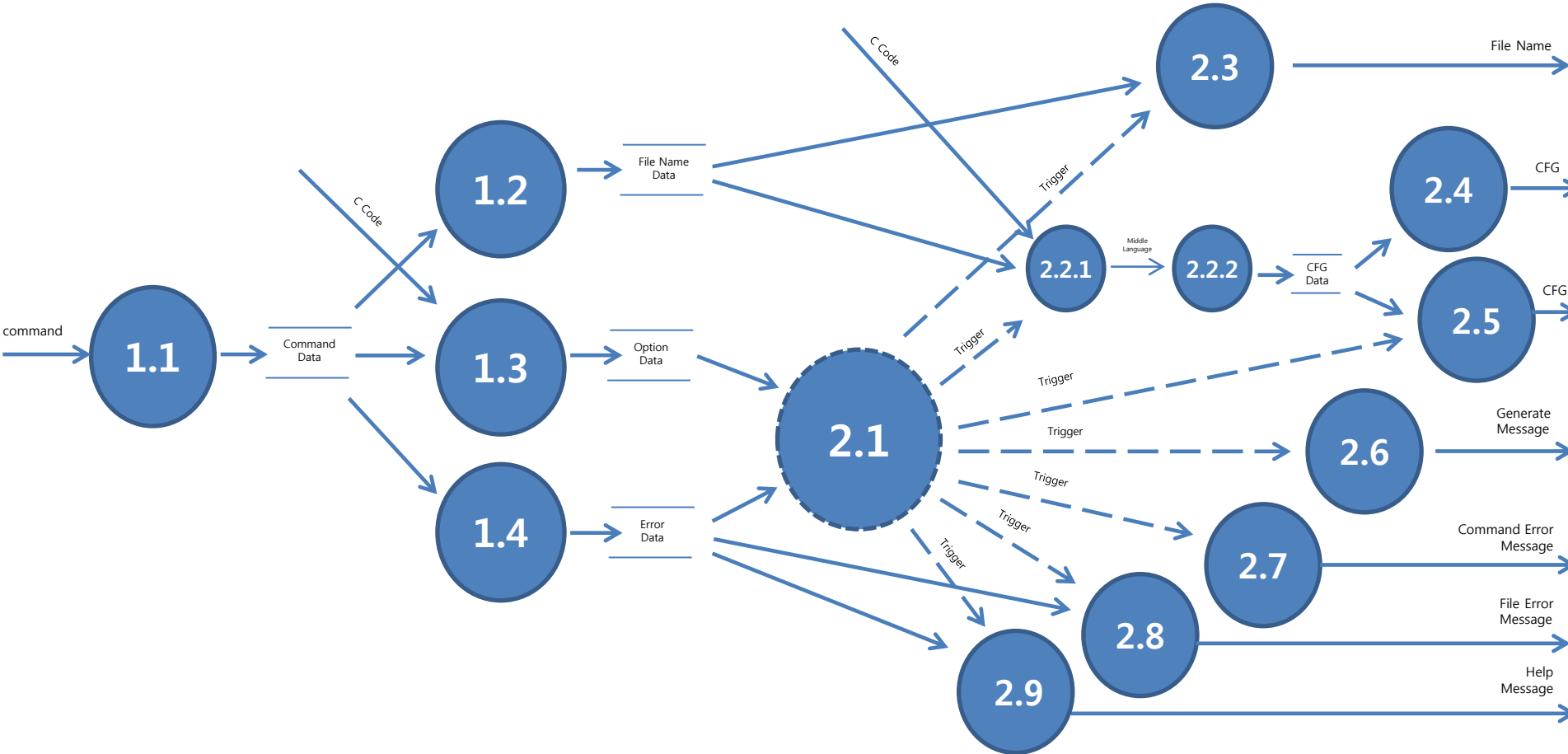
-  State
-  Terminal State
-  Initial State
-  flow

- OO : Output Option (2)**
- PF : Print File(Default)
 - PC : Print Console with File
- ED : Error Data (3)**
- FE : File Error
 - CE : Command Error
 - NE : None Error

/Trigger "Print Report Name"

/Trigger "Print Report Name"

Data Flow Diagram





Thank Yoo

and us

Process Specification

Process Specification Level 2

Reference	2.1
Name	Controller
Input	Error Data, Output Option Data
Output	Trigger
Process Description	"Error Data" and "Output Option data" come in this process for divergence. Controller can make a decision with that data for process of Program. The Operation of this Process is represented to "state machine" specifically.

Process Specification Level 2

Reference	2.3
Name	Print Report Name
Input	Trigger, File Name Data
Output	File Name
Process Description	<p>After Triggered, this process print report name as meaning of finishing print CFG and succeeding this program.</p> <p>Since this process can't take report name to controller, it take report name to File Name Data</p>

Process Specification Level 2

Reference	2.4
Name	Print CFG To File
Input	Trigger, CFG Data
Output	CFG
Process Description	After Triggered, this process, "Print CFG To File" print CFG Data to File. Since CFG go out two stream, File and Console, this process is named 'Print CFG To File' not 'Print CFG'.

Process Specification Level 2

Reference	2.5
Name	Print CFG To Console
Input	Trigger , CFG Data
Output	CFG
Process Description	<p>After Triggered, this process ,‘Print CFG To Console’, print CFG Data to Console.</p> <p>This process operate selectively, the contrary “Print CFG To File” operate unconditionally after generate CFG.</p>

Process Specification Level 2

Reference	2.6
Name	Print Generate Message
Input	Trigger
Output	Generate Message
Process Description	After Triggered, this process print Generate Message meaning of 'No Error' and starting 'Generate CFG Data'

Process Specification Level 2

Reference	2.7
Name	Print Cmd Error
Input	Trigger
Output	Command Error Message
Process Description	<p>When Cmd Error occur, The controller trigger this process to print command error message. After print command error message, controller trigger 'print help' process . If there is no error in command, this process does not print error message.</p>

Process Specification Level 2

Reference	2.8
Name	Print File Error
Input	Trigger
Output	File Error Message
Process Description	<p>When File Error occur, The controller trigger this process to print file error message. After print command error message, controller trigger 'print help' process. If there is no error in command, this process does not print error message.</p>

Process Specification Level 2

Reference	2.9
Name	Print Help Message
Input	Trigger
Output	Help Message
Process Description	<p>Whenever Error occur, this process is triggered.</p> <p>This process help user to operate 'CFG Generator' correctly.</p> <p>After print help message, this program is terminated.</p>

Process Specification Level 3

Reference	2.2.1
Name	Parse Code
Input	Trigger, File Name data, C Code
Output	Middle Language
Process Description	<p>When Error isn't occurred, this process read 'C code' from File.</p> <p>This process translates C code to middle language, reading each line of c code.</p>

Process Specification Level 3

Reference	2.2.2
Name	Construct CFG
Input	Middle Language
Output	CFG Data
Process Description	After Middle Language is generated, This process construct CFG with middle language. Finishing the constructing, this process save CFG Data for print report and console.