

'11. 11. 23, Final SASD

Control Flow Graph Generator

T11

200412351 정백균

200811468 강민석

200913987 이승효

201011340 신승우

Contents.

◆ Structured Analysis

- ✓ Statement of Purpose
- ✓ System Context Diagram
- ✓ Event List
- ✓ DFD(Data Flow Diagram
 - Data Dictionary & Storage
 - Process Specification

◆ Structured Design

- ✓ Structured Chart

Contents.

◆ Structured Analysis

- ✓ Statement of Purpose
- ✓ System Context Diagram
- ✓ Event List
- ✓ DFD(Data Flow Diagram)
 - Data Dictionary & Storage
 - Process Specification

◆ Structured Design

- ✓ Structured Chart

Statement of Purpose

◆ *Software Requirement Specification*(SRS)

- ① 입력은 C-language로 작성한 파일과 사용자가 입력한 명령을 받는다.
- ② 출력은 CFG의 정보를 담는 CUI와 report File이다.
 - Report file에는 입력된 코드로 작성된 CFG의 모든 State를 표시해 준다.

◆ *Environment*

- ③ Cygwin 환경에서 제작되고, 실행은 Command Line 명령어 형태로 한다.
 - Ex) #gcc ./CG Inputcode.c result.txt
 - 잘못된 형태의 명령어를 입력했을 시에는 도움말을 출력하도록 한다.

◆ *Source Code*

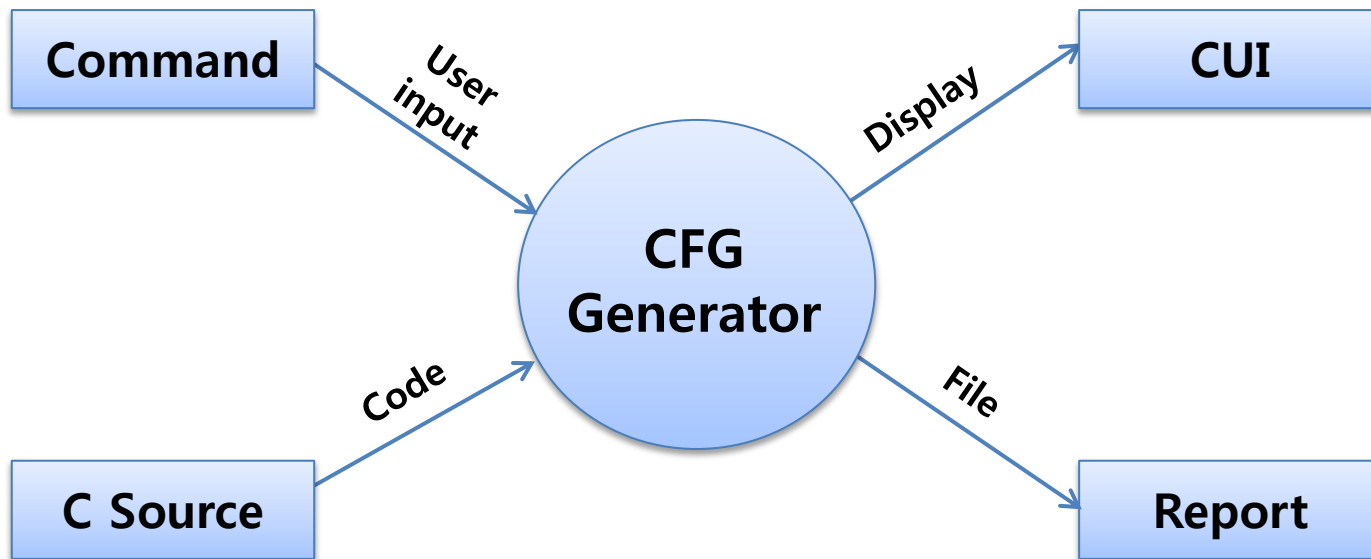
- ④ 코드의 크기는 100~200줄 내외의 프로그램을 대상으로 하고, Main Function을 포함하는 코드여야 한다.
- ⑤ 단일 파일로 되어 있는 코드에 대하여 작동한다. 사용자가 정의한 헤더를 사용한 파일에 대해서는 작동을 보장하지 않는다.
- ⑥ 포인터를 사용하지 않은 코드를 대상으로 한다.
- ⑦ 입력으로 받는 source file의 문법검사는 하지 않는다.

Statement of Purpose

◆ *Our CFG Program*

- ⑦ 입력으로 받은 C Source의 내용을 Block으로 나누고, 이 Block으로 CFG를 생성한다.
- ⑧ 모든 statement를 번호를 이용해 명시화하며 번호순서는 입력으로 받은 파일의 order와 같다.
- ⑨ 함수 호출, for문, while문, switch문에 대한 goto 문으로 처리를 한다.
- ⑩ Block과 Block 사이를 Goto Edge or Down Edge 로 연결한다.
- ⑪ CFG는 변환결과 Report, Block, Edge, Error Message로 구성된다.
- ⑫ CFG에 input으로 들어가는 C Source 파일은 한 라인당 한 instruction만 있어야 한다.

System Context Diagram



Event List

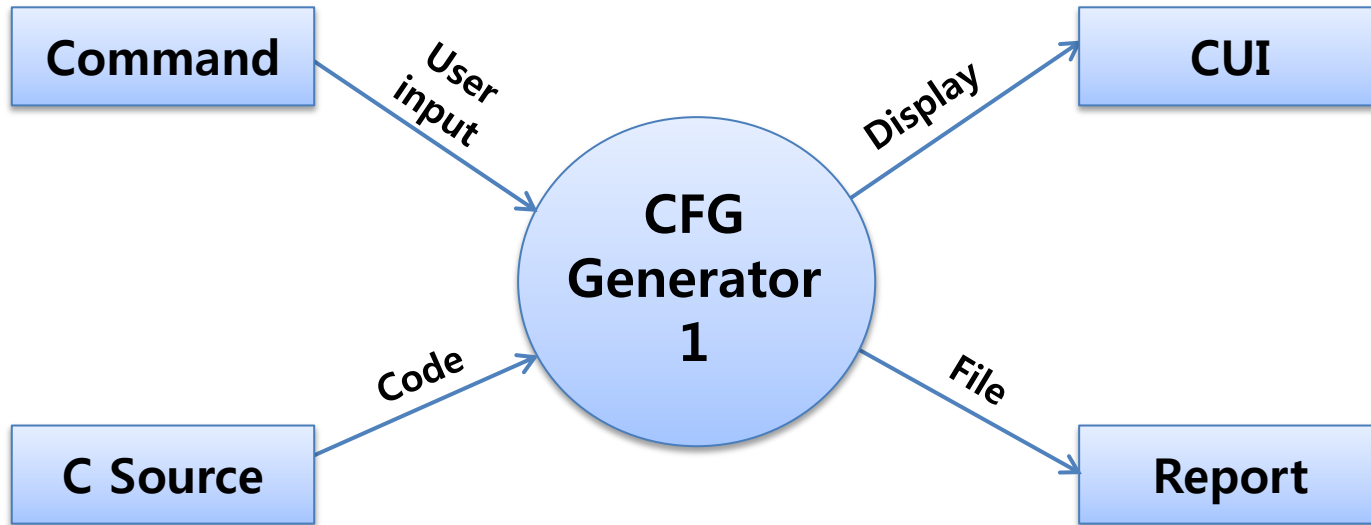
◆ Input Event

Input Event	Description	Format / Type
C Source File	100-200줄 내외의 정상적으로 작동하는 C Source	*.C file
Command	입력 커맨드 ex) #gcc ./CG Inputcode.c result.txt	gcc 명령어

◆ Output Event

Output Event	Description	Format / Type
CUI	결과로 나온 CFG를 CUI로 출력	Display
Report	결과로 나온 CFG를 파일에 출력	Text file

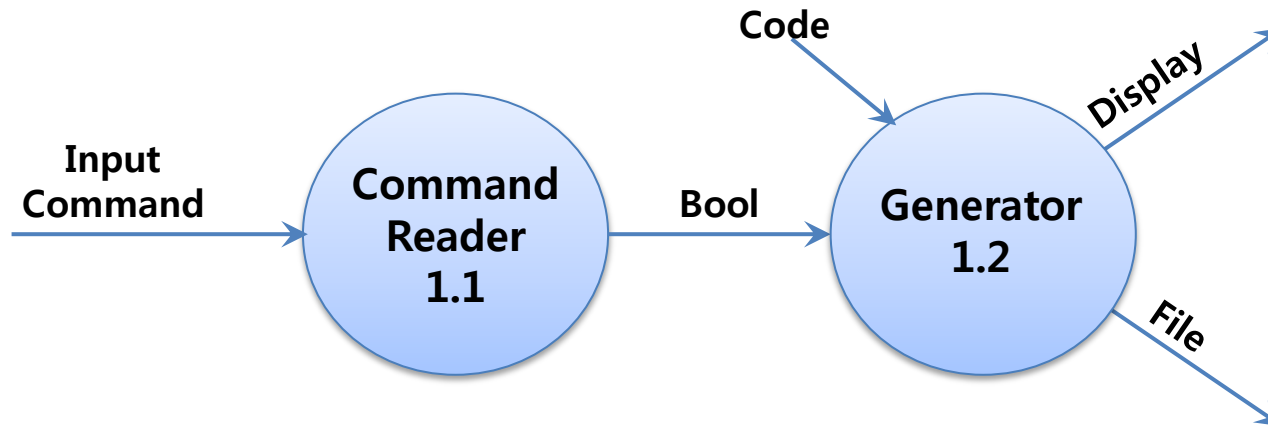
DFD Level 0



◆ Data Dictionary

Input Event	Description	Format / Type
Input Command	사용자가 입력하는 명령어로, 입력 받을 C Source File의 경로와 CFG를 출력할 File 이름을 지정	String
Code	정상적으로 작동하는 C 소스 파일	*.c
Display	완성된 CFG를 콘솔 화면에 출력	Console Display
File	완성된 CFG를 파일로 생성	*.txt

DFD Level 1



◆ Data Dictionary

Input Event	Description	Format / Type
Bool	Integer형 옵션값. FALSE : 0, TRUE : 0이 아닌 모든 수	Integer

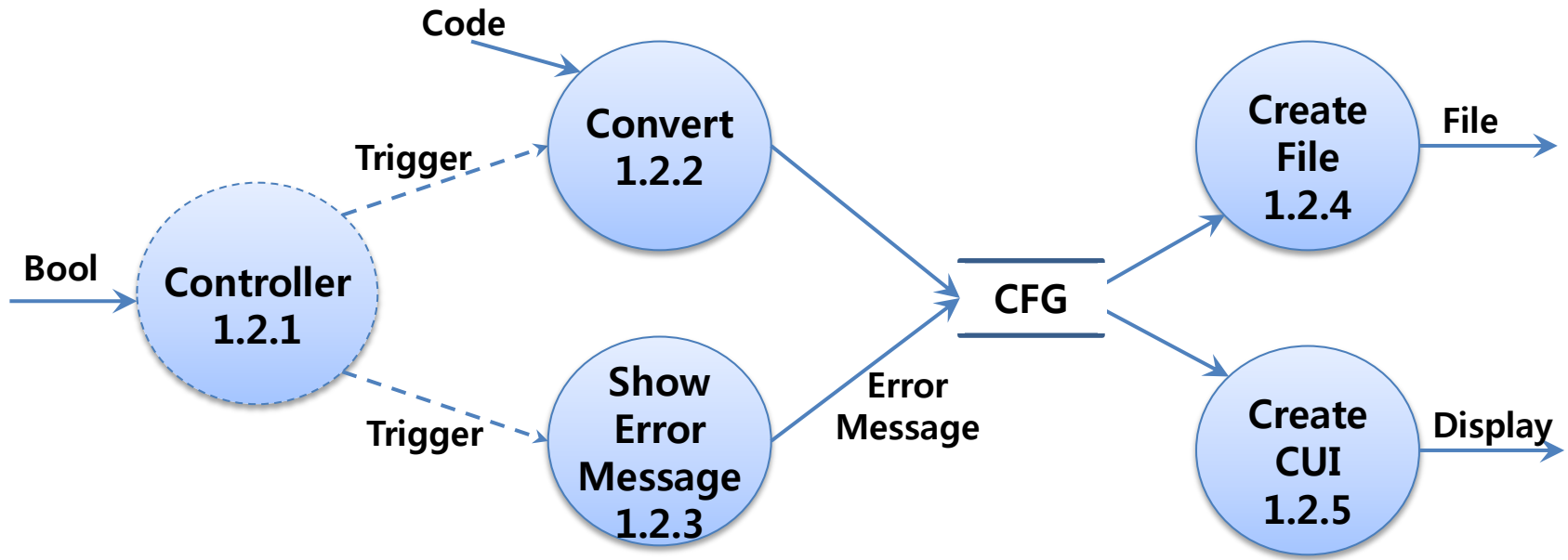
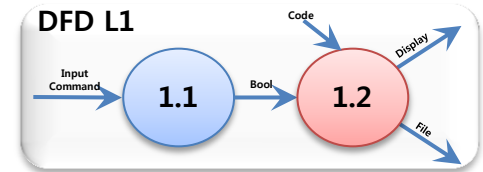
Process Specification

- DFD Level 1

Reference No.	1.1
Name	Command Reader
Input	Command
Output	bool
Process Description	프로그램 실행 시 커맨드를 받은 뒤, 이 커맨드가 적합한지 검사하고, 적합여부를 bool로 출력한다. 입력 받는 커맨드에 들어가는 항목은 입력 c파일의 경로, 출력 report파일 이름이다.

Reference No.	1.2
Name	Generator
Input	bool, C Source
Output	CUI, File
Process Description	bool결과에 따라 CSource를 읽어 오고, 이 소스를 CFG로 변환한 뒤, CUI와 File형태로 출력하다.

DFD Level 2



◆ Data Dictionary

Input Event	Description	Format / Type
Error Message	Controller의 입력인 Bool이 FALSE일 때 호출하는 Show Error Message 프로세스가 만들어내는 String. 어느 부분이 잘못되었는지에 대한 정보를 담는다.	String
CFG	Control Flow Graph의 약자. 하나의 구조체로, 블록들의 리스트, 블록을 연결하는 엣지들의 리스트, 에러 발생 유무 등의 정보를 담는다.	Struct

Process Specification

- DFD Level 2

Reference No.	1.2.1
Name	Controller
Input	bool
Output	Trigger
Process Description	Bool결과에 따라 프로세스 Convert 혹은 Show Error Message에 Trigger를 보낸다.

Reference No.	1.2.2
Name	Convert
Input	Trigger, Code
Output	CFG
Process Description	트리거를 받으면 C Source를 CFG로 변환 한 뒤, CFG 데이터 스토어에 저장한다.

Process Specification

- DFD Level 2

Reference No.	1.2.3
Name	Show Error Message
Input	Trigger
Output	CFG
Process Description	Trigger를 받으면 CFG 데이터 스토어에 에러메시지를 기록한다.

Reference No.	1.2.4
Name	Create File
Input	CFG
Output	File
Process Description	완성된 CFG를 File 형태로 만들어 출력한다.

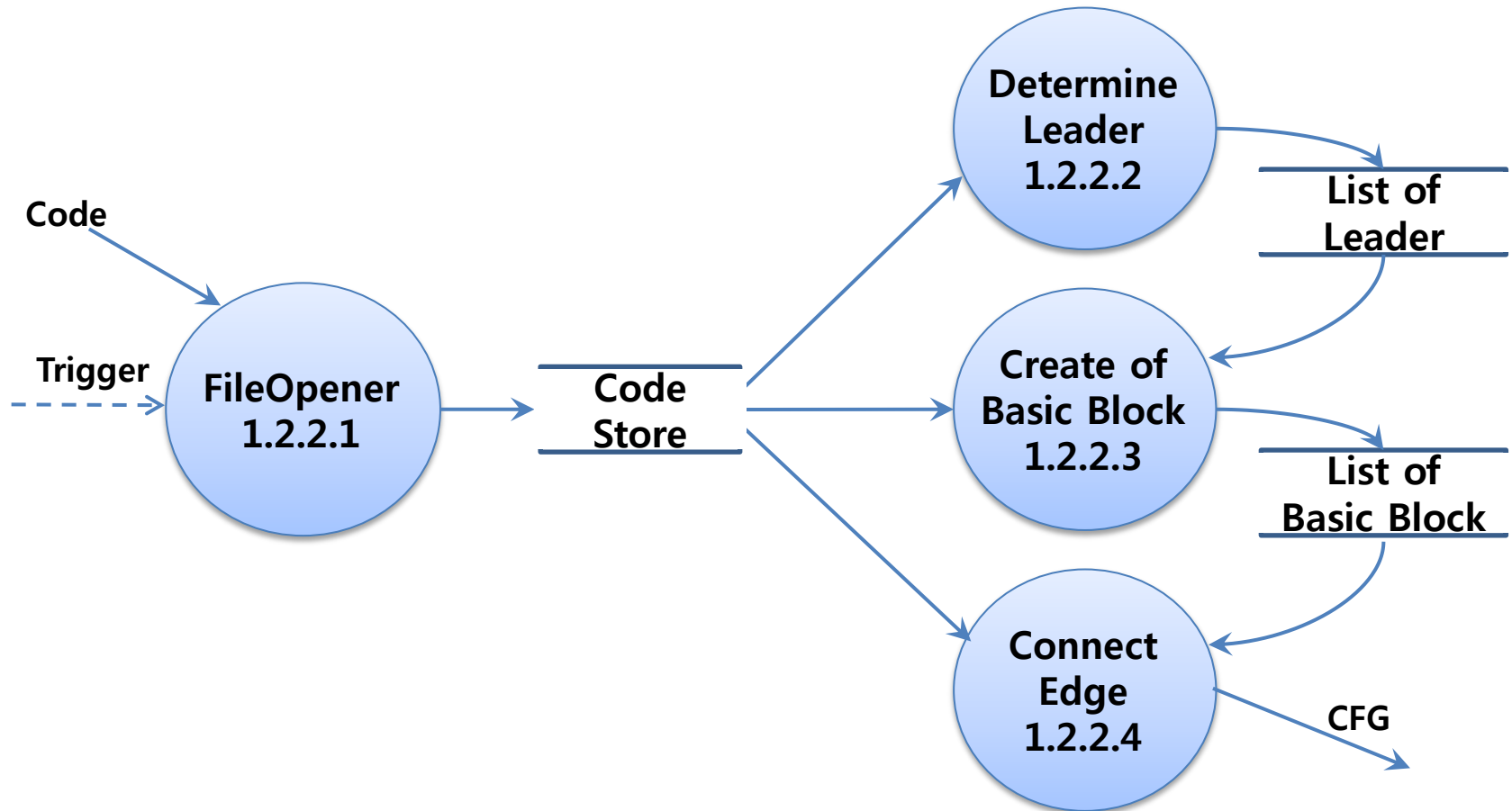
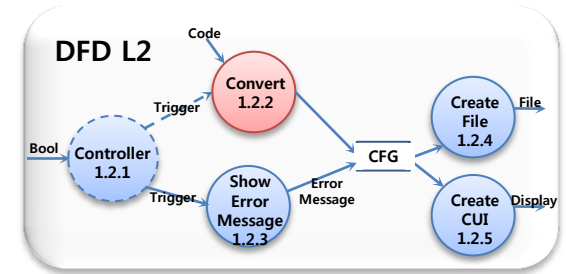
Process Specification

- DFD Level 2

Reference No.	1.2.5
Name	Create CUI
Input	CFG
Output	Display
Process Description	완성된 CFG를 Display 형태로 만들어 출력한다.

DFD Level 3

- Convert



DFD Level 3

- Convert

◆ Data Dictionary

Input Event	Description	Format / Type
Code Store	File Opener로부터 열린 FILE을 다른 프로세스가 접근할 수 있도록 저장한다.	FILE *
List of Leader	Leader Node를 리스트 형태로 묶은 자료구조이다.	List
Node of Leader	블록의 시작이 되는 statement - Leader가 되는 행의 행 번호를 int를 사용해 나타낸다. - Leader가 되는 조건은 다음과 같다. ① 프로시저의 첫 번째 행 ② Goto문의 타겟이 되는 행 ③ Goto문의 바로 다음 행 ④ 분기문 행	Integer
List of Basic Block	Basic Block의 리스트	List
Node of Basic Block	c소스 중 동작을 할 때 함께 동작하여 하나로 구분될 수 있는 부분을 블록화 한 구조체이다. 이것은 Leader에서 다음 Leader 전까지의 문장의 집합이다.	Struct

Process Specification

- DFD Level 3

Reference No.	1.2.2.1
Name	File Opener
Input	Trigger, Code
Output	Code Store
Process Description	Trigger를 받으면, fopen으로 Code를 열어 Code Store에 저장한다.

Reference No.	1.2.2.2
Name	Determine Leader
Input	Code Store
Output	List of Leader
Process Description	Code Store의 각 statement를 읽어 Leader인지 검사하여 Leader의 리스트를 뽑아낸다.

Process Specification

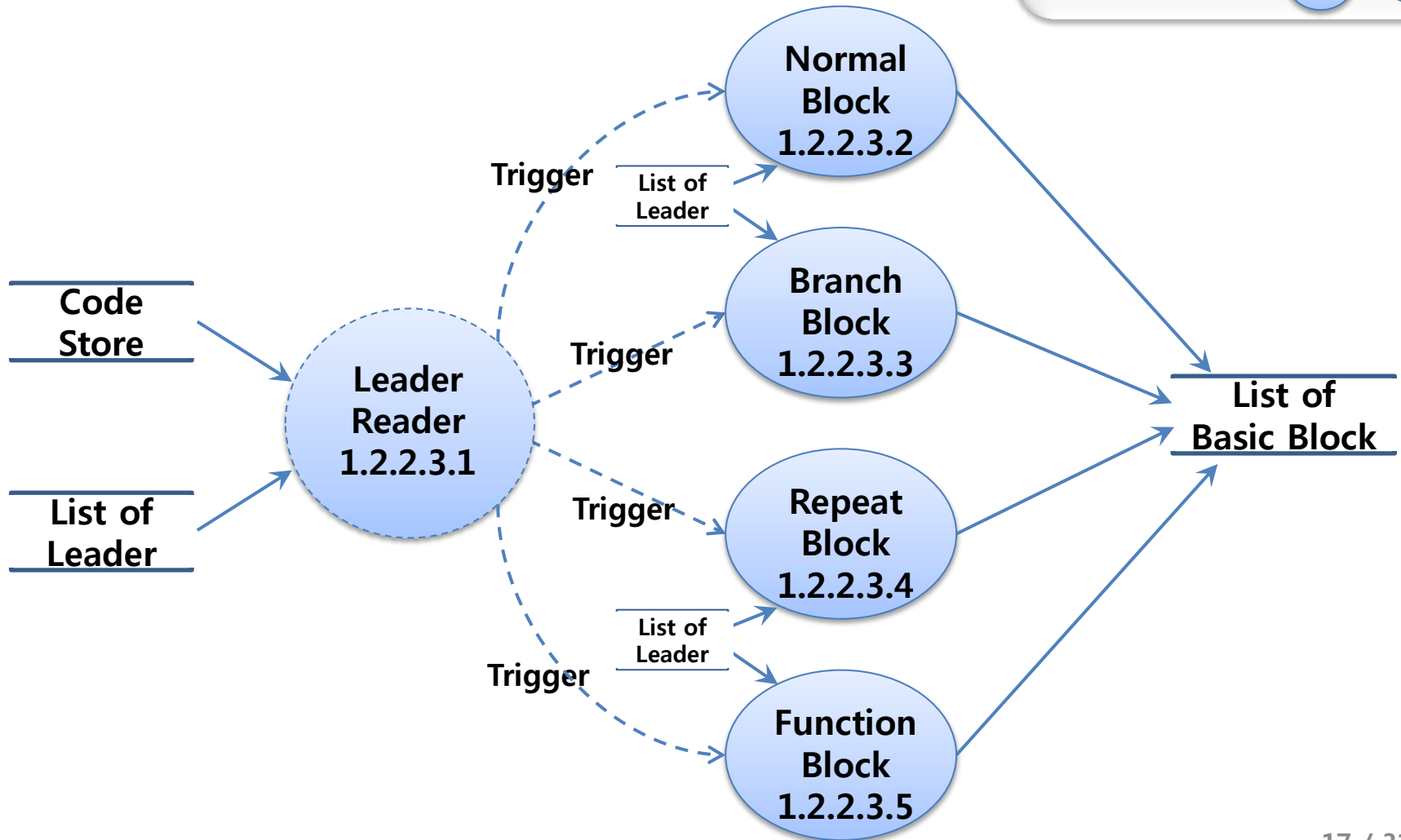
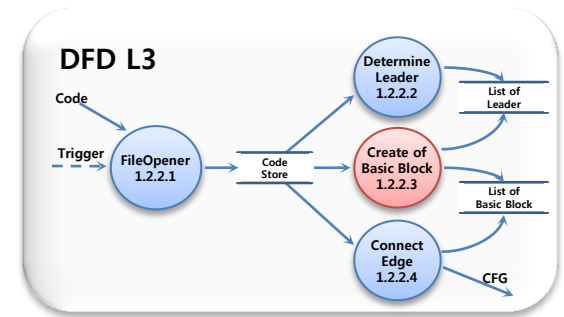
- DFD Level 3

Reference No.	1.2.2.3
Name	Create Basic Block
Input	List of Leader, Code Store
Output	List of Basic Block
Process Description	Leader의 리스트를 받아 Basic Block을 구성 한 뒤, 리스트 화 하여 데이터 스토어에 저장한다.

Reference No.	1.2.2.4
Name	Connect Edge, Code Store
Input	List of Basic Block
Output	CFG
Process Description	Basic Block의 리스트를 받아 Edge를 생성하고 완성된 CFG를 데이터 스토어에 기록한다.

DFD Level 4.1

- Create of Basic Block



Process Specification

- DFD Level 4.1

Reference No.	1.2.2.3.1
Name	Leader Reader
Input	List of Leader, Code Store
Output	Trigger
Process Description	Leader List와, Code Store를 받아, 각 Leader의 문장에 따라 Block의 형식을 분류해 각 Block Transfer로 트리거를 보낸다.

Reference No.	1.2.2.3.2
Name	Normal Block
Input	List of Leader, Trigger
Output	Basic Block Node
Process Description	트리거가 들어오면 일반 Block을 생성하여 Basic Block List에 Node로써 추가한다.

Process Specification

- DFD Level 4.1

Reference No.	1.2.2.3.3
Name	Branch Block
Input	List of Leader, Trigger
Output	Basic Block Node
Process Description	트리거가 들어오면 분기Block을 생성하여 Basic Block List에 Node로써 추가한다.

Reference No.	1.2.2.3.4
Name	Repeat Block
Input	List of Leader, Trigger
Output	Basic Block Node
Process Description	트리거가 들어오면 Block을 반복Block을 생성하여 Basic Block List에 Node로써 추가한다.

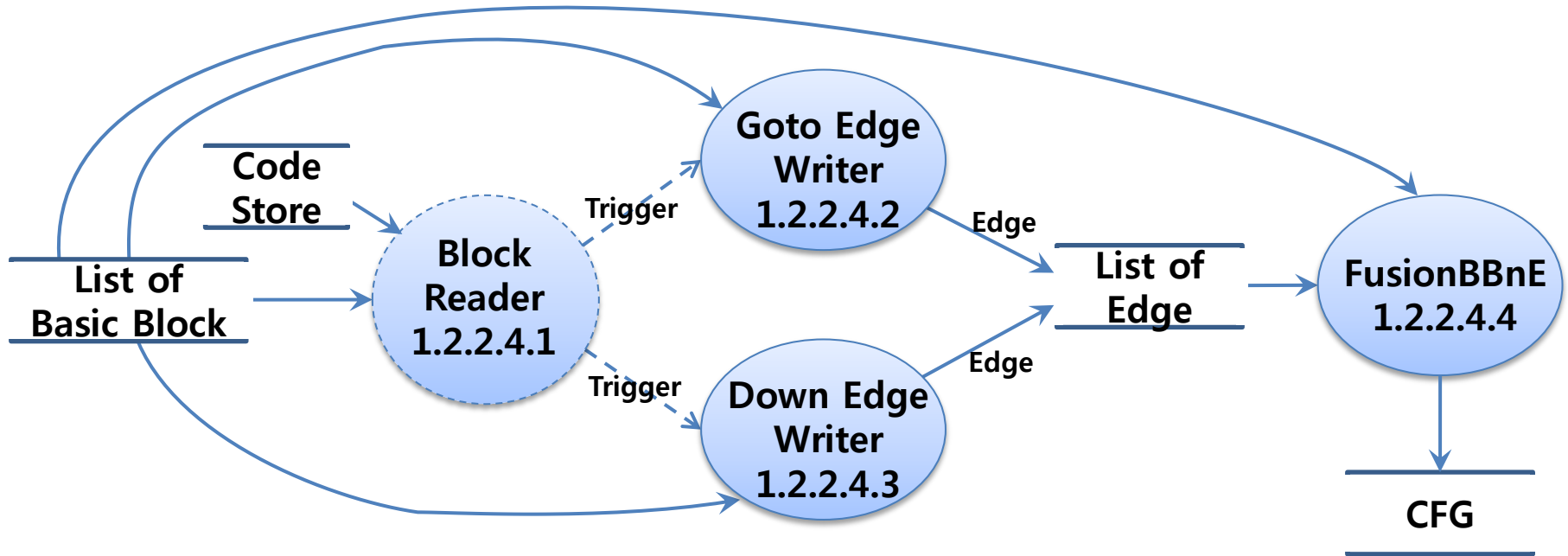
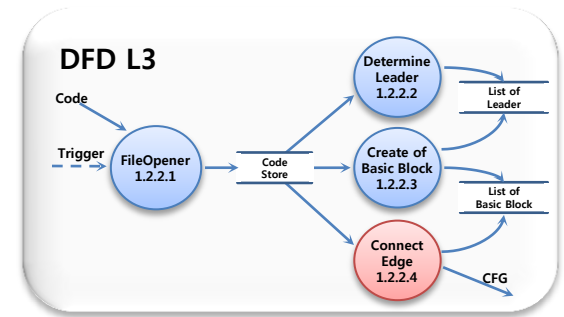
Process Specification

- DFD Level 4.1

Reference No.	1.2.2.3.5
Name	Function Block
Input	List of Leader, Trigger
Output	Basic Block Node
Process Description	트리거가 들어오면 Block을 함수Block을 생성하여 Basic Block List에 Node로써 추가한다.

DFD Level 4.2

- Connect Edge



◆ Data Dictionary

Input Event	Description	Format / Type
Edge	Basic Block을 연결한다. 시작 Basic Block의 번호와 이동하는 Basic Block의 번호를 저장한다.	Struct

Process Specification

- DFD Level 4.2

Reference No.	1.2.2.4.1
Name	Block Reader
Input	List of Basic Block, Code Store
Output	Trigger
Process Description	Basic Block의 리스트와 Code Store를 Block의 끝부분을 조사 한 뒤, Goto Edge Writer 혹은 Down Edge Writer에 Trigger를 보낸다.

Reference No.	1.2.2.4.2
Name	Goto Edge Writer
Input	List of Basic Block, Trigger
Output	CFG
Process Description	Trigger가 들어오면 Block에 해당하는 Goto edge를 만든다.

Process Specification

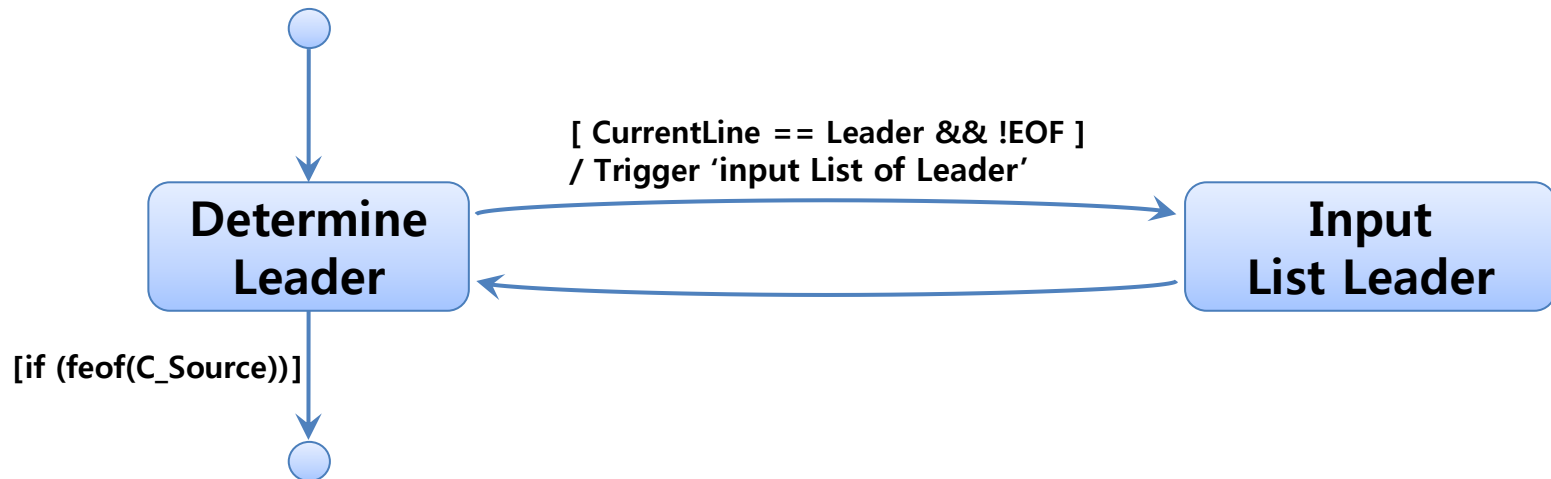
- DFD Level 4.2

Reference No.	1.2.2.4.3
Name	Down Edge Writer
Input	List of Basic Block, Trigger
Output	CFG
Process Description	Trigger가 들어오면 Block에 해당하는 Down edge를 만든다.

Reference No.	1.2.2.4.4
Name	FusionBBnE
Input	List of Edge, List of Basic Block
Output	CFG
Process Description	완성된 베이직 블록과 Edge의 리스트를 연결 순서에 맞게 CFG구조체에 올린다.

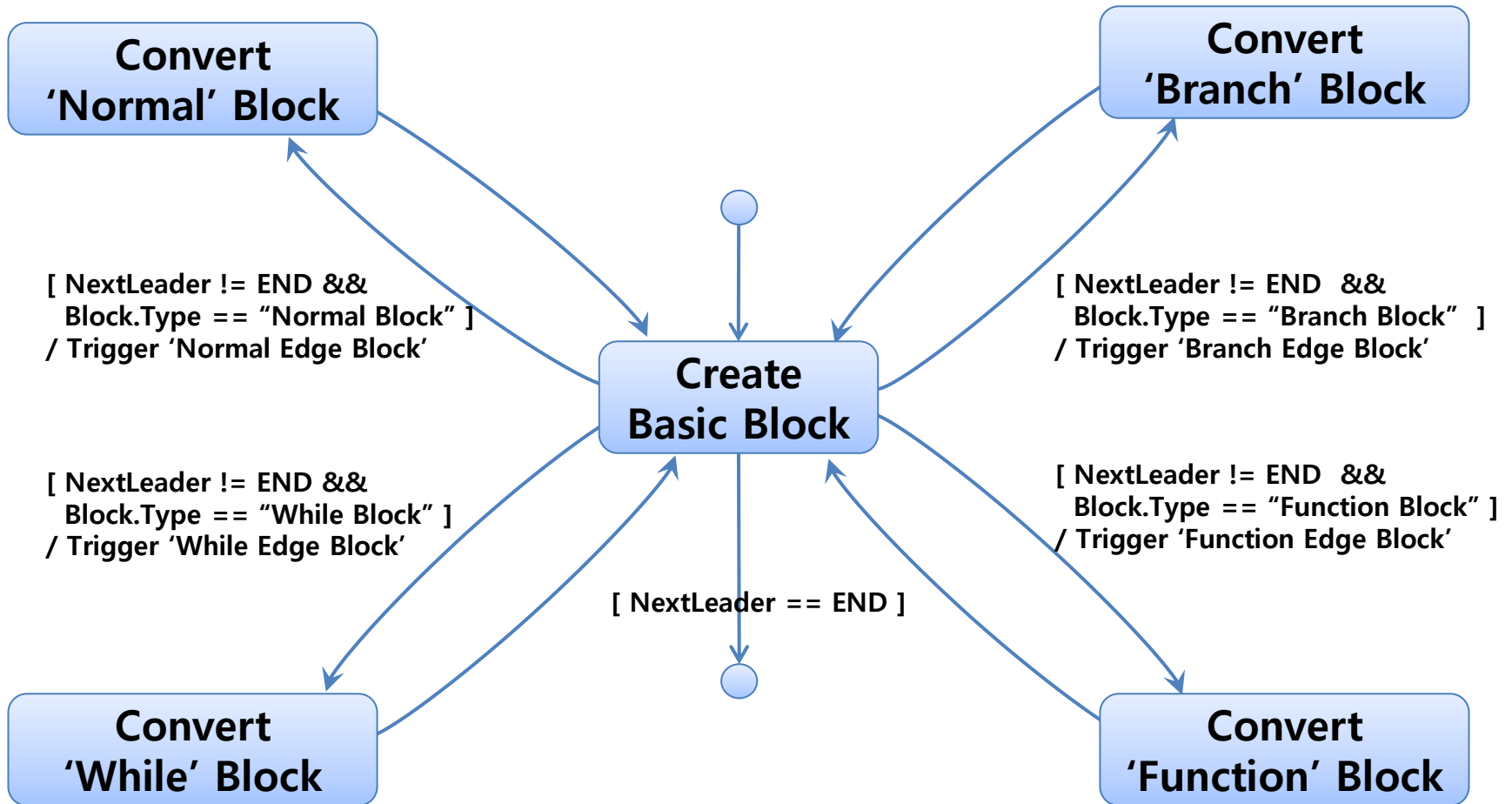
DFD Level 5

- Finite State Machine 1



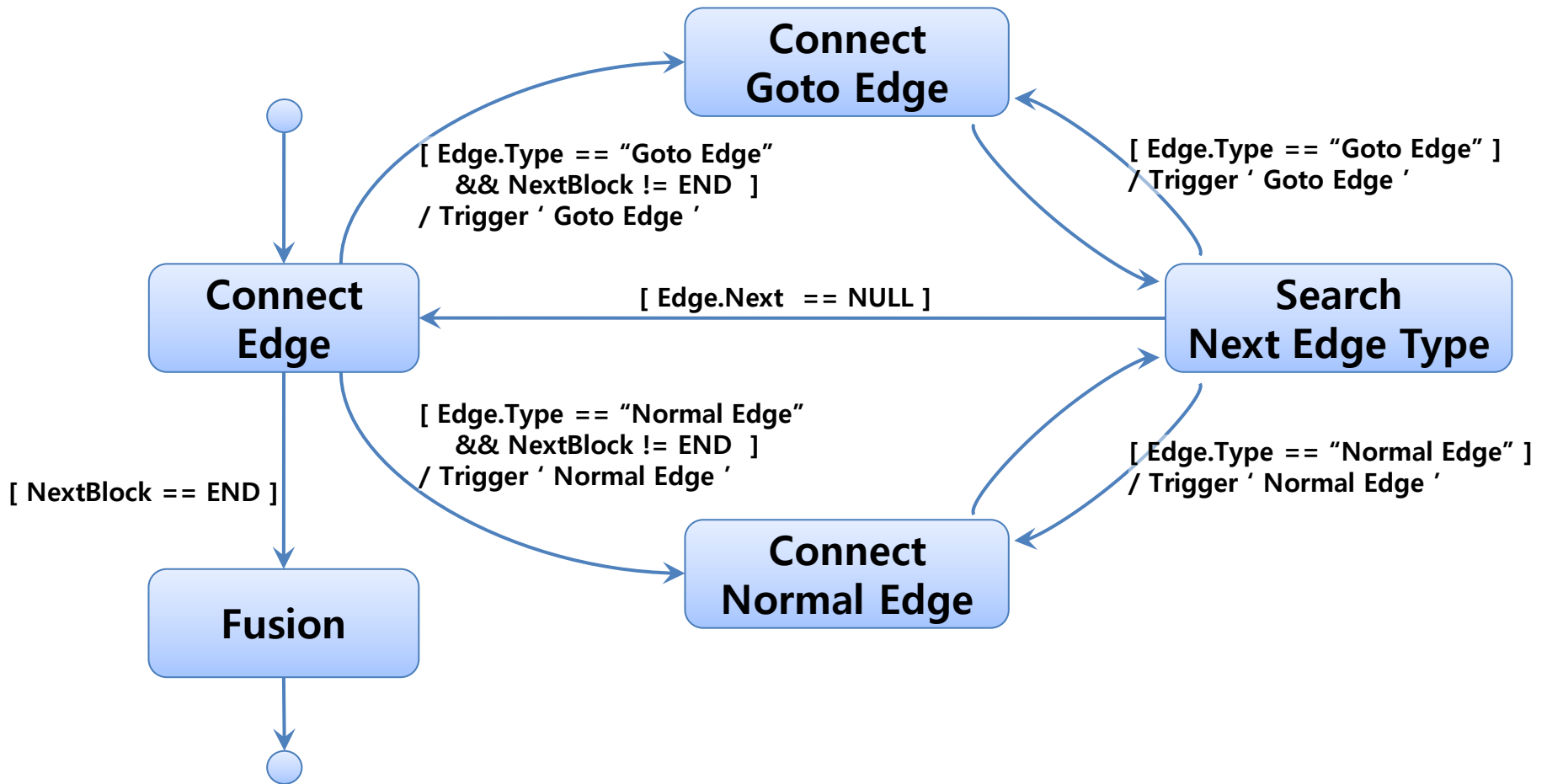
DFD Level 5

- Finite State Machine 2

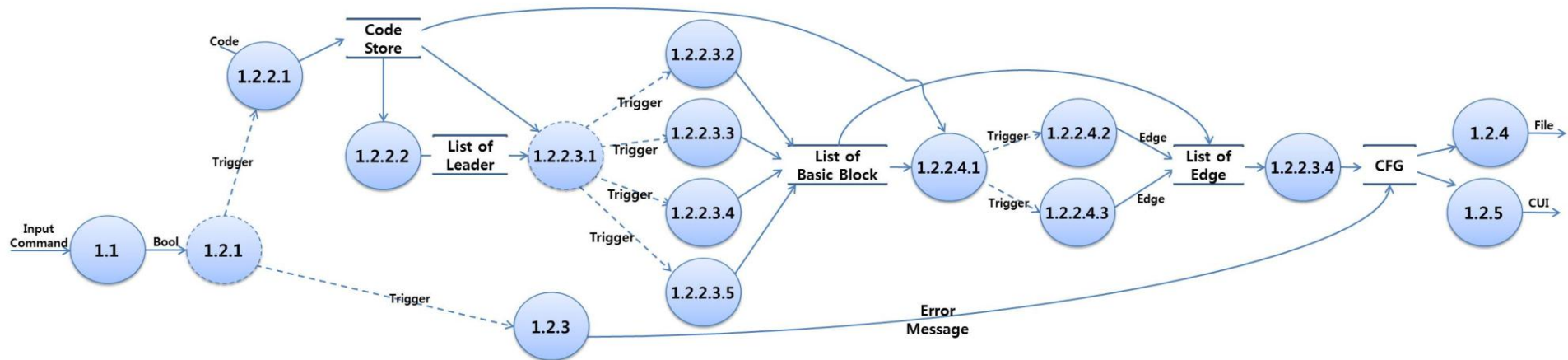


DFD Level 5

- Finite State Machine 3



DFD - Overall



Contents.

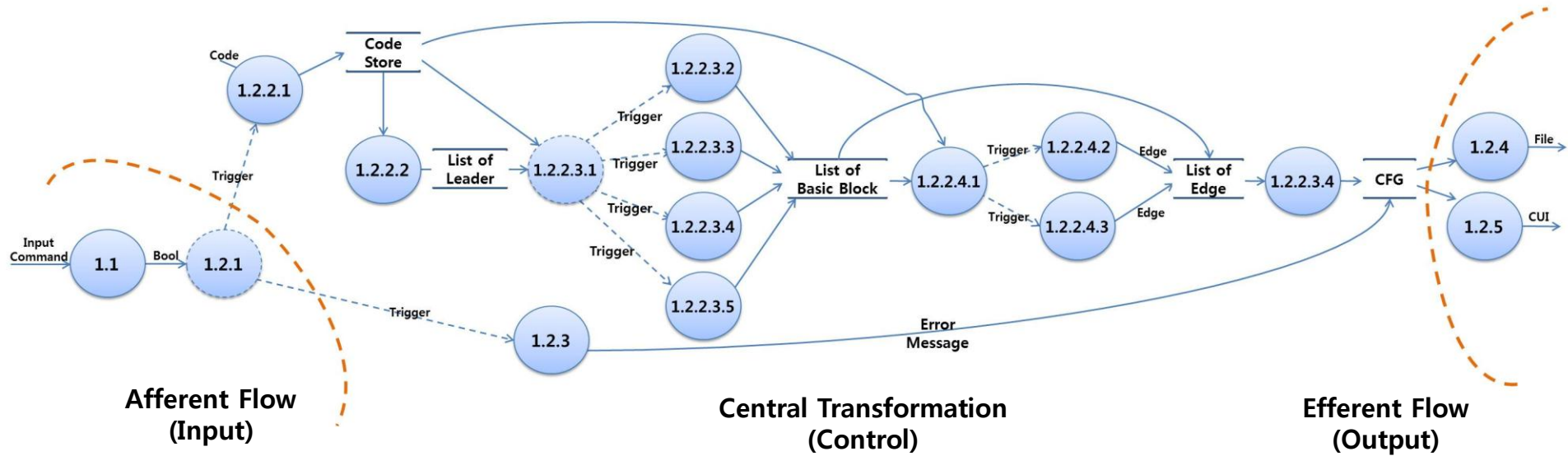
◆ Structured Analysis

- ✓ Statement of Purpose
- ✓ System Context Diagram
- ✓ Event List
- ✓ DFD(Data Flow Diagram
 - Data Dictionary & Storage
 - Process Specification

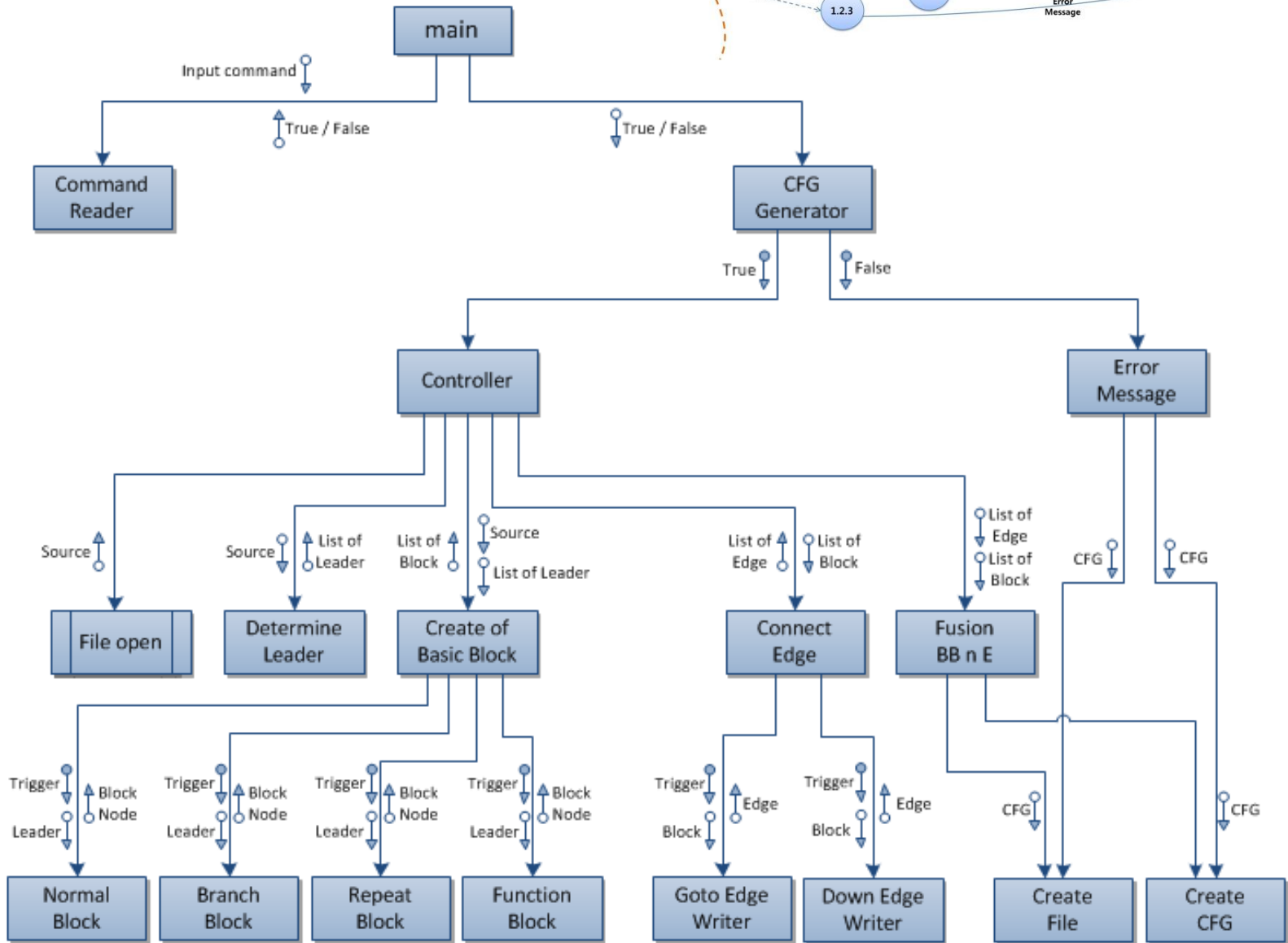
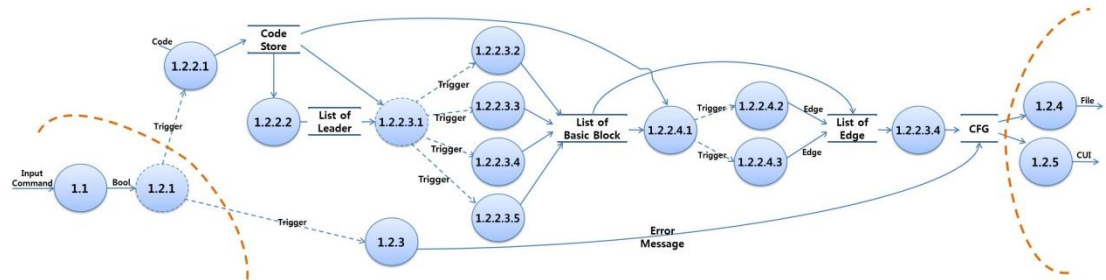
◆ Structured Design

- ✓ Structured Chart

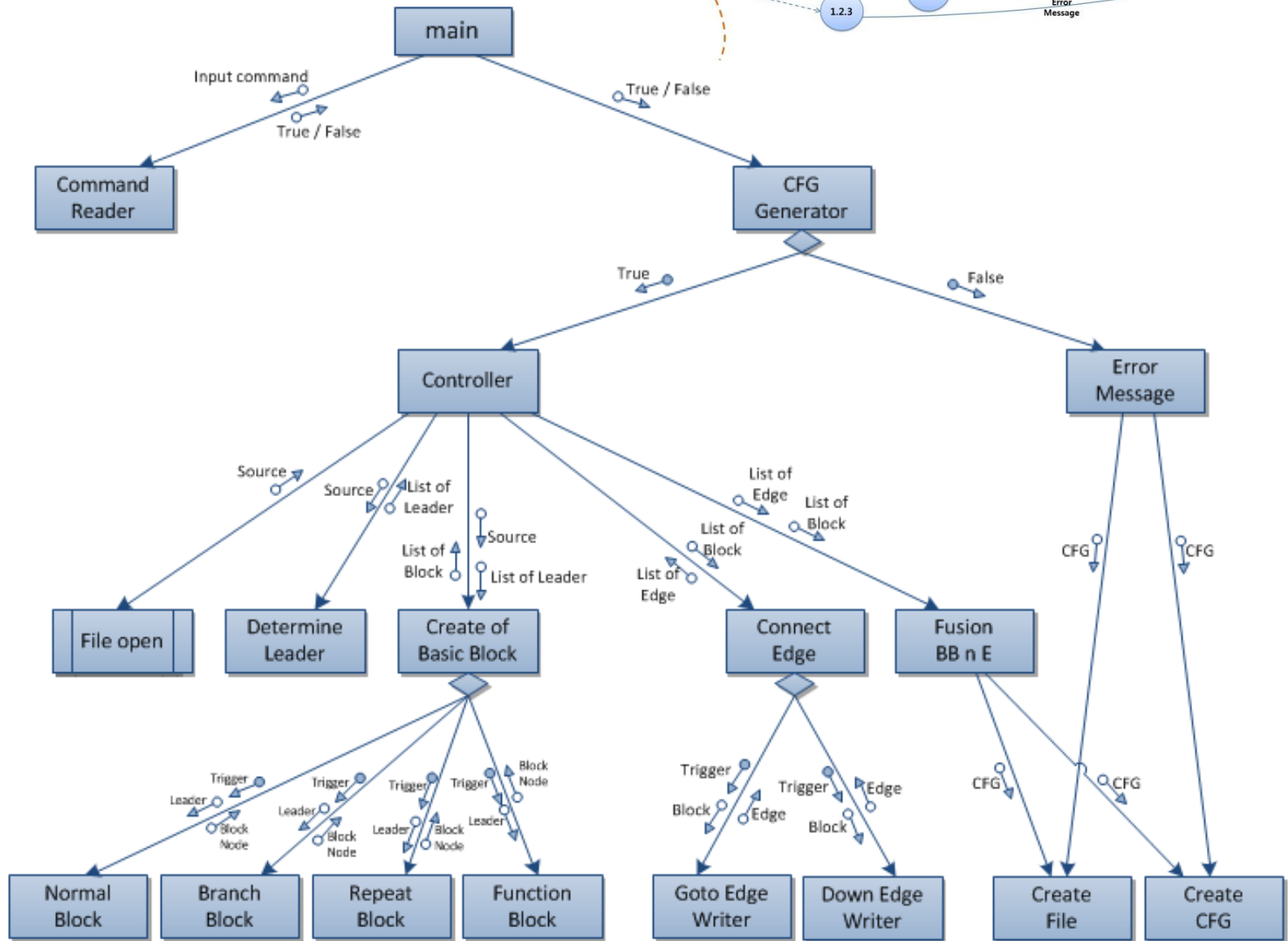
In DFD Overall



Structured Chart(Basic)



Structured Chart(Advanced)



References.

◆ Definition of DFD

- 소프트웨어 공학 : 구조적 방법론 중심, 이성구, 2010

◆ Definition of CFG

- **Wikidepia keyword : “Control Flow Graph”**
 - http://en.wikipedia.org/wiki/control_flow_graph
- **Software Visualization** - [Stephan Diehl](#), 2007
 - pp.40 : 3.2.2 Control-Flow graph

◆ Algorithm : Create of CFG

- **Representation and Analysis of Software**
 - Gregg Rothermel, University of Nebraska – Lincoln, January 10, 2005



Thank You!!