

# Structured Analysis Control Flow Graph (CFG) Generator

*A반 T2 - 김우빈 (201011321)  
임국현 (201011358)  
박대규 (201011329)*

# 목차 (SA)

- Environmental Model
  - ⊙ • Statement of Purpose
  - ⊙ • System Context Diagram
  - ⊙ • Event List
- Behavioral Model
  - ⊙ • Data Flow Diagram (DFD)
  - ⊙ • Data Dictionary (DD)
  - ⊙ • Process Specification

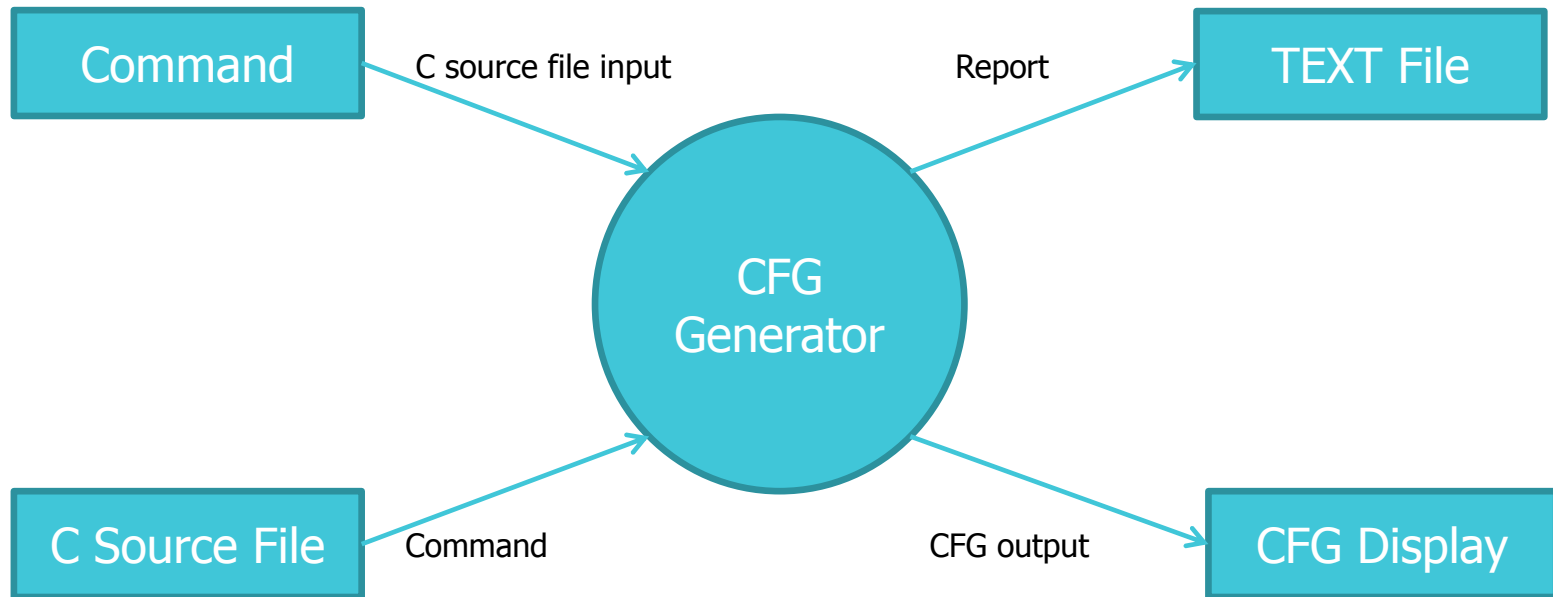
# 목차 (SD)

- Structure Charts
  - ⊙ • Transform Analysis
  - ⊙ • Notation
  - ⊙ • Structured Charts (CFG Generator)

# Statement Of Purpose

- 이 프로그램은 C소스 코드를 받아 CFG(Control Flow Graph)를 그리는 프로그램입니다.
- C소스 코드를 불러와 소스를 사용하기 편하게 처리를 하고 약간의 문법적 오류나 다른 헤더파일 사용을 체크합니다.
- 처리된 소스코드를 한 문장씩 읽어와 State와 Edge들을 생성해 냅니다.
- 각 처리단계 및 State와 Edge의 생성 성공/실패를 나타내어 줍니다. 만약 실패상태가 된다면 프로그램을 종료합니다.
- 이 프로그램의 핵심 요소는 소스코드를 분석하는 알고리즘 설계 그리고 분석된 내용을 CFG로 그려내는 것 입니다.

# System Context Diagram

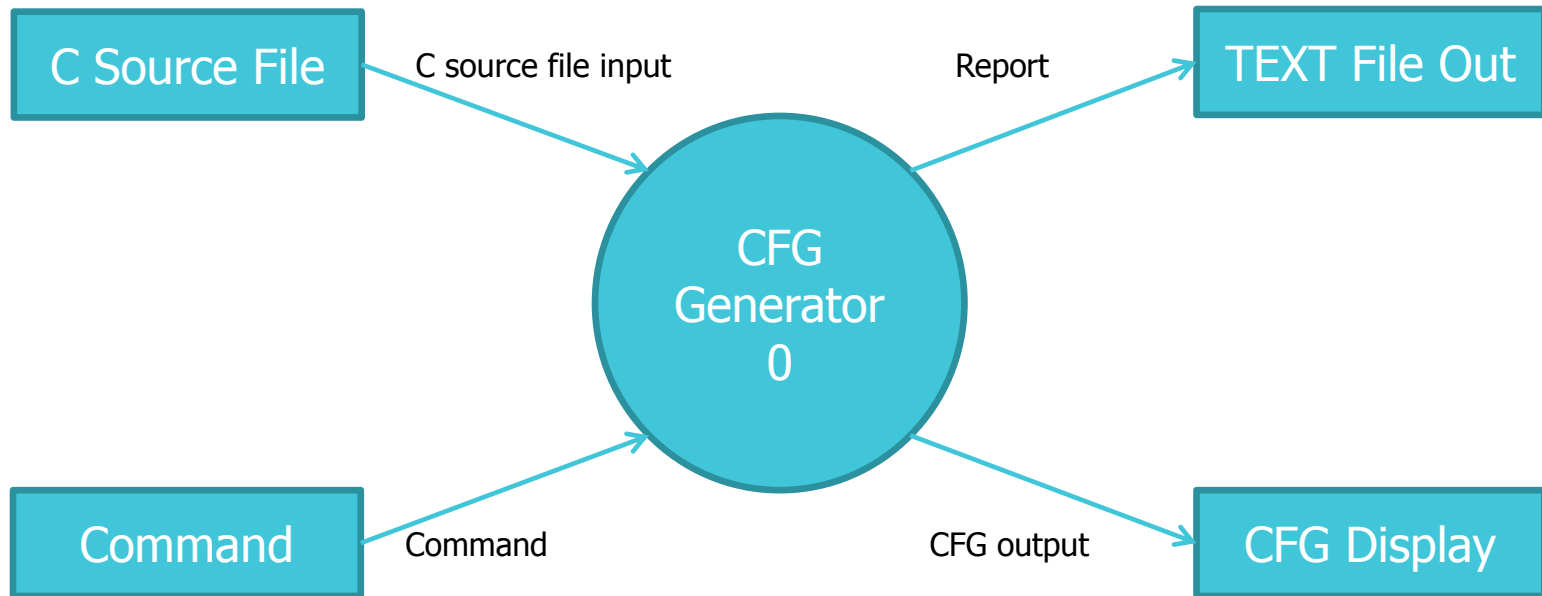


# System Context Diagram

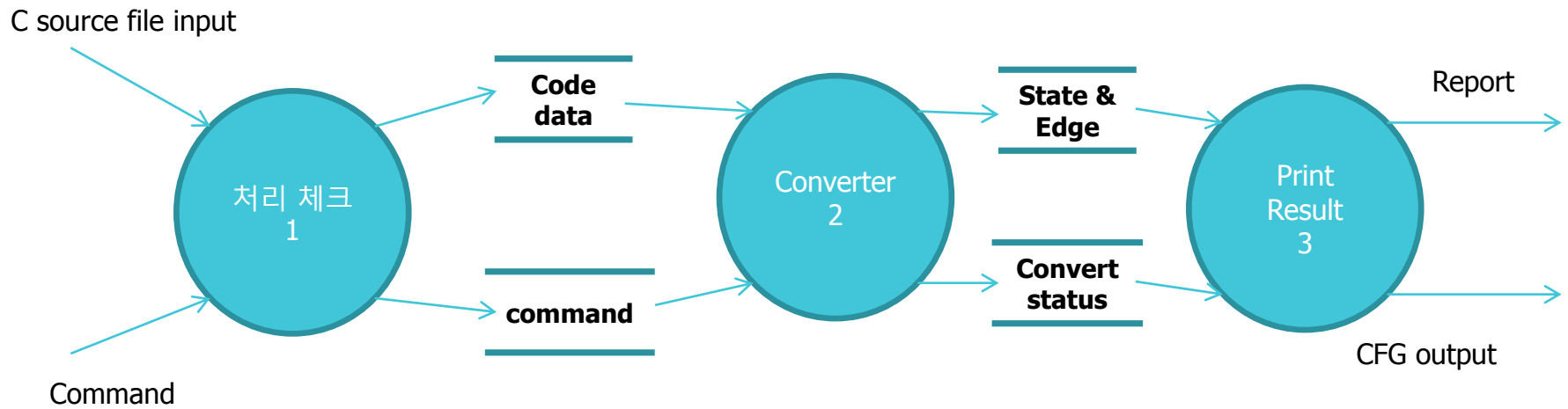
## -Event List-

Input/ Output Event	Description
Command	파일을 CFG로 만들기 위해 명령을 입력 받는다.
Report	변환된 파일의 데이터를 Txt 형태로 저장한다.
C source file input	CFG 로 바꿀 파일 이다.
CFG output	바뀐 CFG 파일의 display 출력이다.

# DFD – Level 0

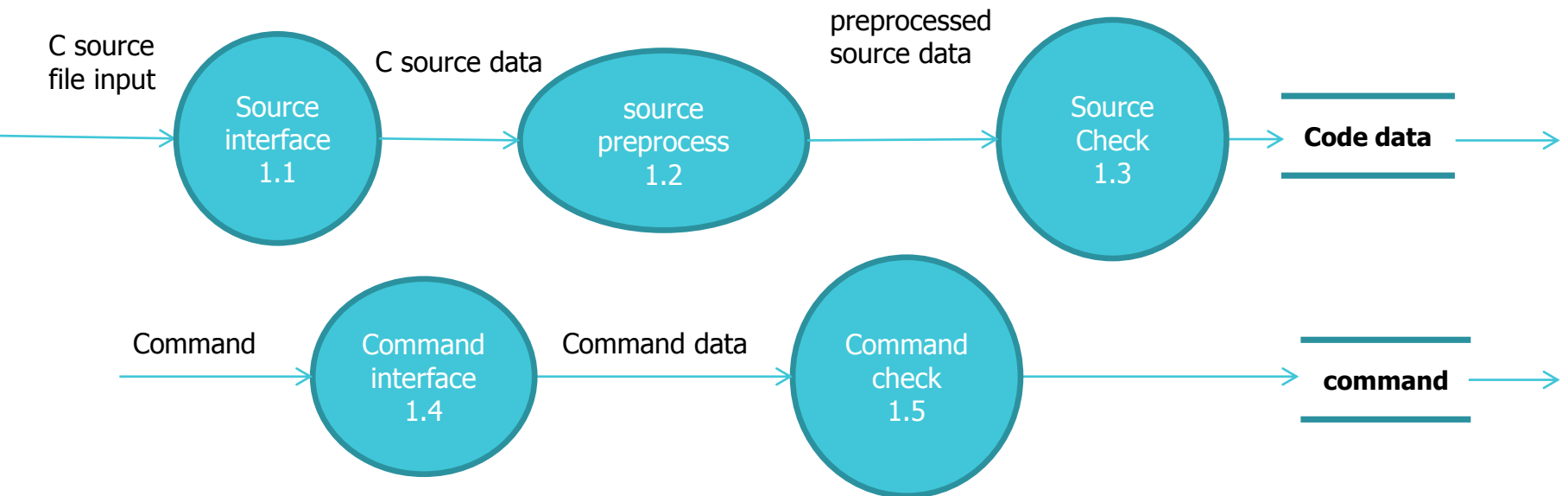


# DFD – Level 1

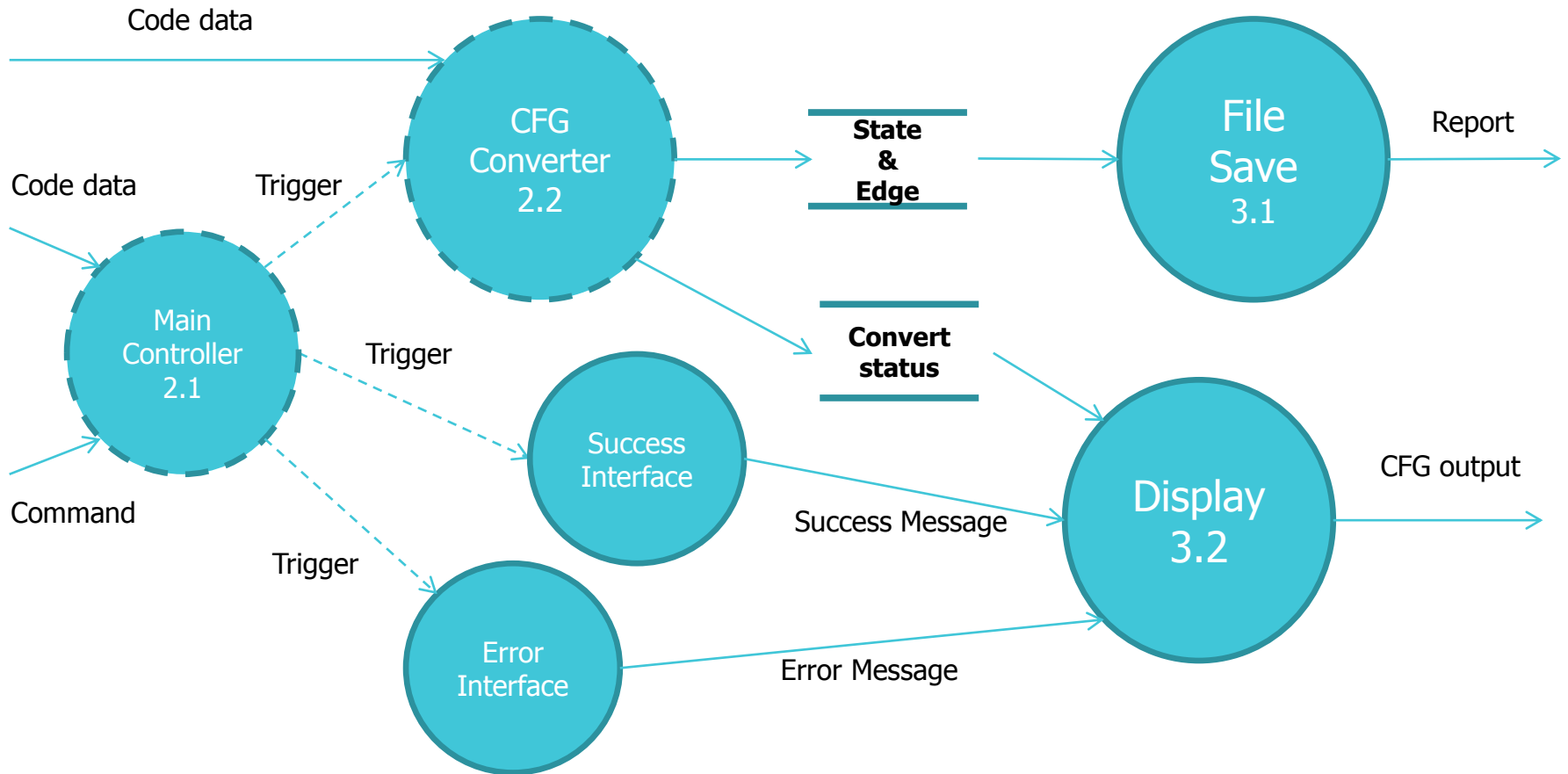




# DFD – Level 2



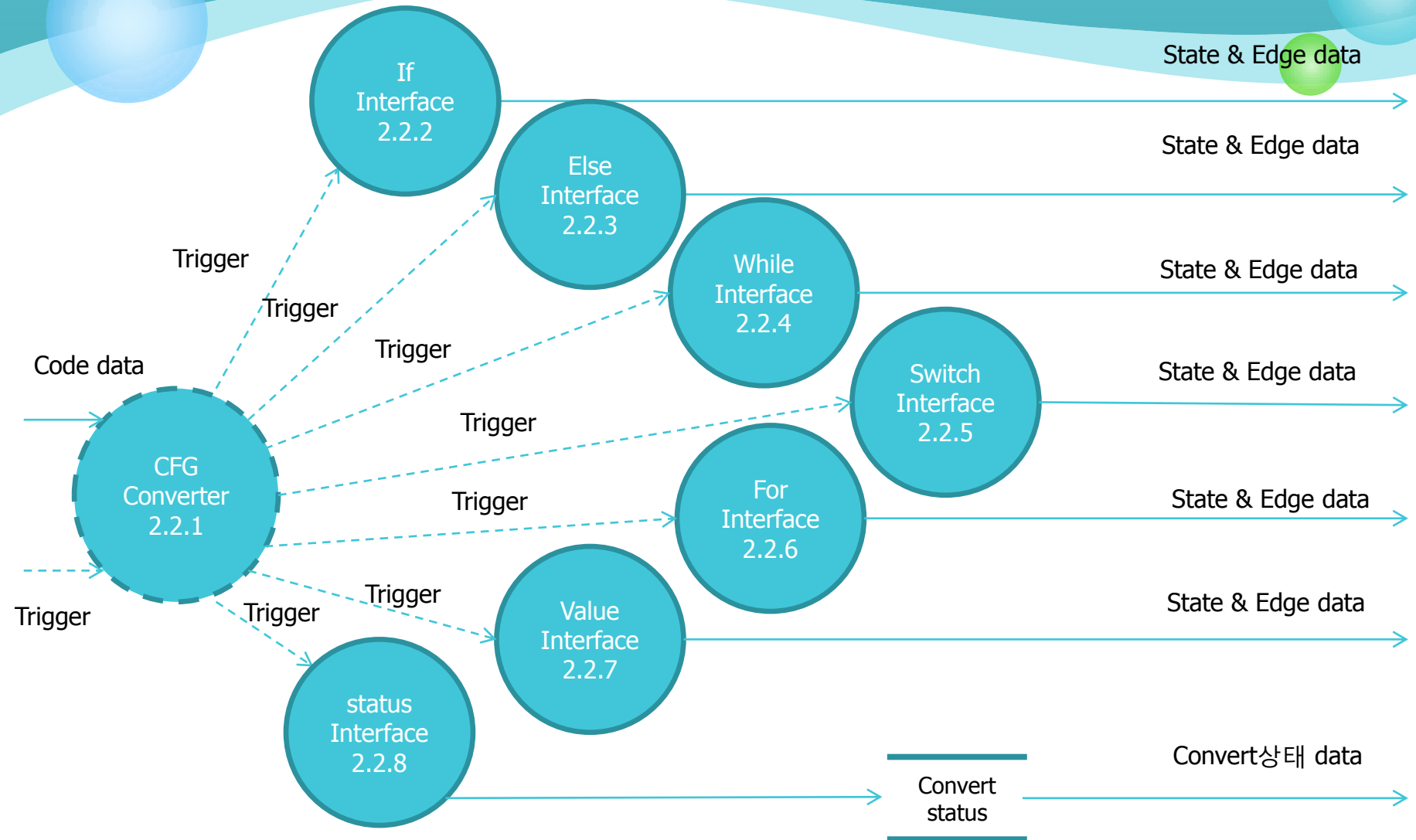
# DFD – Level 2



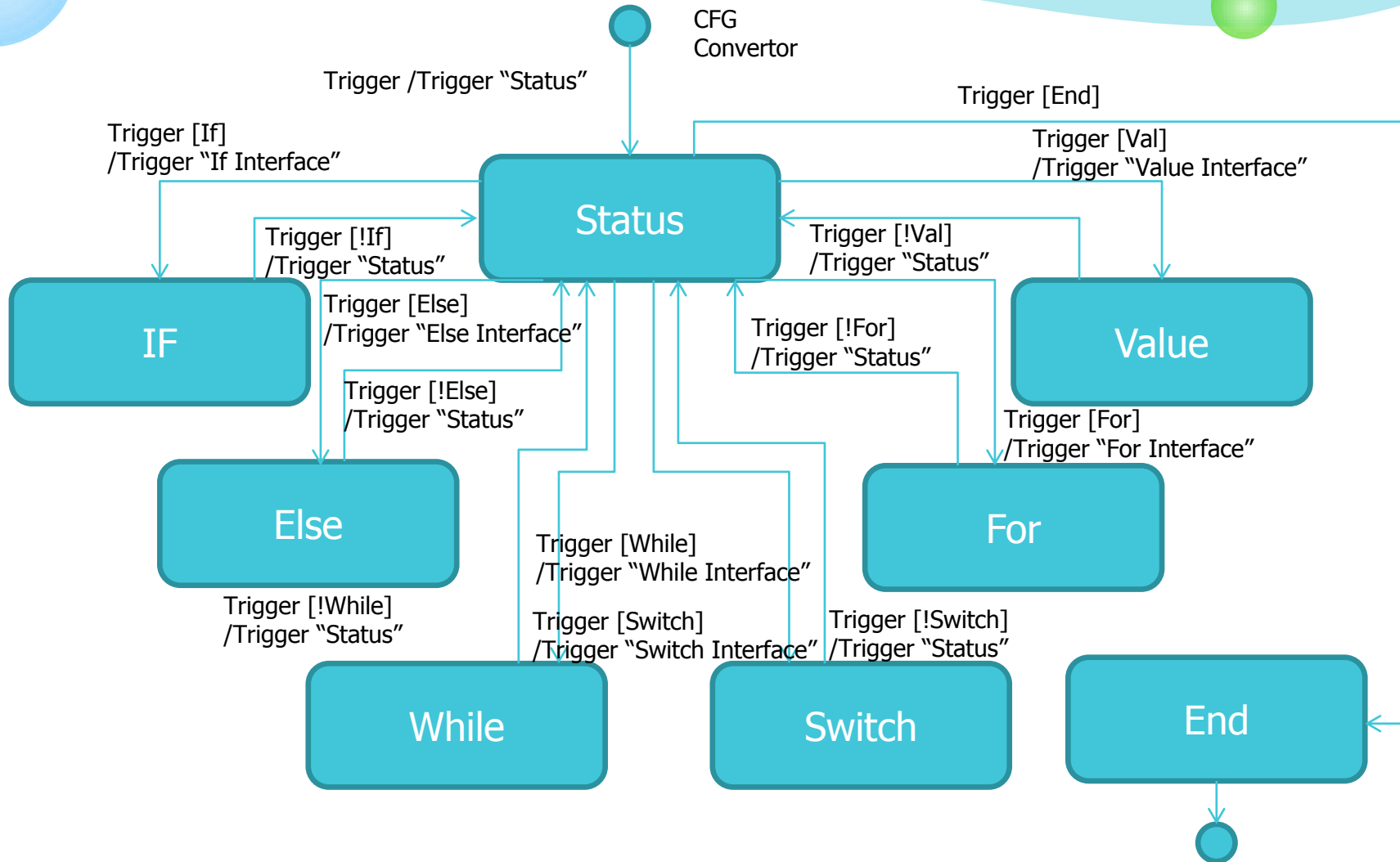
# Data Dictionary

Input/ Output Event	Description
C source data	입력된 source code의 data
Preprocessed Source data	Preprocess된 소스코드의 data.
Command data	입력 받은 명령의 data
Code data	Check 된 code data
command	Command에 대한 data
State & Edge	State와 Edge들의 data
Convert status	Convert상태를 저장한 data

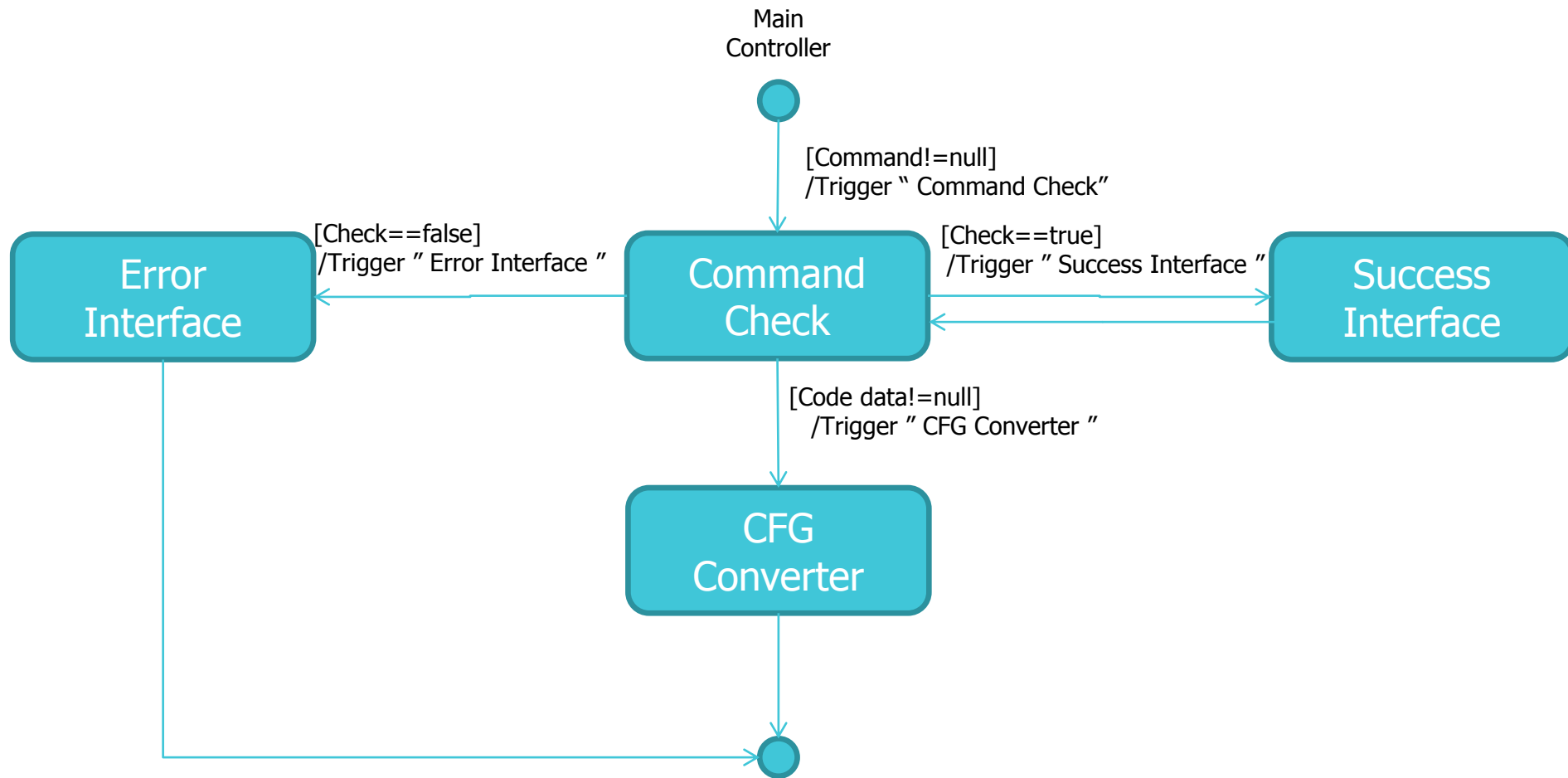
# DFD – Level 3



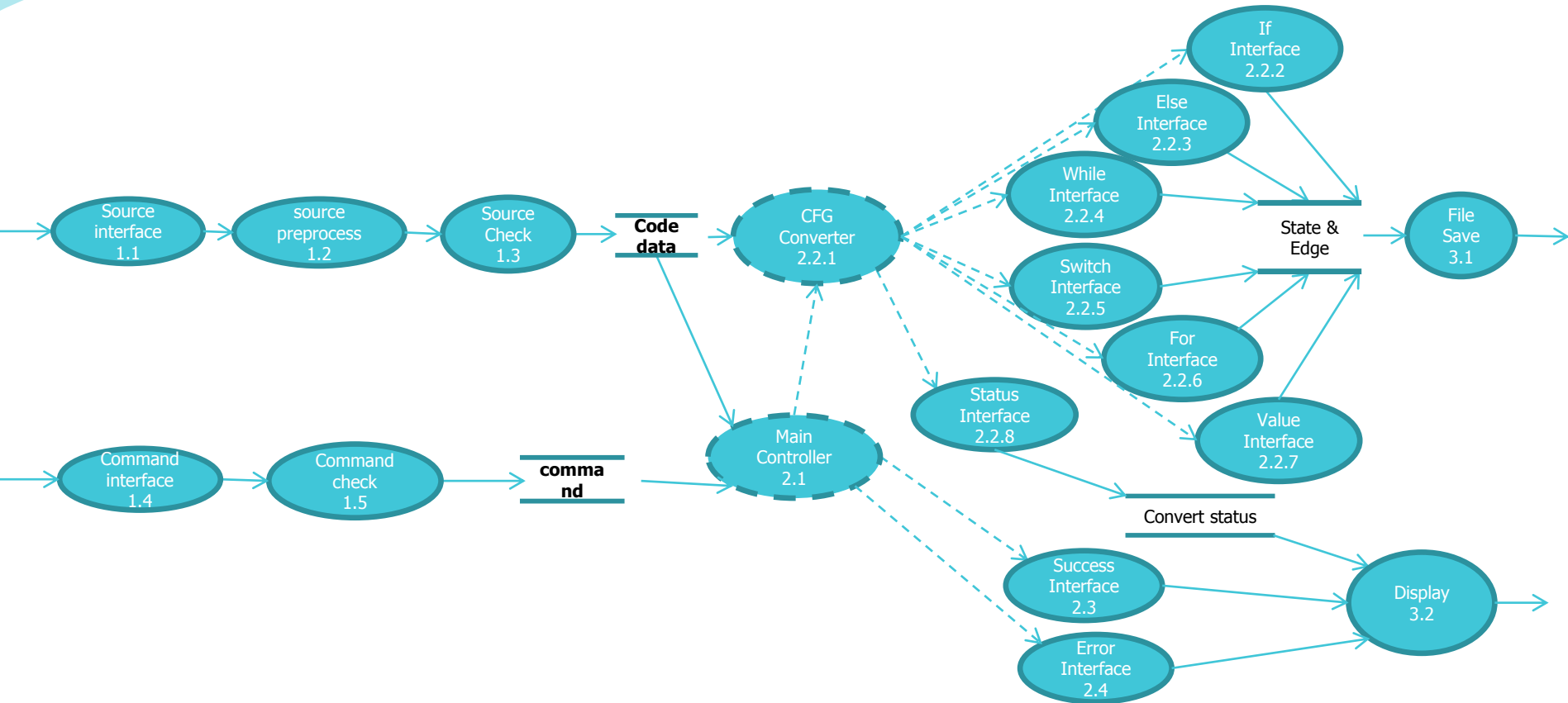
# DFD – Level 4



# DFD – Level 4



# DFD – AII



# Process Specification

<b>Reference No.</b>	1.1
<b>Name</b>	Source Interface
<b>Input</b>	C source file input
<b>Output</b>	C source data
<b>Process Description</b>	C source 파일을 받아 C source data로 바꾸어준다.

<b>Reference No.</b>	1.2
<b>Name</b>	source preprocess
<b>Input</b>	C source data
<b>Output</b>	preprocessed source data
<b>Process Description</b>	C source data에서 main을 제외한 함수 #define등을 main으로 넣어 전처리 하여준다.



# Process Specification

<b>Reference No.</b>	1.3
<b>Name</b>	Source Check
<b>Input</b>	preprocessed source data
<b>Output</b>	Code data
<b>Process Description</b>	전 처리된 소스에서 "stdio.h"이외의 헤더파일, 포인터의 유무등을 체크한다.

<b>Reference No.</b>	1.4
<b>Name</b>	Command interface
<b>Input</b>	Command
<b>Output</b>	Command data
<b>Process Description</b>	받은 명령을 data로 바꾸어준다.

# Process Specification

<b>Reference No.</b>	1.5
<b>Name</b>	Command check
<b>Input</b>	Command data
<b>Output</b>	Command
<b>Process Description</b>	Command가 제대로 된 command 인지 체크한다.

<b>Reference No.</b>	2.1
<b>Name</b>	Main Controller
<b>Input</b>	Code data, command
<b>Output</b>	Trigger
<b>Process Description</b>	Code data와 command 를 받아 프로세스들에 적절히 명령을 내린다.

# Process Specification

<b>Reference No.</b>	2.2.1
<b>Name</b>	CFG Converter
<b>Input</b>	Trigger, Code data
<b>Output</b>	Trigger
<b>Process Description</b>	Code data에 대해 적절한 프로세스로 명령을 내린다.

<b>Reference No.</b>	2.2.2
<b>Name</b>	If Interface
<b>Input</b>	Trigger
<b>Output</b>	State & Edge data
<b>Process Description</b>	If 구문을 state와 edge를 적절하게 만든다.

# Process Specification

<b>Reference No.</b>	2.2.3
<b>Name</b>	Else Interface
<b>Input</b>	Trigger
<b>Output</b>	State & Edge data
<b>Process Description</b>	Else 구문을 state와 edge를 적절하게 만든다.

<b>Reference No.</b>	2.2.4
<b>Name</b>	While Interface
<b>Input</b>	Trigger
<b>Output</b>	State & Edge data
<b>Process Description</b>	While 구문을 state와 edge를 적절하게 만든다.

# Process Specification

<b>Reference No.</b>	2.2.5
<b>Name</b>	Switch Interface
<b>Input</b>	Trigger
<b>Output</b>	State & Edge data
<b>Process Description</b>	Switch 구문을 state와 edge를 적절하게 만든다.

<b>Reference No.</b>	2.2.6
<b>Name</b>	For Interface
<b>Input</b>	Trigger
<b>Output</b>	State & Edge data
<b>Process Description</b>	For 구문을 state와 edge를 적절하게 만든다.

# Process Specification

<b>Reference No.</b>	2.2.7
<b>Name</b>	Value Interface
<b>Input</b>	Trigger
<b>Output</b>	State & Edge data
<b>Process Description</b>	변수 값 입력에 대해 state와 edge를 적절하게 만든다.

<b>Reference No.</b>	2.2.8
<b>Name</b>	Status Interface
<b>Input</b>	Trigger
<b>Output</b>	Convert status
<b>Process Description</b>	현재 동작상태를 data로 만든다.

# Process Specification

<b>Reference No.</b>	2.3
<b>Name</b>	Success Interface
<b>Input</b>	Trigger
<b>Output</b>	Success response
<b>Process Description</b>	파일입력의 성공을 검사한다.

<b>Reference No.</b>	2.4
<b>Name</b>	Error Interface
<b>Input</b>	Trigger
<b>Output</b>	Error response
<b>Process Description</b>	파일입력의 오류를 검사한다.

# Process Specification

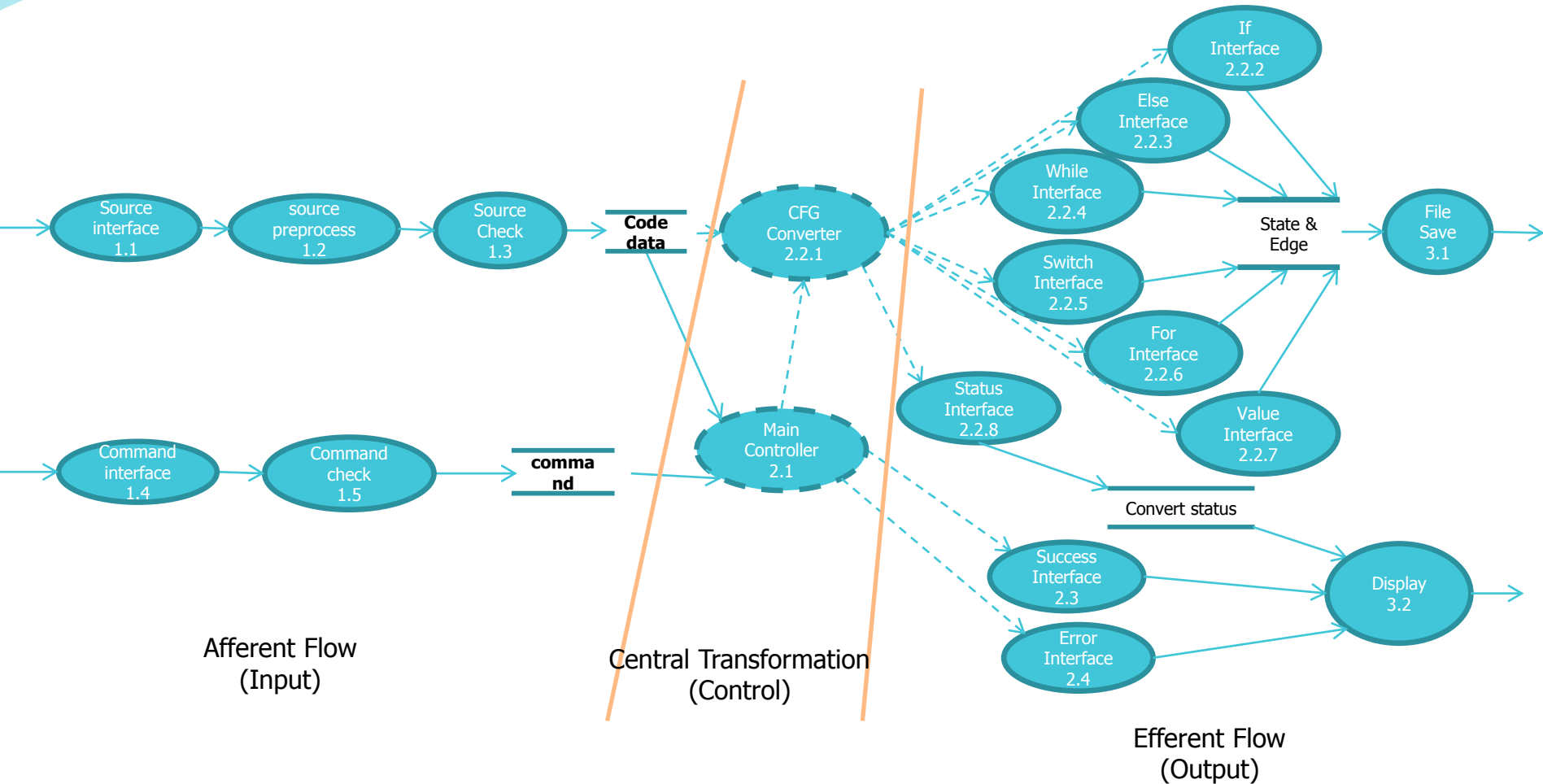
<b>Reference No.</b>	3.1
<b>Name</b>	File Save
<b>Input</b>	State & Edge data
<b>Output</b>	Report
<b>Process Description</b>	저장된 State와 Edge데이터를 File로 만들어 저장한다.

<b>Reference No.</b>	3.2
<b>Name</b>	Display
<b>Input</b>	Convert status, Success response, Error response
<b>Output</b>	
<b>Process Description</b>	파일입력의 성공/실패와 변환상태를 출력한다.



# Structured Charts

## - Transform Analysis -



# Structured Charts - Notation

## Basic Notation [Yourdon 1989]



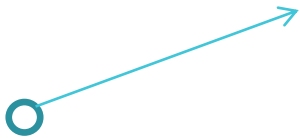
Modules



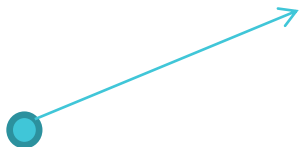
Library modules



Module call



Data Flow

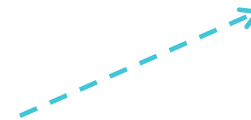


Control Flow

## Variations



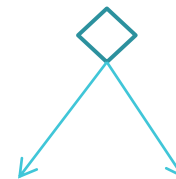
Data module



Asynchronous  
Module call

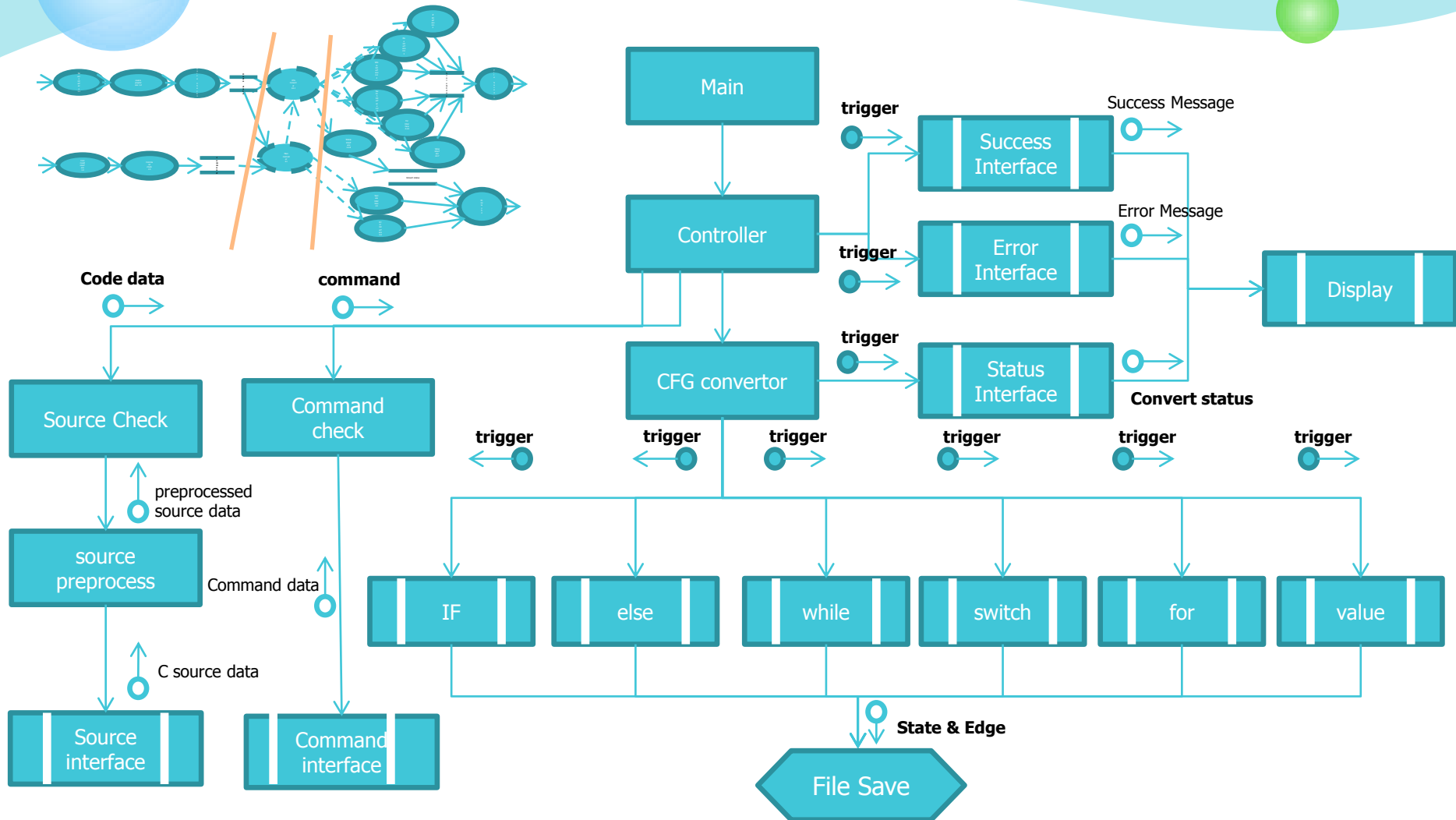


Iteration



Decision

# Structured Charts – CFG Generator (basic)



# Structured Charts – CFG Generator (advanced)

