

2011-2 <소프트웨어공학개론> Team Project #3.

# CFG Generator SA

Class A - T1

발표자 - 201011342안혜수

조원 - 201011309권선일, 201011336백인선,  
201011357이주희,

# Structured Analysis

---

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



# Statement of Purpose

---

- 주어진 C언어 파일을 받아 해당 소스코드의 CFG를 생성
- Main()에 있는 코드를 대상
- 코드 분류를 크게 General Operation과 Control Statement로 분류하여 처리
- CFG Block은 독립적인 Block과 Control Statement에 따라 나타나는 분기 Block으로 분류하여 처리
  - 처리할 수 없는 형태의 명령어나 main이외의 코드, 구조체나 포인터 등에 대해선 처리하지 못한다.
  - 허용 라인 수는 100줄 이내로 제한

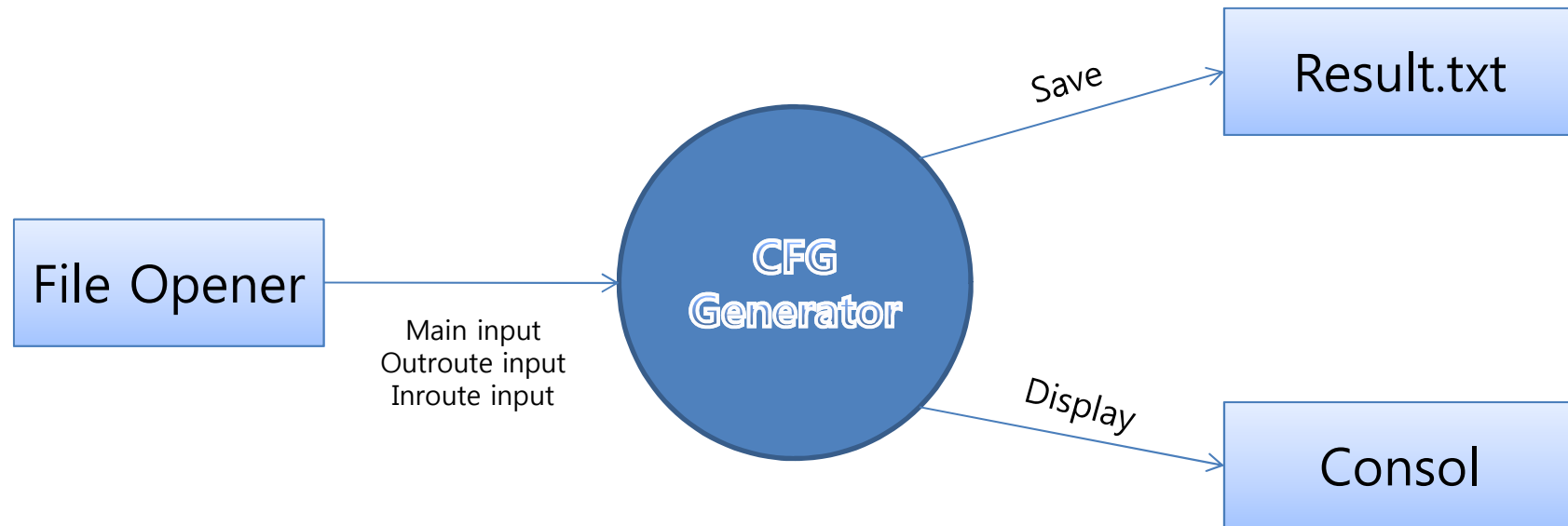
# Statement of Purpose

---

- 소스코드의 논리적 오류는 감지하지 못한다.
- 입력된 C코드가 부적합하거나 변환에 실패했을 시 오류사항, 변환에 성공했을 시 생성 파일 명을 사용자가 볼 수 있도록 출력한다.
- 생성된 결과는 .txt로 저장된다.
  - 독립적인 Block은 "#Block\_Number(0,1,2,...)+Contents"
  - 분기 Block은 "Block\_Number(0.1, 0.2, 1.1, 1.2, 1.3, ...)+Contents" 형태로 저장
  - Edge는 (pre\_Block\_Number, post\_Block\_Number) 형태로 저장
  - 최종 Block개수와 Edge개수

# System Context Diagram

---



# Event List

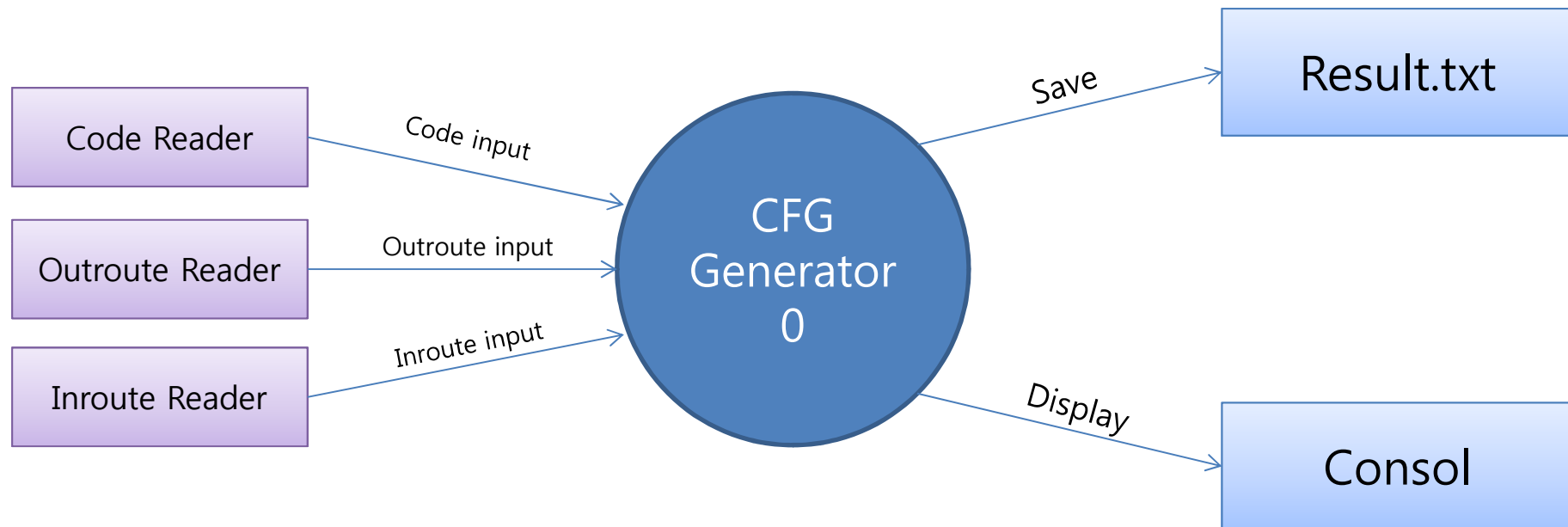
---

Input Event	Description
Code input	CFG로 변환할 .C 파일의 Source Code
Outroute input	Report가 저장될 파일의 경로
Inroute input	코드를 읽어올 파일의 경로

Output Event	Description
Save	CUI형태의 변환된 CFG를 저장 경로에 저장
Display	사용자가 처리사항을 볼 수 있도록 consoli화면에 진행결과 출력

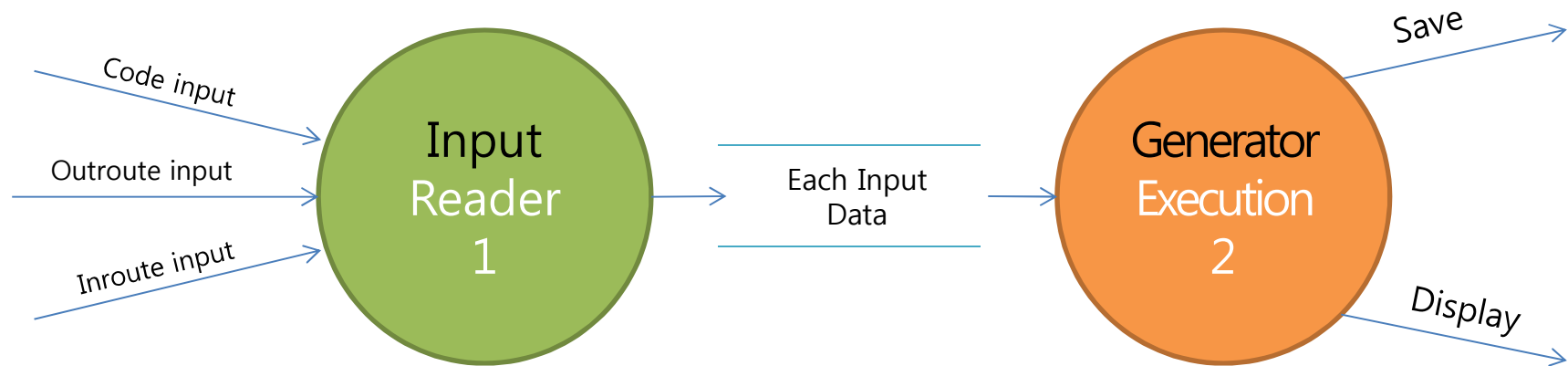
# Data Flow Diagram – Level 0.

---



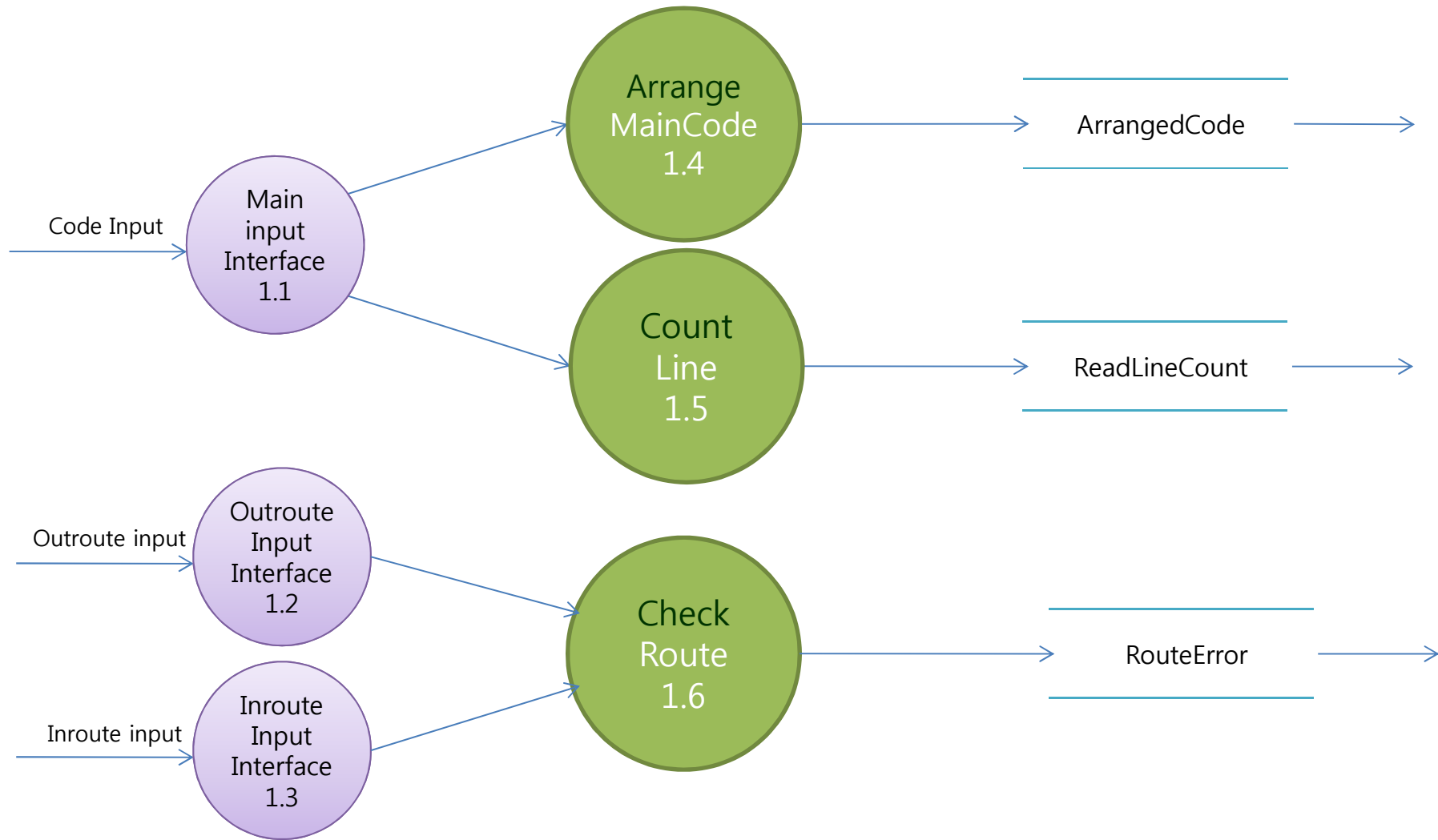
# Data Flow Diagram – Level 1.

---





# Data Flow Diagram – Level 2.



# Process Specification

Name		1.1 Code input Interface
Input	Code input	
Output	Original Source Code	
Process Description	사용자가 입력한 변환 하고자 하는 "Code input"을 디지털 신호로 변환 후 전달한다.	

Name		1.2 Outroute input Interface
Input	Outroute input	
Output	Outroute Suitability	
Process Description	파일 출력을 위해 입력된 경로가 유효한 경로인지 True/False의 디지털 신호로 변환후 전달	

Name		1.3 Inroute input Interface
Input	Inroute input	
Output	Inroute Suitability	
Process Description	파일 입력을 위해 입력된 경로가 유효한 경로인지 True/False의 디지털 신호로 변환후 전달	

# Process Specification (Cont.)

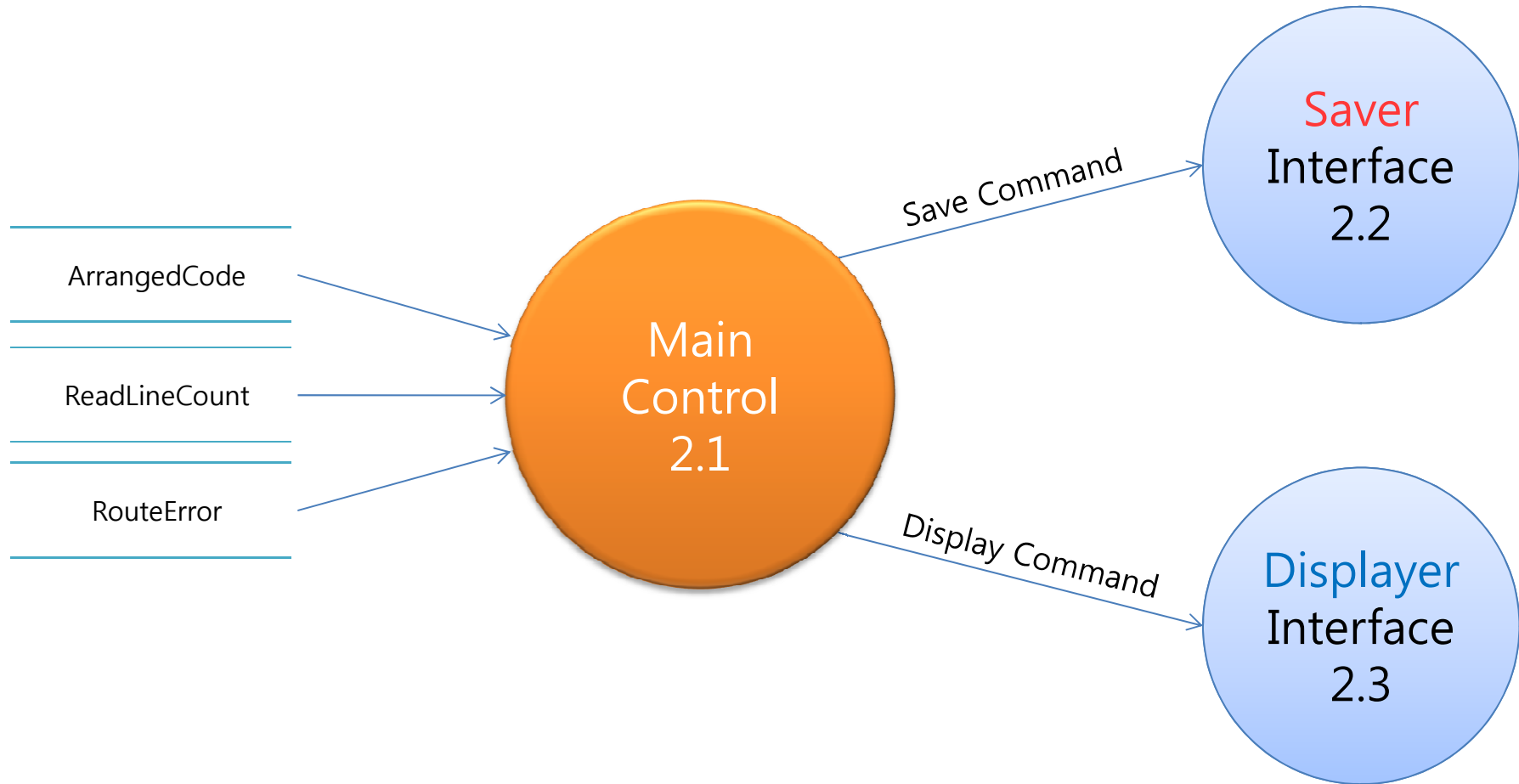
1.4 Arrange MainCode	
Input	Original Source Code
Output	Arranged Code
Process Description	Original Source Code의 main()함수내의 코드 중 처음부분의 변수선언을 제외한 Source code를 추출하여 CFG변환의 대상이 되는 부분만 추출하여 Arranged Code를 내보낸다.

1.5 Count Line	
Input	Original Source Code
Output	ReadLineCount
Process Description	';'을 단위로 Line을 판단하여 Main()내의 ';'를 카운트한다. Printf함수의 ()가로 내의 ';'는 카운트에서 제외한다.

1.6 Check Route	
Input	Outroute input, Inroute input
Output	RouteError
Process Description	실행 Command로 입력받은 In, Out File 경로의 유효성을 검사한다. 2개의 input중 하나라도 false 라면 false를 내보낸다.

# Data Flow Diagram – Level 2.

---

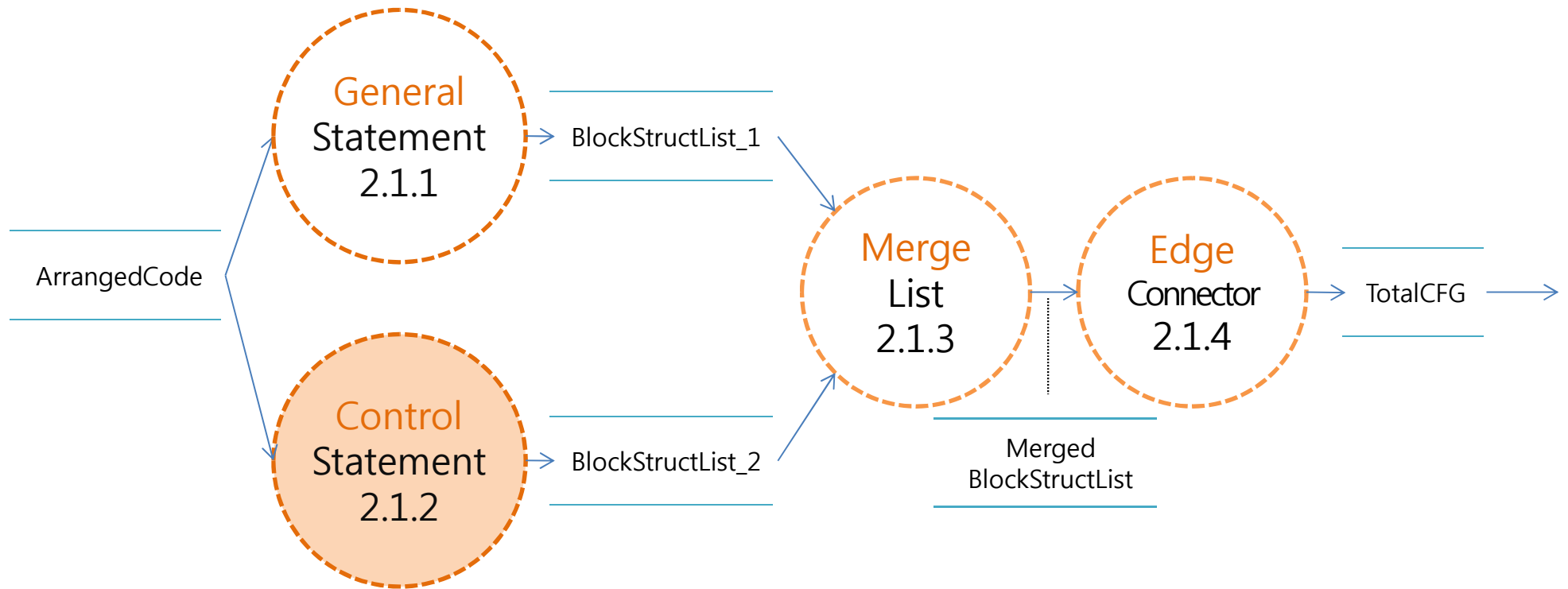


# Process Specification

2.1 main Control	
Input	ArrangedCode, ReadLineCount, RouteError
Output	Save Command, Display Command
Process Description	각 Data들을 전달 받아 RoutError와 LineOverError를 판단하고, CFG의 Block을 구조체 리스트로 작성하여, 최종 저장과 관련한 Save Command 와 변환결과를 출력할 Display Command를 산출한다.

2.2 Saver Interface		2.3 Displayer Interface	
Input	Saver Command	Input	Displayer Command
Output	Save	Output	Display
Process Description	Saver Command의 명령을 디지털 신호로 전환하여 Saver process에 전달한다.	Process Description	Displayer Command의 명령을 디지털 신호로 전환하여 Displayer Process에 전달한다.

# Data Flow Diagram – Level 3.



# Process Specification

Name	2.1.1 General Statement
Input	ArrangedCode
Output	BlockStructList_1 Data
Process Description	ArrangedCode를 ' ; ' 단위로 읽으며, Control Statement Keyword가 포함되지 않은 GeneralStatement를 StructList로 작성한다.

Name	2.1.2 Control Statement
Input	ArrangedCode
Output	BlockStructList_2 Data
Process Description	ArrangedCode를 ;단위로 읽으며, 정해진 Control Statement Keyword (If Else, Switch, for, While)가 포함된 ControlStatement를 StructList로 작성한다.

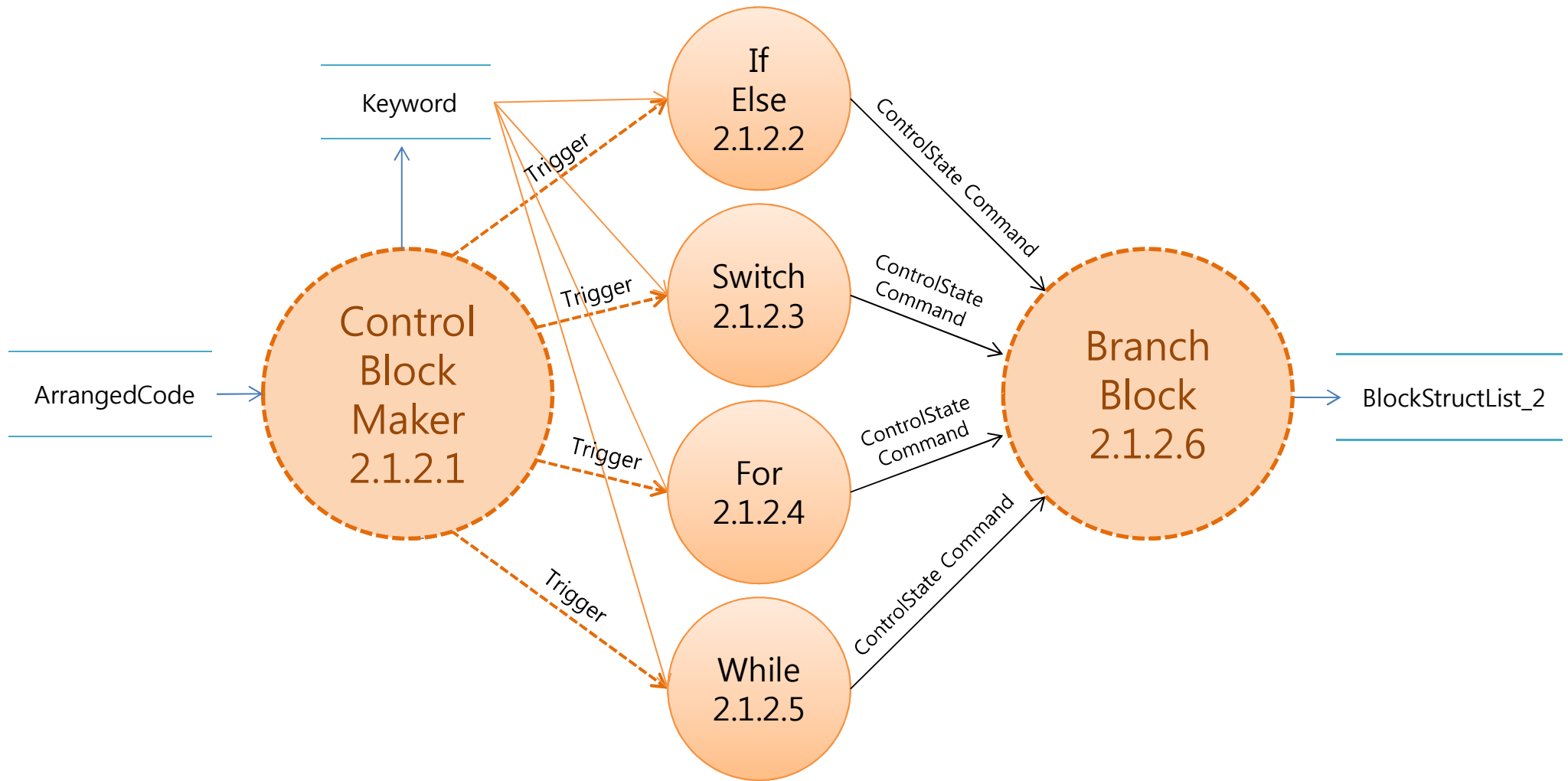
# Process Specification (cont.)

Name	2.1.3 List Merge
Input	BlockStructList_1, BlockStructList_2
Output	MergedBlockStructList
Process Description	입력 받은 BlockStructList1과 BlockStructList2를 Block의 번호에 맞게 정리하여 List를 정리하여 MergedStrustList를 내보낸다.

Name	2.1.4 Edge Connector
Input	MergedBlockStructList
Output	TotalCFG
Process Description	입력 받은 MergedBlockStructList를 통해 각각의 Block들 사이의 OutEdge 여부를 판단하여 그에 따라 알맞은 Edge를 생성하여 전체적인 CFG를 출력하여 준다.



# Data Flow Diagram – Level 4.



# Process Specification

Name	2.1.2.1 Control Block Maker
Input	ArrangedCode Data
Output	Keyword Data, Trigger
Process Description	입력 받은 ArrangedCode 에서 Keyword(If, Switch, For, While)를 판별하여 각각에 알맞은 Statement를 판단하는 processor로 보내준다.

Name	2.1.2.6 Branch Block
Input	ControlState Command
Output	BlockStructList_2
Process Description	ControlState Command 를 입력 받아 해당하는 분기 Block를 생성하여 BlockStructList2에 저장하여 준다.

# Process Specification (cont.)

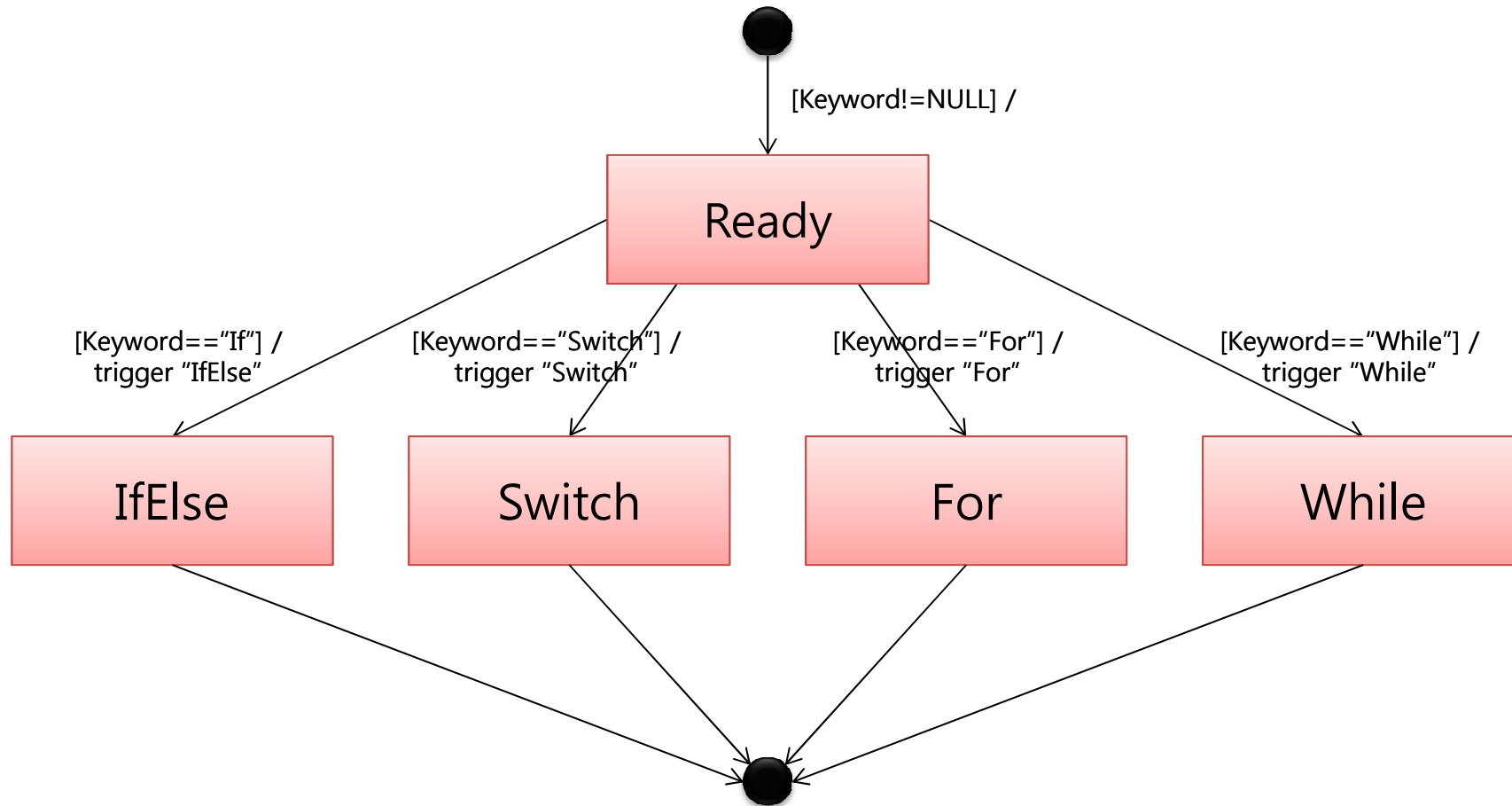
Name		2.1.2.2 If Else	Name		2.1.2.3 Switch
Input		Trigger, keyword	Input		Trigger, keyword
Output		ControlState Command	Output		ControlState Command
Process Description		입력 받은 keyword가 "IfElse"라면 그에 해당하는 Control State Command를 내려준다.	Process Description		입력 받은 keyword가 "Switch"라면 그에 해당하는 Control State Command를 내려준다.

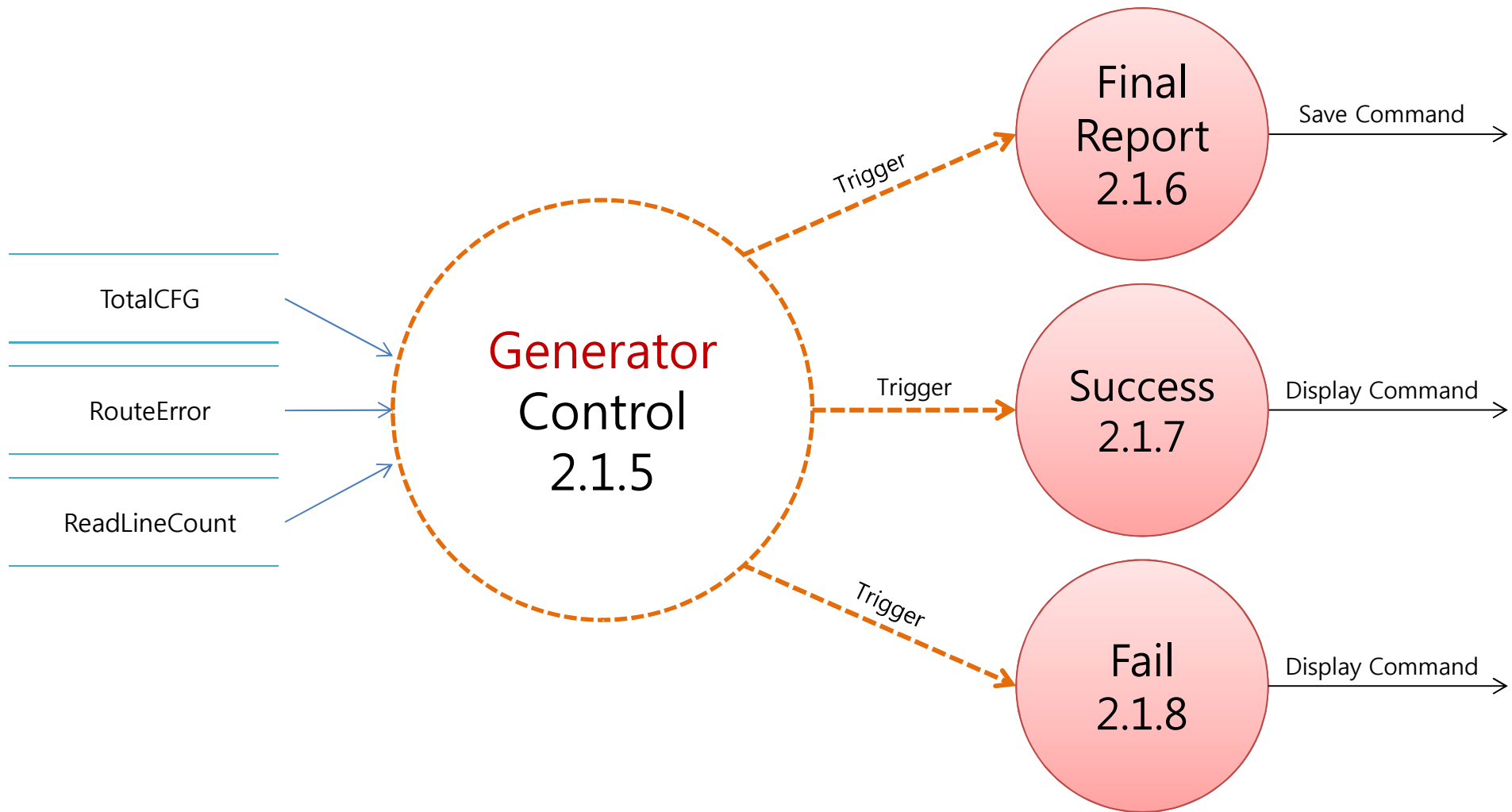
Name		2.1.2.4 For	Name		2.1.2.5 While
Input		Trigger, keyword	Input		Trigger, keyword
Output		ControlState Command	Output		ControlState Command
Process Description		입력 받은 keyword가 "For"라면 그에 해당하는 Control State Command를 내려준다.	Process Description		입력 받은 keyword가 "While"라면 그에 해당하는 Control State Command를 내려준다.

# Data Flow Diagram - Level 4.

## - State Transition Diagram for Controller2.1.2.1



# Data Flow Diagram – Level 5.



# Process Specification

Name	2.1.5 Generator Control
Input	TotalCFG, RouteError, ReadLineCount
Output	Trigger
Process Description	TotalCFG, RouteError, ReadLineCount Data들을 입력 받아 최종적으로 프로그램이 성공되었는지, 실패하였는지를 판단하는 Control Process이다. 성공하였다면 Success와 Final Report를 Trigger하여주고, 실패하였다면 Fail을 Trigger하여 준다.

Name	2.1.6 Final Report
Input	Trigger
Output	Save Command
Process Description	CFG 변환에 성공하였다면 TotalCFG에 최종 Block과 Edge에 관한 Final Report를 추가 작성하여준다.

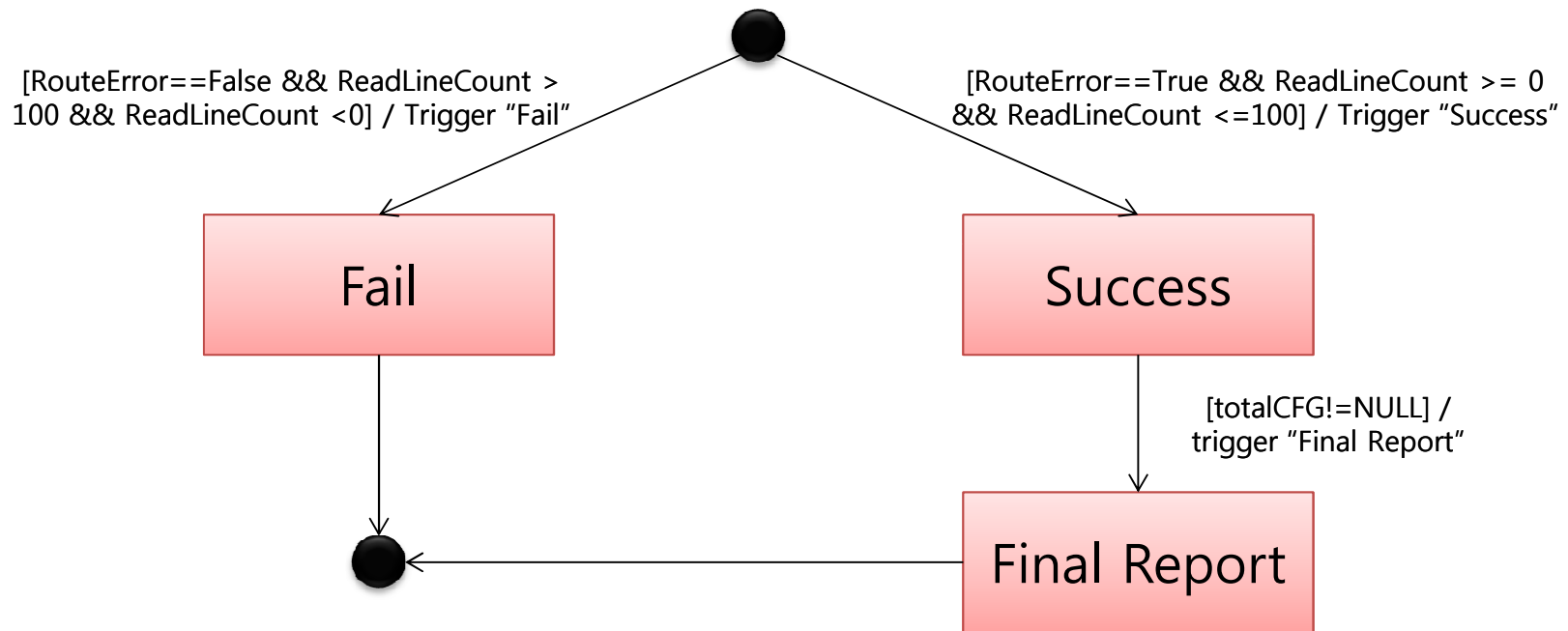
# Process Specification (cont.)

Name	2.1.7 Success
Input	Trigger
Output	Display Command
Process Description	TotalCFG Data의 유효성을 판단하여 Display Command로 출력해준다.

Name	2.1.8 Fail
Input	Trigger
Output	Display Command
Process Description	TotalCFG Data의 부적합성이나, 경로 및 라인초과의 오류를 Display Command로 출력해준다.

# Data Flow Diagram - Level 4.

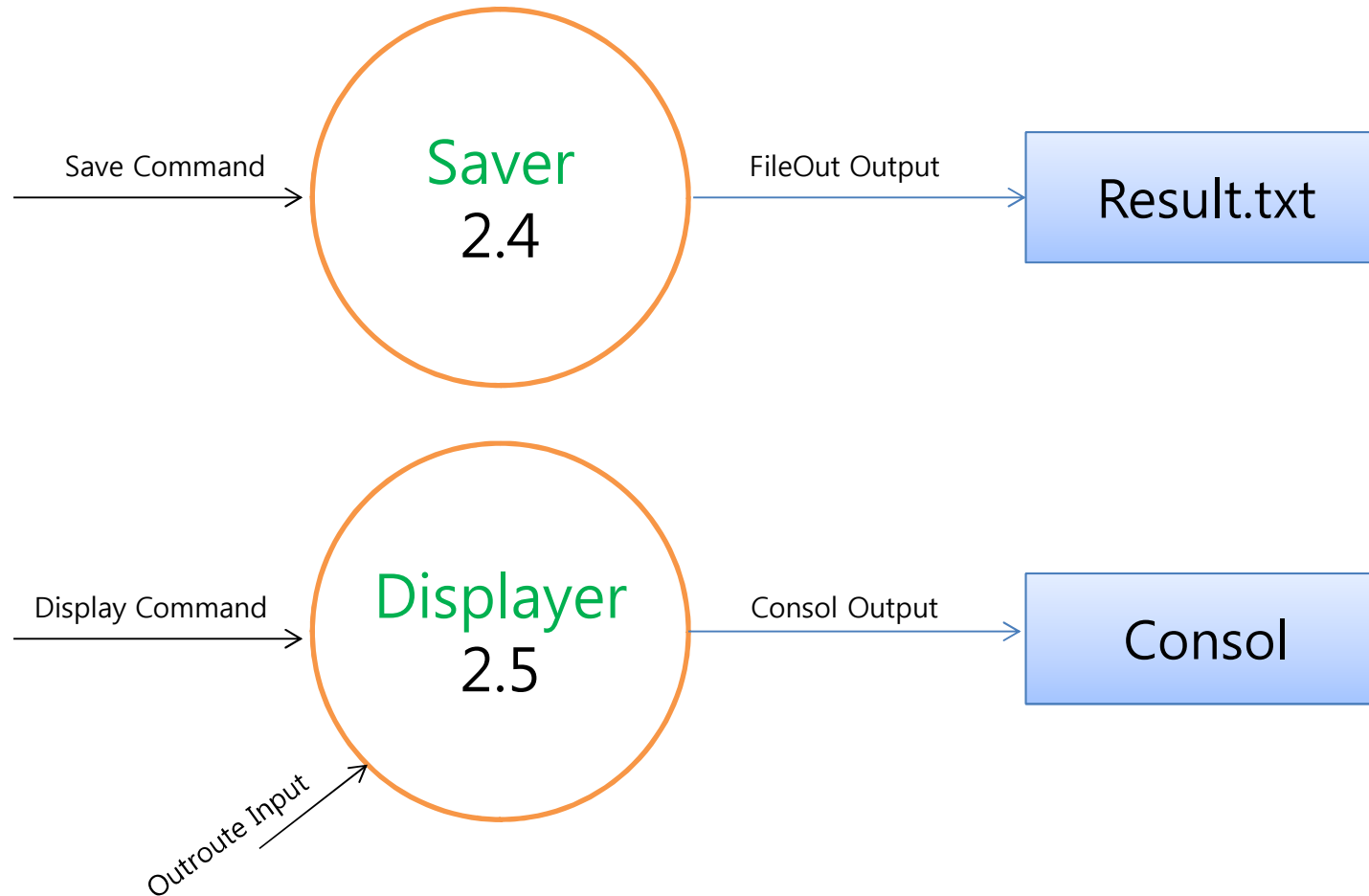
## - State Transition Diagram for Controller2.1.5





# Data Flow Diagram – Level 5.

---

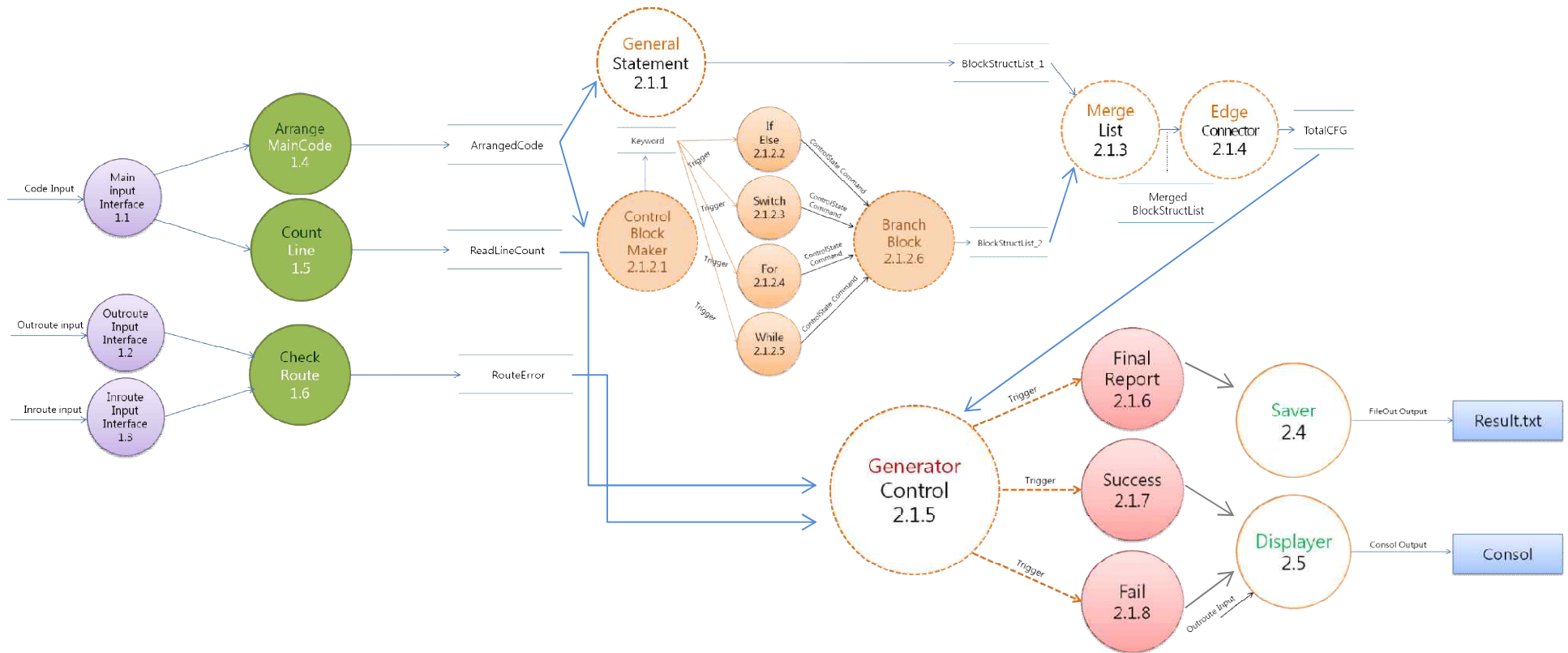


# Process Specification

Name	2.4 Saver
Input	Save Command
Output	FileOut output
Process Description	입력 받은 Save Command를 통해 순차적으로 Result.txt에 TotalCFG Data를 최종적으로 출력하여준다(CUI형식).

Name	2.5 Displayer
Input	Display Command
Output	Consol Output
Process Description	입력 받은 Display Command(성공 혹은 실패)를 통해 Consol 창에 변환 결과를 출력하여 준다. 성공시엔 변환된 Outfile 경로와 성공 메시지를, 실패시엔 실패 메시지와 오류의 원인이 라인수 초과이거나 입력 및 출력 파일 경로에 관한 에러일 경우 실패 원인에 대한 정보도 출력해준다.

# Total DFD



# Data Dictionary

Data 명	설 명	형식
<b>ArrangedCode</b>	Source 코드 내의 main()부분에서 CFG변환 용 부분을 저장한 Data	File
<b>ReadLineCount</b>	Main 코드로부터 읽어 들인 라인 수 Data	Integer
<b>RouteError</b>	File I/O에 필요한 파일의 경로 적합성 판단한 Data	True/False
<b>BlockStructList_1</b>	<p>Block의 정보를 저장하는 구조체로서 Arranged Code 내의 일반적인 Statement 또는 Operation Statement의 Block 정보를 저장한다. 이 구조체는 BlockNumber, OutEdge, BlockContents가 속해있다.</p> <ul style="list-style-type: none"> <li>- BlockNumber : 각 블록들의 번호로서 독립적인 블록의 경우 1, 2, 3 등의 정수형태로, 분기 블록일 경우 1.2, 1.3, 2.1 등의 double 형태로 구성되어 있다.</li> <li>- OutEdge : 각각의 블록이 Out Edge를 가지고 있는지 판단하는 변수로서 있을 경우 Out Edge가 가리키는 BlockNumber를 저장하고, 없을 경우 Null값을 저장한다.</li> <li>- BlockContents : 블록의 내용을 저장한다.</li> </ul>	Struct { int/double, int, string } List[]

# Data Dictionary (cont.)

Data 명	설 명	형식
<b>BlockStructList_2</b>	Block의 정보를 저장하는 구조체로서 Arranged Code 내의 Keyword 형태 (If, Switch, For, While)인 Control Statement의 Block 정보를 저장한다. 이 구조체는 앞의 BlockStructList_1과 같이 BlockNumber, OutEdge, BlockContents가 속해있다.	Struct { int/double, int, String } List[]
<b>TotalCFG</b>	Block과 Edge를 모두 생성하여 전체적인 CFG를 저장하고 있는 Data	Struct Linked List

# The End

---

감사합니다!