

# CFG Generator - SASD

이소연 201011351

하서희 201011374

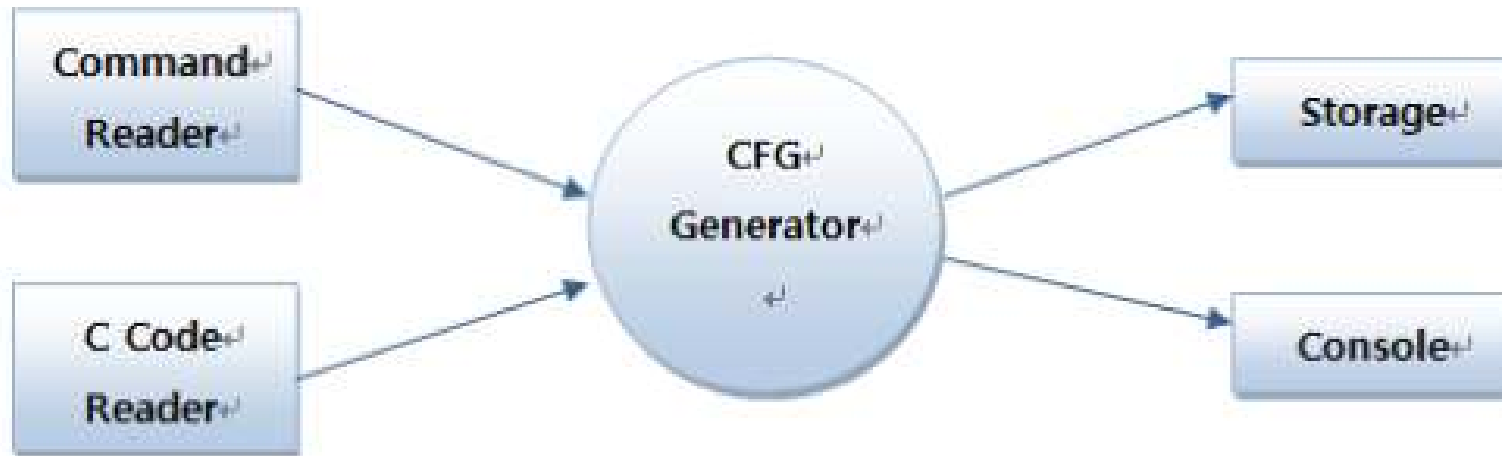
# contents

- 1.statement of purpose
- 2.system context diagram
- 3.event list
- 4.
  - DFD level 0,1,2,3,4
  - FSM level
- 5.full DFD
- 6.Data dictionary
- 7.process specification
- 8.structured chart (basic)
- 9.structured chart (advanced)

# Statement Of Purpose

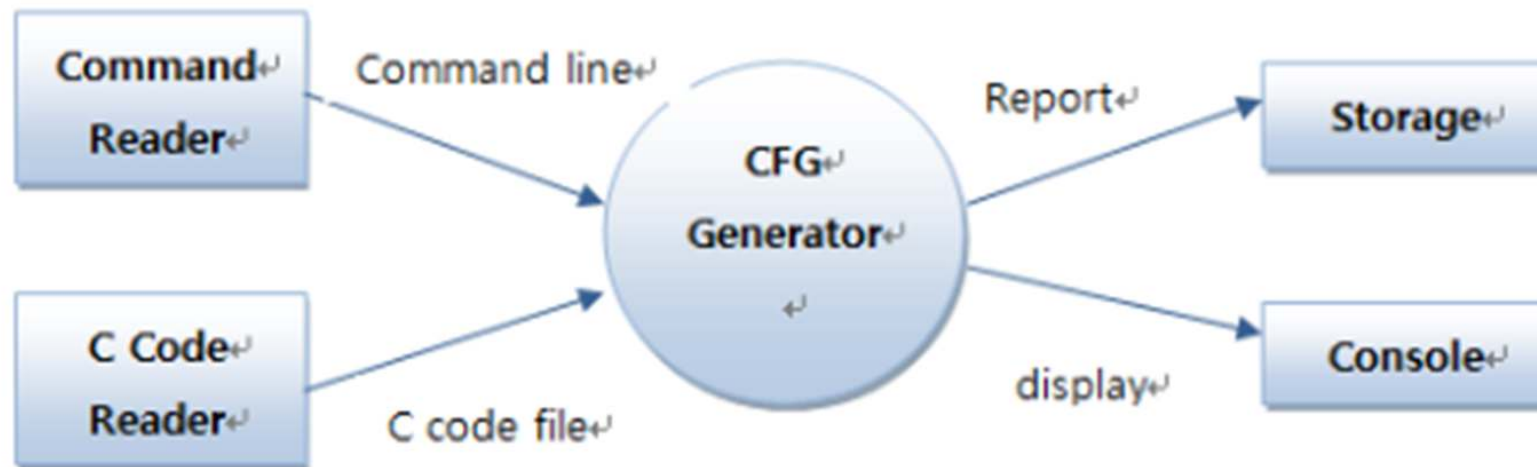
- C코드를 CFG로 생성(텍스트 형식)
- Command line형태로 명령어를 제공받음  
./CG (변환하기 위해 읽어들이 C파일).c (report를 텍스트 형식으로 출력할 파일).txt
- State목록과 Edge목록을 list로 report 제공
- 잘못된 command line입력시 도움말 및 에러 출력후 system out
- 프로그램의 수행과정을 텍스트 형식으로 나타냄
- 변환할 코드 파일에 대한 입력의 성공/실패 여부를 알려줌
- 변환 시작 여부를 알려줌
- Report 생성 후 report파일명 제공
- 메인 함수를 포함하는 코드를 변환
- 사용자 정의 헤더는 대상에서 제외하고 포인터가 제외된 단일 파일을 대상으로 한다

# System Context Diagram

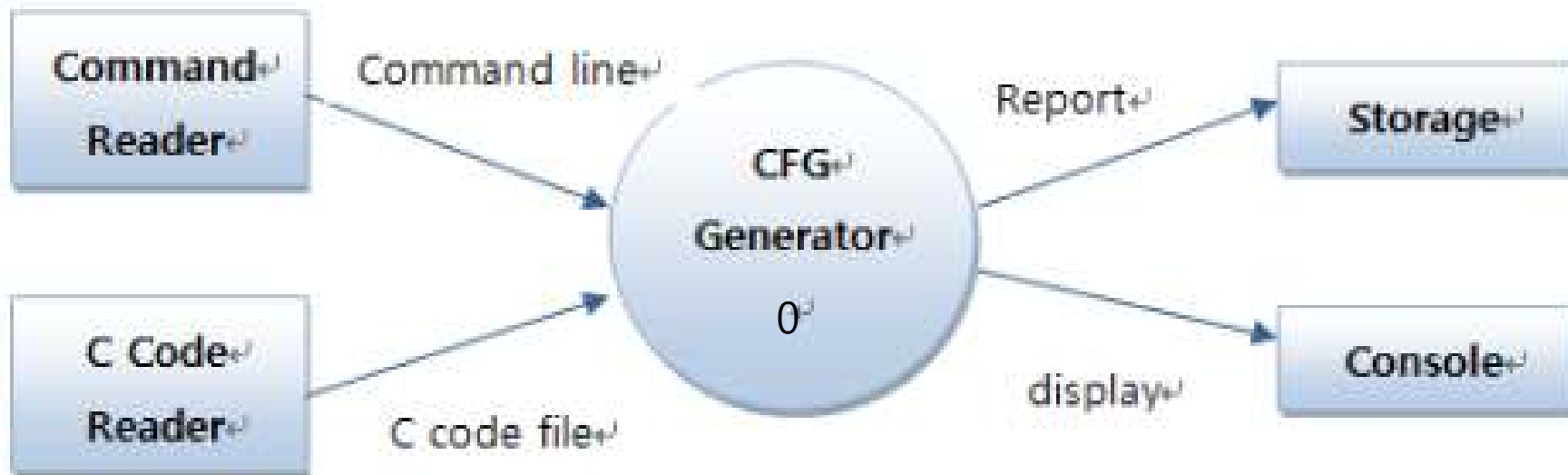


# Event list

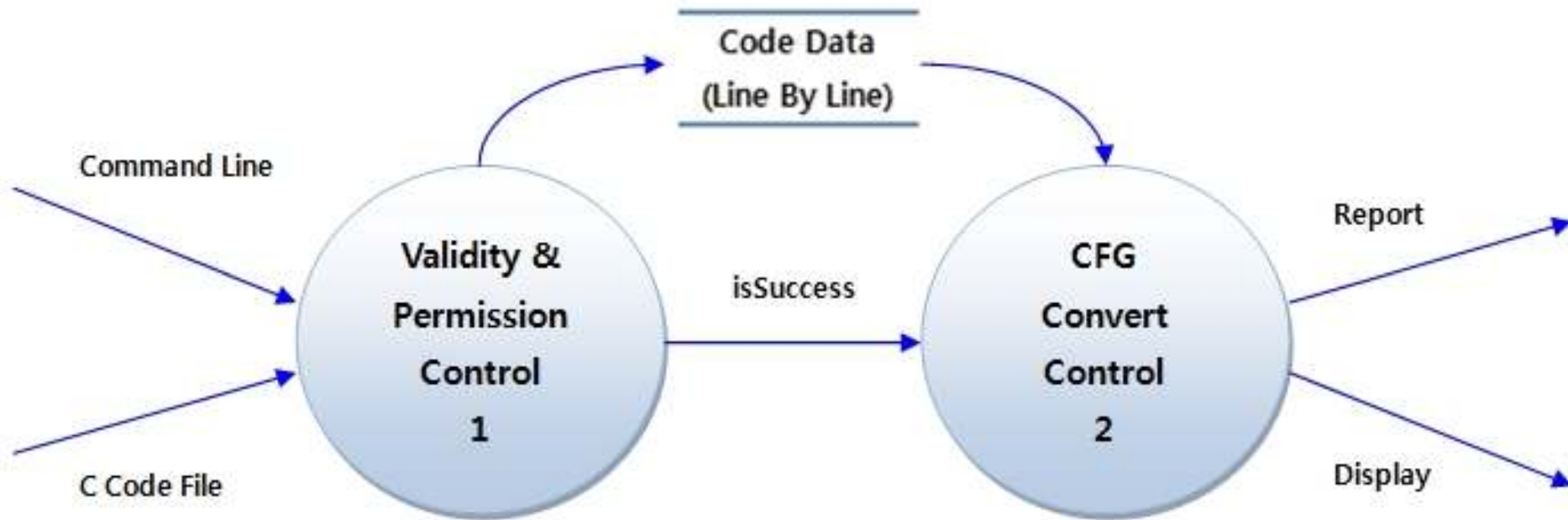
| Event(input/output) | Description                                   |
|---------------------|---|
| Command Line        | 사용자로부터 Command Reader를 통해 Command Line을 읽어 들임 |
| C Code File         | C Code Reader로부터 파일 데이터를 읽어 들임                |
| Report              | State list, Edge list에 저장된 정보들로 부터 전체 목록을 출력  |
| Display             | 텍스트형식으로 생성시킨 CFG 들을 실제로 출력하여 나타냄              |



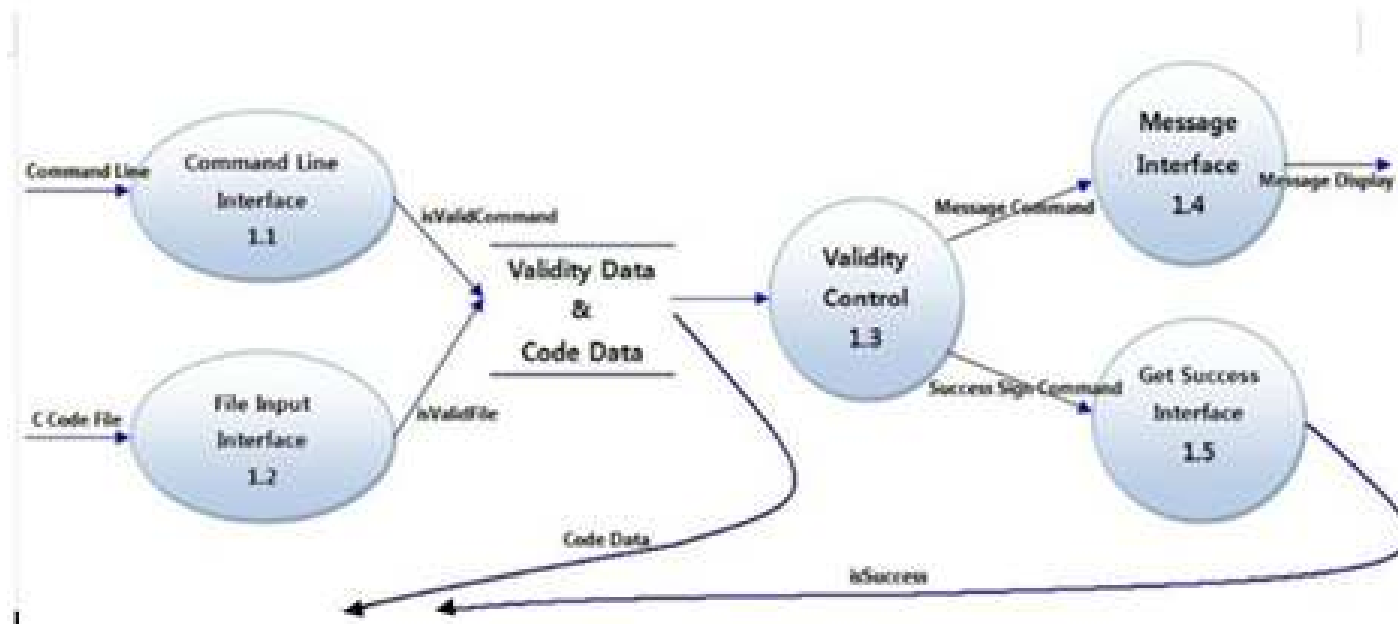
# DFD Level 0



# DFD Level 1

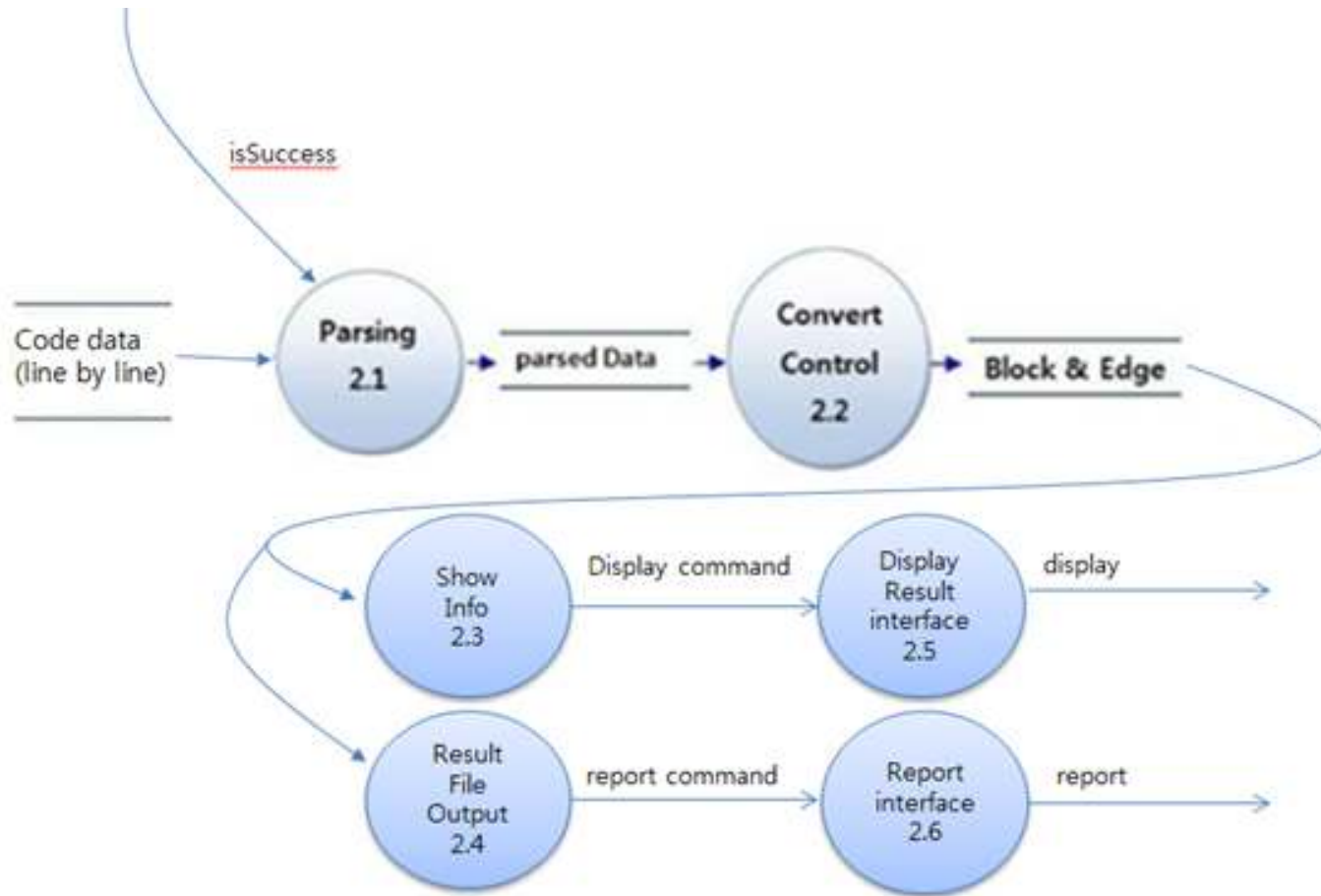


# DFD level 2 (validity&permission control)

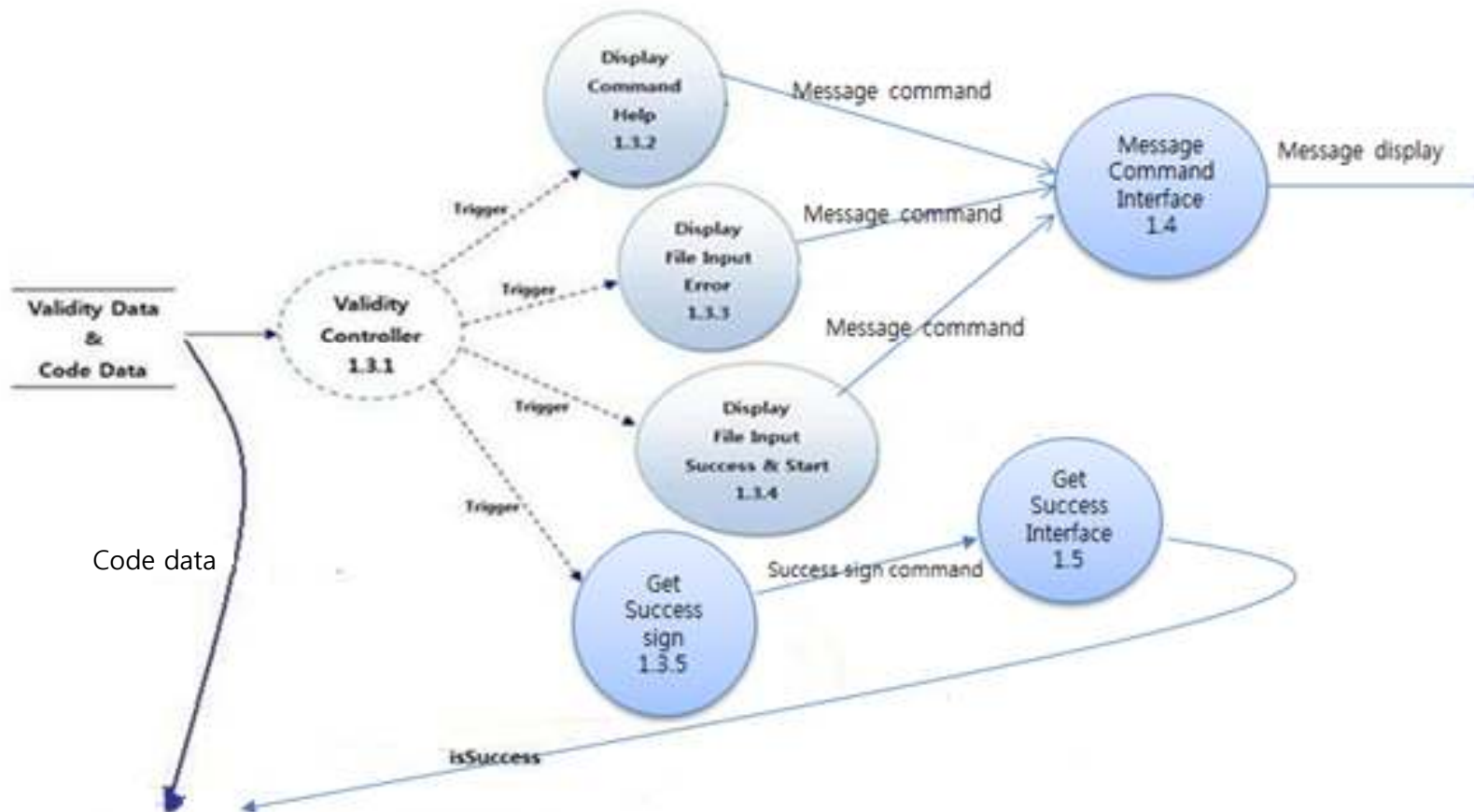




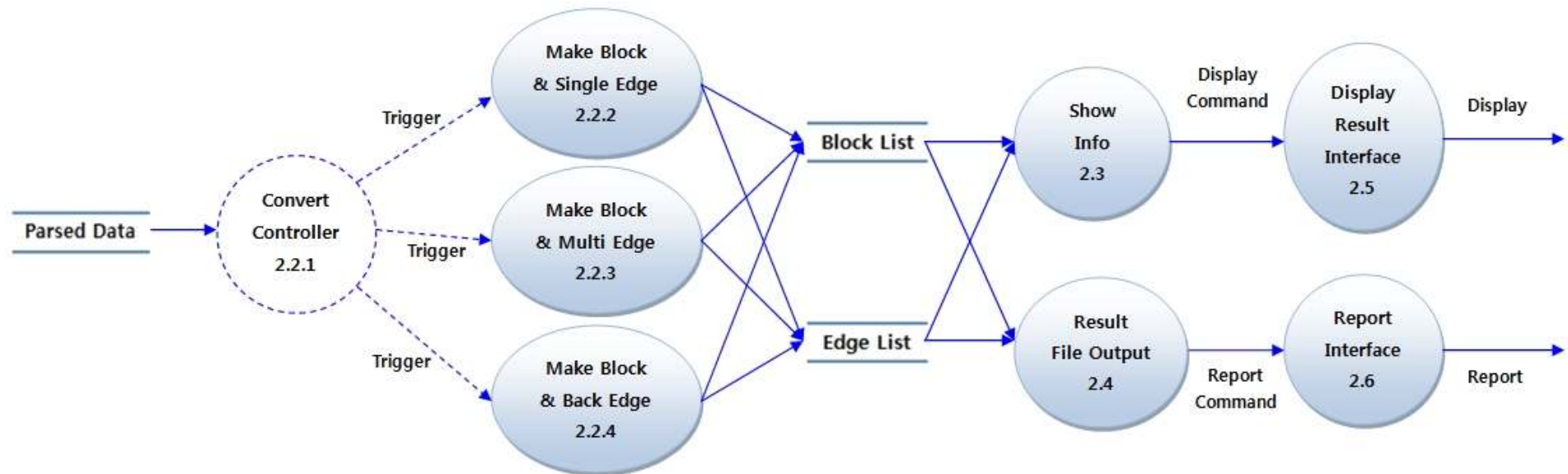
# DFD level 2 (CFG convert control)



# DFD level 3 (validity control)

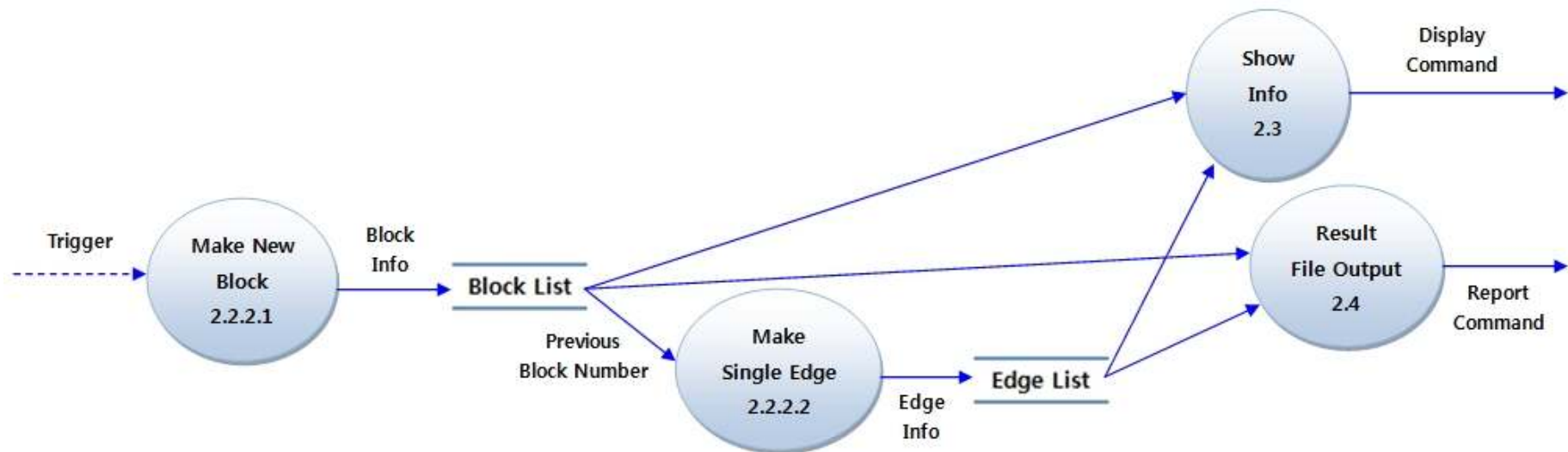


# DFD level 3 (convert control)



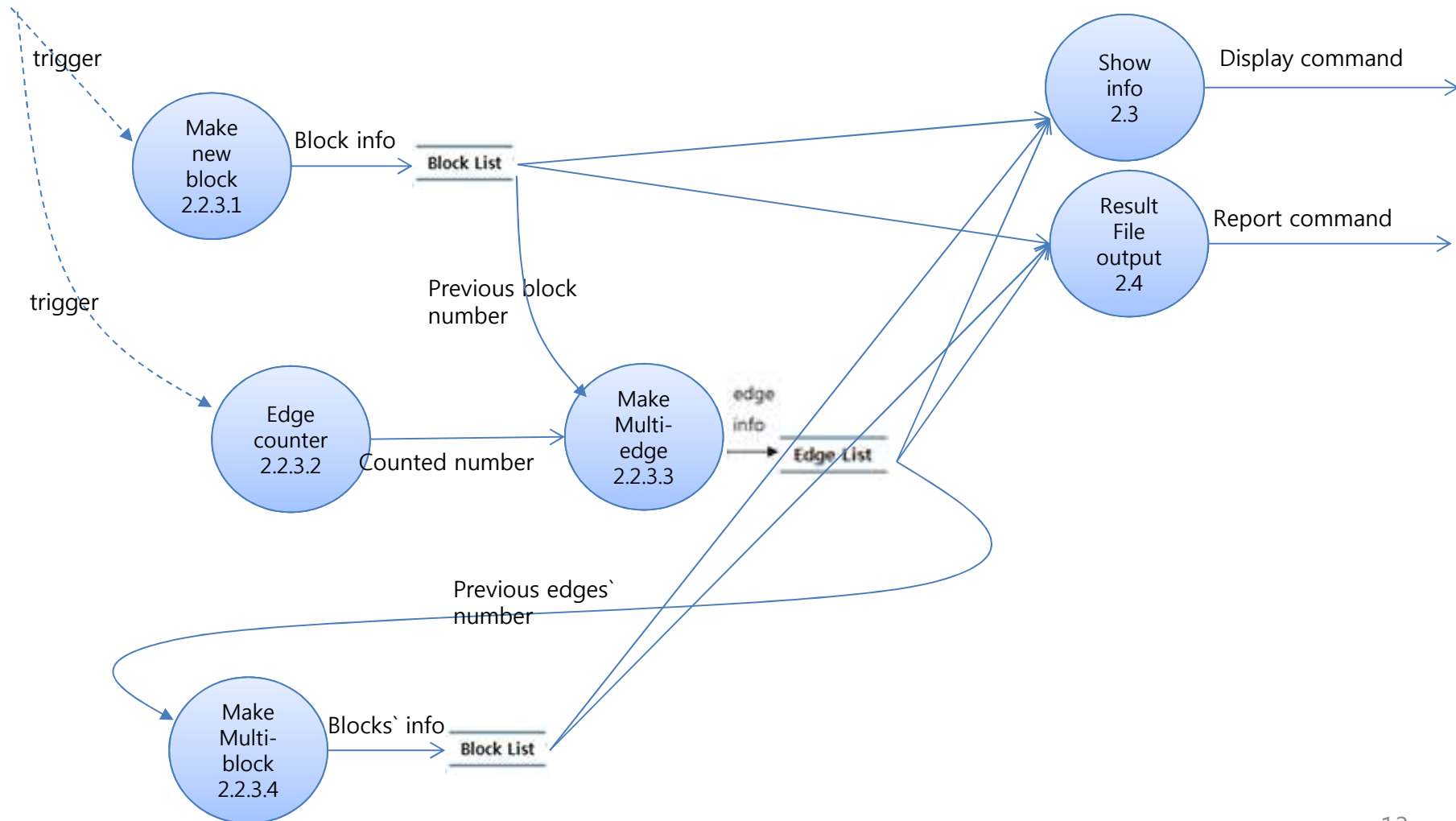
# DFD level4

(make block & single edge 2.2.2)



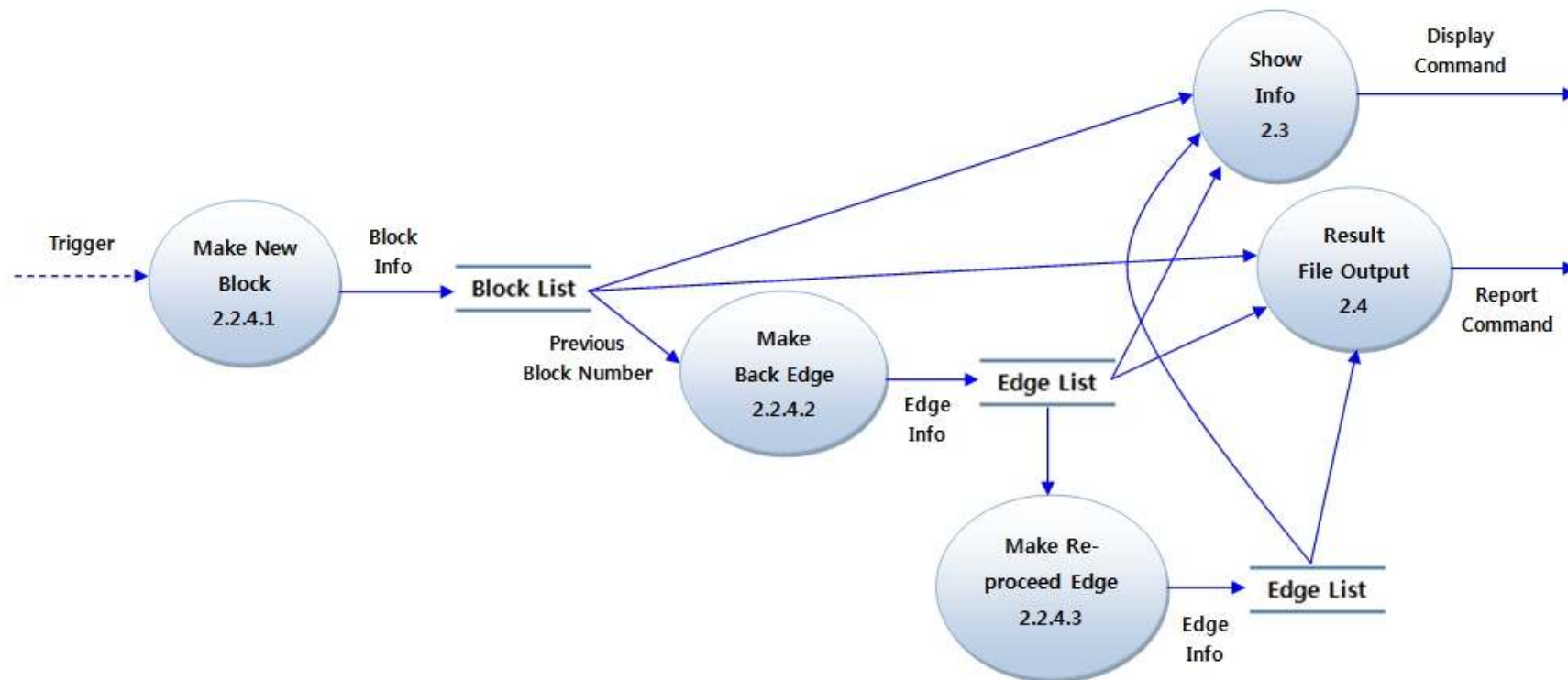
# DFD level4

(make block & multi edge 2.2.3)

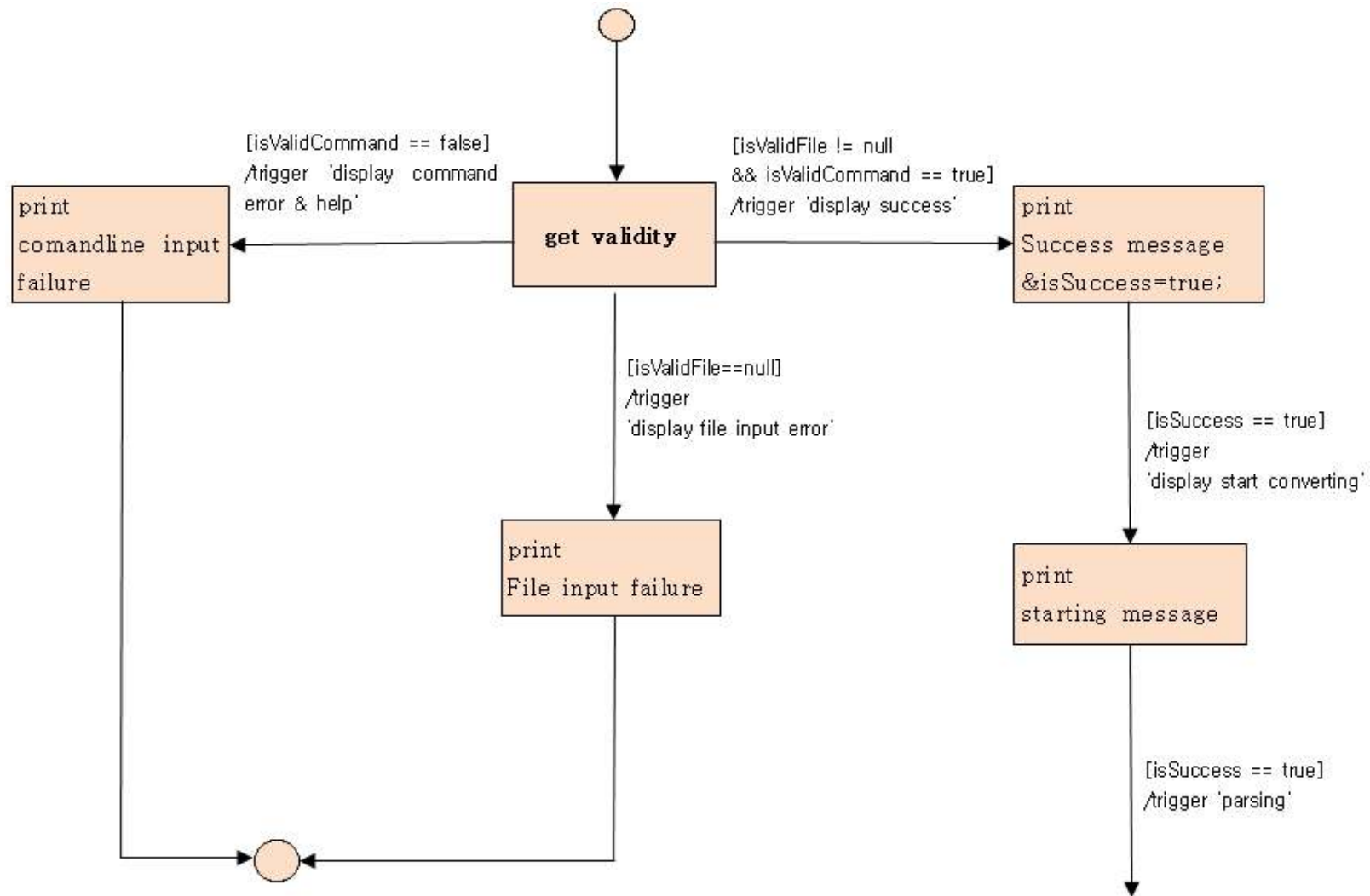


# DFD level4

(make block & back edge 2.2.4)



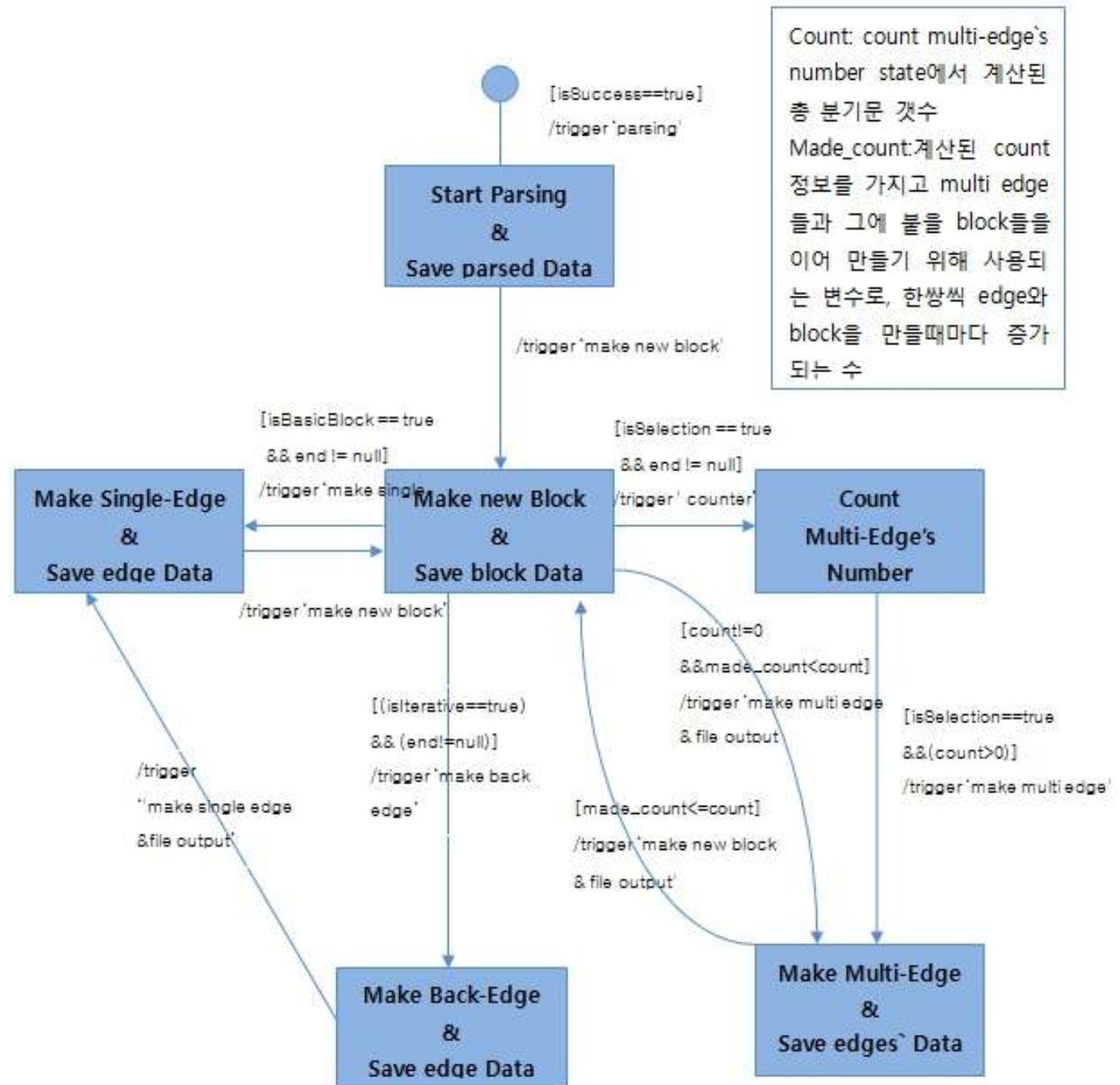
# DFD Level 4 – validity controller



# DFD

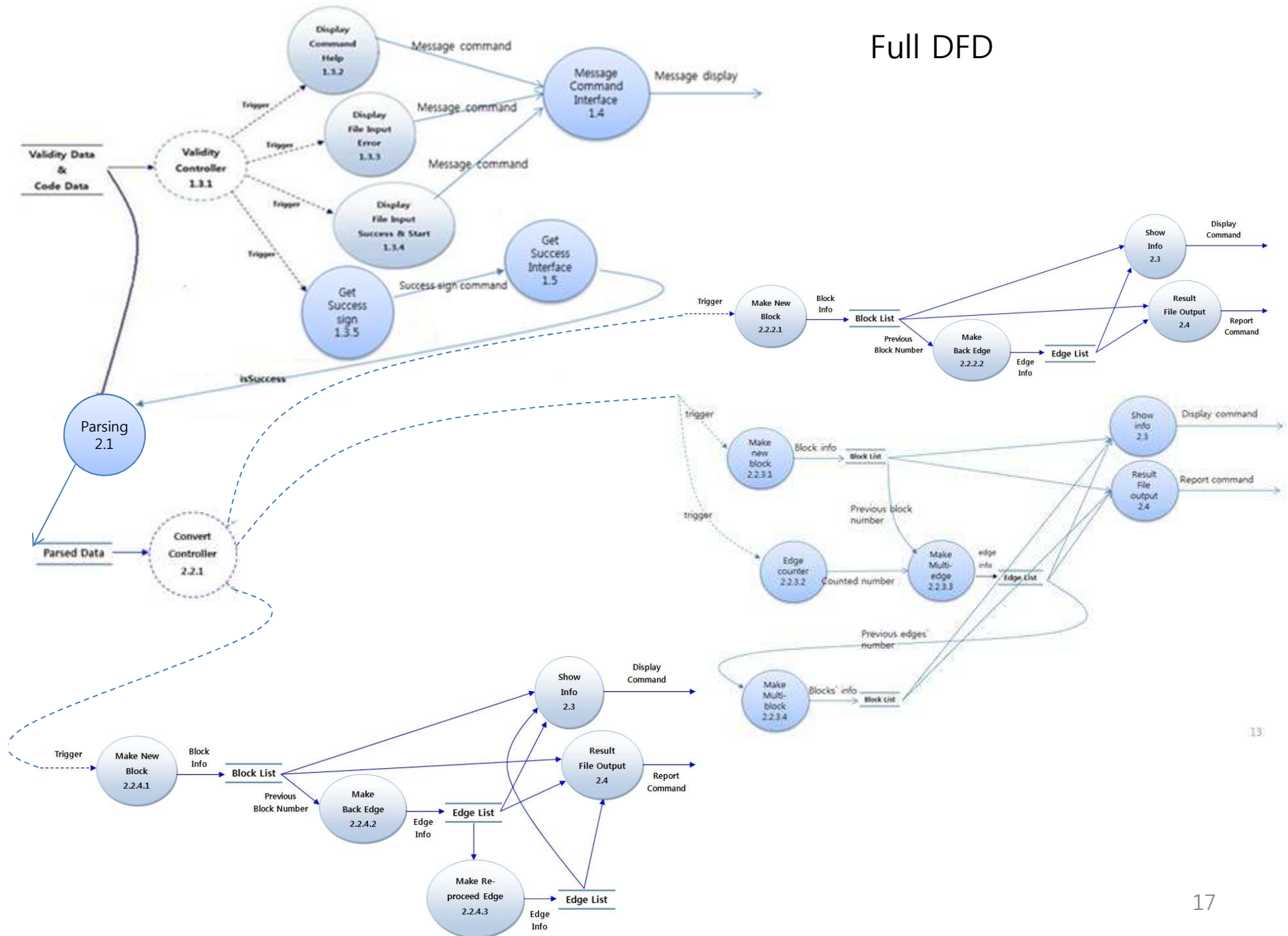
## Level 4

- convert controller





# Full DFD



| Data Name            | Description  | Type  |
|----------------------|--|---|
| Command Line         | 사용자로 부터 Command Reader로부터 받아들이 명령어 문장  | String  |
| C Code File          | C Code Reader로부터 받아들이 C Source Code  | File(*.c)   |
| Report               | Block list와 Edge list에 저장된 정보들을 파일 출력을 통해 실제 report.txt파일에 실질적으로 쓰기 위한 데이터                 | File(*.txt)   |
| Display              | Text 형식으로 생성한 CFG 목록을 실질적으로 Console에 출력할 하기 위한 데이터   | String  |
| Message Command      | Help, Error, Success, Start 등의 에러 및 성공 여부Message출력을 위한 command                             |   |
| Success Sign Command | get Success Interface프로세스에 보내지는 ,성공적으로 파일과 명령어를 읽어들이었다는 true /false 값을 얻기 위한 명령            |   |
| Message Display      | 위 Message Command에 의해 들어온 출력해야할 Message들을 실질적으로 출력   |   |
| Report command       | Report interface에 보내지는 command로 최종적으로 파일 출력을 통해 각 block,edge 정보를 가지고 report 파일 작성을 위한 명령   |   |
| Display command      | Display result interface에 보내지는 command로 이전 생성한 block과 edge정보들을 가지고 이 정보들을 보여주기 위해 내보내지는 명령 |   |
| Block Data           | 생성된 Block들이 가지고 있는 해당 블록 만의 데이터로, 생성된 block number, 블록안에 들어갈 코드내용을 데이터로 가진다                | Integer, String<br>[Block #,<br>Contents]                 |
| Edge Data            | 생성된 Edge들이 가지는 해당 edge들만의 데이터로, 생성된 edge들의 번호, 해당edge의 시작 Block, 도착 Block의 정보를 가진다         | All Integers<br>[Edge #, Start<br>Block #,<br>Destination |

# Data Dictionary

| Data Name      | Description   | Type                 |
|----------------|---|----------------------|
| isValidCommand | 사용자가 입력한 command line이 유효한 명령어였다면 true가, 아니면 false가 저장된다  | Boolean              |
| isValidFile    | 존재하지 않거나 유효하지 않는 파일이라면 null을, 또는 C source code file 이 파일입출력을 통해 모두 잘 읽어들이었다면 end pointer 값을 가지는 file pointer                        | FILE<br>*isValidFile |
| isSuccess      | 받아들인 Command Line과 C Code File의 입력이 모두 정상적으로 이루어졌을 때 CFG로의 변환 시작을 알리고 다음 변환 단계로 넘어가기 위해 최종적으로 Success or fail를 나타내는 true/false data | Boolean              |
| isBasicBlock   | Parsing을 마친 Data가 분기문이나 반복형태의 문장이 아닌 일반 문장이라면 basic block이라 판단, 이에 대한 true or false값이 저장되어 있는 데이터                                   | Boolean              |
| isSelection    | 해당 Parsed Data가 if-else문, multi-if 또는 switch문이라는 것으로 판단되면, isSelection에 true, 혹은 아닐경우 false 값이 들어가는 데이터                             | Boolean              |
| isIterative    | Parsed Data가 반복형태의 for문 또는 while, do-while문으로 판단되면 isIterative에 true 혹은, 아닐 경우 false값이 들어가는 데이터                                     | Boolean              |
| end            | Parsing이 완료된 parsed data를 CFG로 생성해감에 있어서 해당포인터가 parsed data의 끝인지 아닌지 판단하는 포인터로 끝이면 null, 아니면 그 다음 변환대상 parsed data의 포인터로 쓰인다.       | string<br>pointer    |
| count          | CFG를 생성할 다음 Parsed Data가 분기문이라면, 즉 isSelection==true일 때, 그 분기 개수에 대한 데이터  | integer              |
| made_count     | 분기문 형태를 CFG로 변환 할 시 처음 0으로 초기화 된 후 multi edge가 생성될때 마다 해당 edge-block을 한쌍씩 완성할때마다 count하는 횟수   | integer              |

# Data Dictionary

| Data Store Name                         | Description  |
|---|--|
| Validity Data & Code Data(line by line) | C Code File을 한 줄씩 읽어 들여 저장한 code Data들을 저장한 공간과, 읽어들이 결과 Interface에서 받아들이 각각의 command line과 c code file의 유효성을 판단의 결과 그 여부의 True/False값을 가진다  |
| Parsed Data                             | 위 code data 자료저장소로 부터 한줄씩 읽어들이었던 code line들을 가지고 Parse 프로세서 안에서 해당 코드를 구문분석한 결과들을 저장한 정보 자료저장소로 각각 일반 블록으로서의 단일 분기, 다중 선택 분기인지 또는 루프를 생성하는 반복 분기인지 등에 관한 정보를 true/false형태로 저장한 모음이다. (isBasicBlock, isSelectionEdge, isIterationEdge로 나뉘어 저장). |
| Block List                              | 변환 과정중에 생성되는 모든 Block들의 번호와 그 Block의 내용을 묶어 저장하는 곳으로 변환과정 중 생성되는 순서를 따라 쌓여있는 해당 Block들의 List를 의미한다.<br>B[Block #, Contents]  |
| Edge List                               | 생성된 Edge의 번호와 그 Edge 상위에 연결된 Block 번호, 하위에 연결된 Block 번호를 묶은 구조로 변환 과정중 생성된 모든 엣지들에 대해 List형태로 저장되어 있다.<br>E[Edge #, Start Block #, Destination Block #]  |

# Process Specification

| Process No. | 1.1  |
|-------------|--|
| Name        | Command Line Interface   |
| Input       | Command Line   |
| Output      | isValidCommand   |
| Description | Command Line을 받아서, 올바른 형태의 명령어를 입력했을 시에는 isValidCommand에 True를, 잘못된 형태의 명령어를 입력했을 시에는 false를 저장한다. |

| Process No. | 1.2  |
|-------------|--|
| Name        | File Input Interface   |
| Input       | C Code File  |
| Output      | isValidFile  |
| Description | C Code File을 입력값으로 받아서, C Code를 한줄씩 읽어들이는 것이 문제없이 진행되었다면 isValidFile에 해당 string열에 대한 결과를 저장한 포인터를 받고, 만약 Code line 입력이 실패하였다면 isValidFile pointer에 null값을 리턴하게 한다. |

# Process Specification

| Process No. | 1.3  |
|-------------|--|
| Name        | Validity Control   |
| Input       | Validity Data  |
| Output      | Message Command, Success Sign Command  |
| Description | Command Line의 유효성과 C Code File의 유효성에 대한 정보 (isValidCommand & isValidFile )를 받아 그에 따른 성공, 실패, 도움말 여부 등에 관한 Message를 출력할 수 있게 하고, 다음 Convert control를 수행하기 위한 isSuccess, 즉 최종 변환준비가 모두 갖추어 졌는지에 대한 데이터를 내보내게 한다. |

| Process No. | 1.3.2   |
|-------------|---|
| Name        | Display Command Help  |
| Input       | Trigger   |
| Output      | Message Command   |
| Description | Validity Controller로부터 Trigger를 받아서, isValidCommand 값이 False이면 도움말을 출력한다. |

# Process Specification

| Process No. | 1.3.3  |
|-------------|--|
| Name        | Display File Input Error   |
| Input       | Trigger  |
| Output      | Message Command  |
| Description | Validity Controller로부터 Trigger에 의해서 호출되며, isValidFile 값이 null이면 Error Message를 출력하고 Program을 종료한다. |

| Process No. | 1.3.4   |
|-------------|---|
| Name        | Display File Input Success & Start  |
| Input       | Trigger   |
| Output      | Message Command   |
| Description | Validity Controller로부터 Trigger를 받아서, isValidCommand 값과 isValidFile 값이 각각 True이고 null이 아니라는 두가지 조건이 만족되면 C Code 입력이 성공했다는 Message를 보이고, CFG로의 변환을 시작한다는 Message를 출력한다. 그리고 isSuccess에 True값을 지정해 준다. |

# Process Specification

| Process No. | 1.3.5  |
|-------------|--|
| Name        | Get Success Sign   |
| Input       | trigger  |
| Output      | Success sign command   |
| Description | Validity controller로 부터 isValidFile==true 이고 isValidCommand==true일때, 즉 파일 읽어들이기에 성공함과 동시에 유효한 command line이 읽어들이었을 경우 trigger되는 프로세스로 ,그 다음 프로세스에서 CFG변환을 위한 준비가 완료되었다는 success sign을 얻기 위해 success sign command를 내보내는 프로세스이다. |

| Process No. | 1.4  |
|-------------|--|
| Name        | Message Interface  |
| Input       | Message Command  |
| Output      | Message Display  |
| Description | Command Help, File Input Error, File Input Success & Start 라는 message의 출력을 위한 Command를 받는 프로세스로, 실질적으로 이를 출력하는 message display데이터를 내보낸다. |



| Process No. | 1.5  |
|-------------|--|
| Name        | Get Success Interface  |
| Input       | Success Sign Command   |
| Output      | isSuccess  |
| Description | Get Success sign 라는 이전 프로세스로 부터 success sign command를 받을 경우 isSuccess에 변환준비가 완료되었다는(command line과 파일 입력 성공) 의미인 true값을 대입하여 해당 변수를 내보내는 프로세스 |

| Process No. | 2.1   |
|-------------|---|
| Name        | Parsing   |
| Input       | isSuccess, Code Data(Line By Line)  |
| Output      | Parsed Data   |
| Description | isSuccess를 True로 받았을 경우에만, 자료저장소인 Validity Data & Code Line Data로부터 Code Data(line by line) 이 저장된 저장소부분만을 참조하여 구문 분석을 하고 , 그 결과에 따라 isBasicBlock, isSelection, isIteration에 각각의 여부에 대한 값을 True/False로 저장한다. |

# Process Specification

| Process No. | 2.2  |
|-------------|--|
| Name        | Convert Control  |
| Input       | Parsed Data(isBabicBlock, isIterative, isSelection)  |
| Output      | Block & Edge List  |
| Description | Code data의 구문분석이 완료된 parsed data 를 이용해 다음 변환하고자 하는 구문이 반복문이거나 선택문인지, 또는 basic block으로 만들 평범한 문장인가에 따라 각각 block으로의 생성과 그에 따른 edge생성 및 연결을 하는 control로 report.txt파일에 출력을 하여 이를 기록으로 남기는 control이다. |

| Process No. | 2.2.2   |
|-------------|---|
| Name        | Make Block & Single-Edge  |
| Input       | Trigger(triggered by the parsed data, isBasicBlock)   |
| Output      | Block Data, Edge Data   |
| Description | isBasicBlock==True일때 trigger 되는 프로세스로, 기본블럭으로 만들 일반 statement인 경우 해당 구문을블록화 한 후, 해당 블록 번호를 받아 단일 엣지를 연결하여 해당 블록과 엣지의 정보를 내보낸다.. |

# Process Specification

| Process No. | 2.2.3  |
|-------------|--|
| Name        | Make Block & Multi-Edge  |
| Input       | Trigger  |
| Output      | Block Data, Edge Data  |
| Description | isSelectionEdge==T일때 trigger되는 프로세스로, 구문 분석을 해보았을 때 다중 분기문으로 분석되었을 경우 해당 조건문 안을 블럭화 한 후, 다중 분기 엣지를 생성한다. |

| Process No. | 2.2.4   |
|-------------|---|
| Name        | Make Block & Back-Edge  |
| Input       | Trigger   |
| Output      | Block Data, Edge Data   |
| Description | isIterationEdge==T일때 trigger되는 프로세스로, 구문 분석 후 반복문 형태의 코드로 판단되었을 때, 해당 조건문 안의 구문을 블럭화 하고, 루프 구조를 생성할 수 있는 back edge형태의 edge를 연결한다. |

# Process Specification

| Process No. | 2.2.2.1 & 2.2.3.1 & 2.2.4.1  |
|-------------|--|
| Name        | Make New Block   |
| Input       | Trigger  |
| Output      | Block Info [block#, 해당 블록의 내용]   |
| Description | Parsed data 가 isSelection==T, isIterative==T, 또는 isBasicBlock==T인지에 따라 각각 새로운 block 을 생성하고 해당 블록에 대한 정보를 그대로 내보낸다. 형식은 [블록 넘버, 그 블록이 가진 statement 내용]이다. |

| Process No. | 2.2.2.2 & 2.2.3.2 & 2.2.3.7 & 2.2.4.2   |
|-------------|---|
| Name        | Show Block Info & File Output   |
| Input       | Block Info[block#, 해당 블록의 내용]   |
| Output      |   |
| Description | 블록을 생성할 때 마다, 변환과정을 나타내기 위해 생성 이후 해당 블록의 정보를 보여해주고(show block Info), 바로 report.txt파일로의 출력을 하여 기록하기 위한 파일 출력 프로세스이다. |

# Process Specification

|             |  |
|-------------|--|
| Process No. | 2.2.2.3  |
| Name        | Make Single Edge   |
| Input       | Previous Block Number  |
| Output      | Edge info  |
| Description | 바로 이전에 생성된 블록에 연결될 edge를 만드는 프로세스로, 이전 블록이 isBasicBlock==T라는 조건 하에 생성되었으므로 이에 연결될 edge를 하나 만드는 프로세스로, 이전블록 number를 해당 엣지 정보로 가질수 있게 하는 프로세스이다. |

|             |   |
|-------------|---|
| Process No. | 2.2.2.4 & 2.2.3.5 & 2.2.4.5   |
| Name        | Show Edge Info & File Output  |
| Input       | Edge info   |
| Output      |   |
| Description | Edge가 새로 생성 될때마다 해당 edge가 가진 정보, [고유 number#, 상위 연결 블록#, 하위 연결 블록#]를 보여주고 이를 바로 파일출력을 통해 report.txt에 저장하는 프로세스이다. |

# Process Specification

|             |   |
|-------------|---|
| Process No. | 2.2.3.3   |
| Name        | Edge Counter  |
| Input       | Trigger(triggered by convert controller when isSelection==T&&!end)  |
| Output      | Edge Info   |
| Description | 이 counter라는 프로세스는, isSelection==T라는 조건하에 convert control에 의해 들어오게 된 프로세스로, 다중 선택문의 경우 일 때 해당 선택 분기 조건이 몇 개로 나뉘는지 알아내어 분기 개수를 counting하는 프로세스이다. |

|             |   |
|-------------|---|
| Process No. | 2.2.3.4   |
| Name        | Make Multi-Edge   |
| Input       | Previous Block Number, counted number   |
| Output      | Edge Info   |
| Description | isSelection==T라는 조건에 있을 때(다중 선택 구조) 사용되는 프로세스로, 같은 조건일때 counter프로세스를 거쳐 알아낸 분기갯수를 이용해 해당 개수만큼 edge를 생성하고 이전에 생성된 block의 번호를 상위 블록 정보로 가지는, 여러 선택의 경우에 대한 edge를 그리기 위한 프로세스이다. |

# Process Specification

| Process No. | 2.2.3.6  |
|-------------|--|
| Name        | Make Multi-Block   |
| Input       | Edge info  |
| Output      | Blocks' Info   |
| Description | 바로 이전 make multi-edge과정에 의해 생성된 여러 엣지에 연결될 블록들을 생성하는 프로세스로, 이전 엣지들에 대한 정보를 가지고 연결된 블록들을 만들어 생성한 block의 정보를 내보낸다. |

| Process No. | 2.2.4.3  |
|-------------|--|
| Name        | Make Back-Edge   |
| Input       | Previous Block Number  |
| Output      | Edge Info  |
| Description | isIterative==T인 경우(반복구조인 경우) 새로운 block을 생성한 이후 거치게 되는 프로세스로, 새로 생성된 block에서 나와 다시 반복적으로 돌아가는 형태의 edge를 생성하여 해당 edge정보를 내보낸다. |

# Process Specification

|             |  |
|-------------|--|
| Process No. | 2.2.4.4  |
| Name        | Make Re-proceed Edge   |
| Input       | Back-edge info   |
| Output      | Edge Info  |
| Description | isIterative==T인 경우 바로 이전에 make back-edge프로세스를 통해 만든 백 엣지의 순환고리에서 빠져나오게 된 후 다시 나아가는 경로를 위한 엣지를 생성하는 프로세스로, 반복구조에서 루프 이후 나아가게 되는 엣지를 생성해 이에 대한 정보를 내보낸다. |



# Process Specification

|             |   |
|-------------|---|
| Process No. | 2.3   |
| Name        | Show Info   |
| Input       | Block List, Edge List   |
| Output      | Display Command   |
| Description | Convert Control 과정에 의해 실질적으로 변환된 Text 형태의 CFG를 변환 과정에 지속적으로 콘솔에 출력, 보여주기 위한 프로세스로 Data Store인 Block List, Edge List로 부터 이에 모아지는 각각의 블록, 엣지들의 정보를 보여주기 위해 display command를 내보내주는 프로세스이다. |

|             |  |
|-------------|--|
| Process No. | 2.4  |
| Name        | Result File Output   |
| Input       | Block List, Edge List  |
| Output      | Report Command   |
| Description | CFG생성 과정 중 block list와 edge list 자료 저장소에 모아진 정보들을 파일 출력을 통하여 실제로 report.txt파일에 쓰기 위한 프로세스로, 이 프로세스가 report command를 내보내 다음 프로세스인 report interface에 도달하여 모든 block과 edge의 정보를 report 파일에 저장하기 위한 프로세스이다. |

# Process Specification

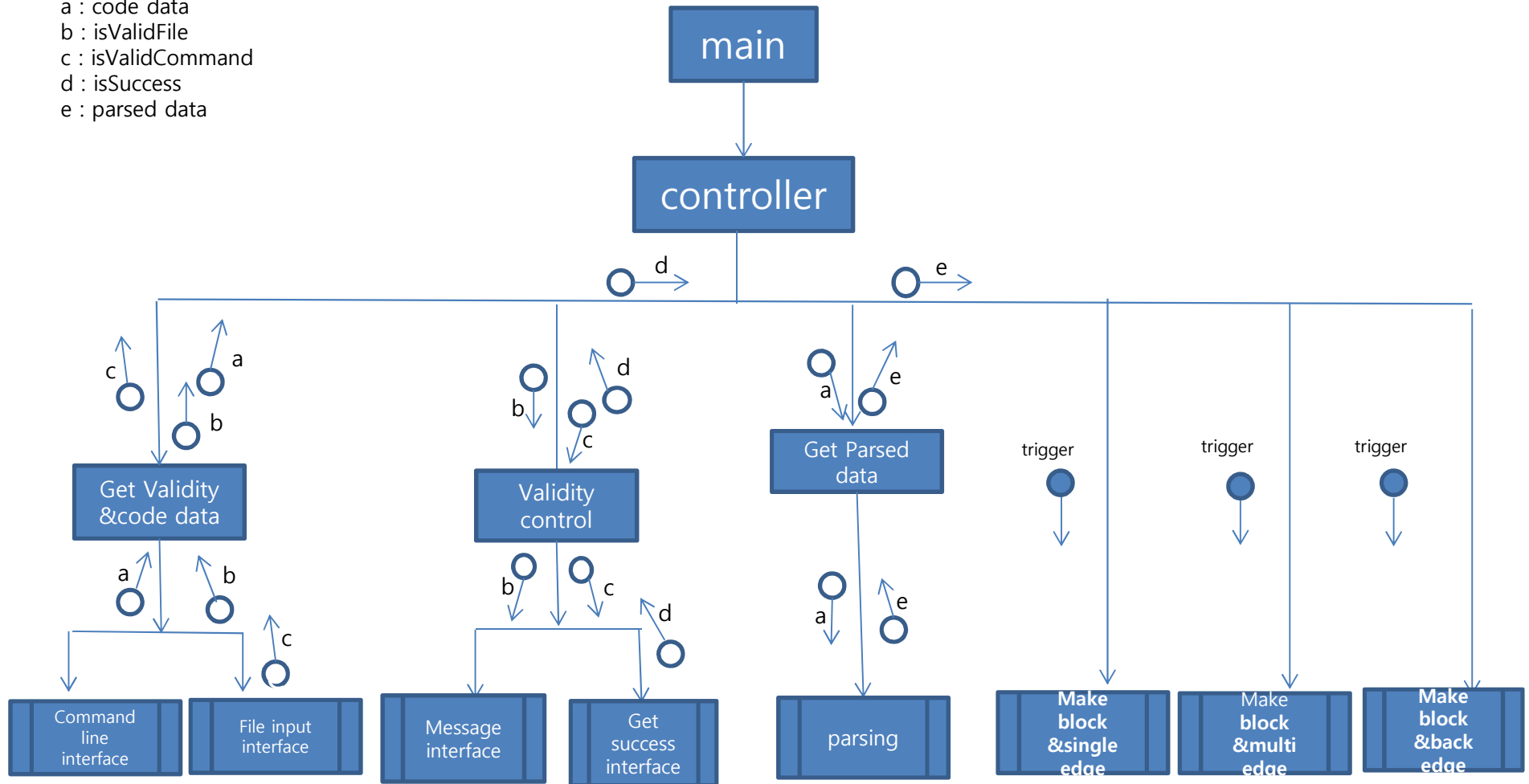
|             |  |
|-------------|--|
| Process No. | 2.5  |
| Name        | Display Result Interface   |
| Input       | Display Command  |
| Output      | Display  |
| Description | 이전 프로세스인 show info가 block list, edge list인 자료저장소로 부터 해당 정보를 출력 하라는, display command를 내보내면 이를 받아 실제로 콘솔에 출력하기 위해 display 를 내보내는 프로세스이다.                   |
| Process No. | 2.6  |
| Name        | Report Interface   |
| Input       | Report Command   |
| Output      | Report   |
| Description | Result file output이라는 연결된 이전 프로세스가 block list, edge list저장소로 부터 출력할 정보들을 가지고 report command를 내보내면 해당 명령을 받아 실제로 report.txt파일에 쓰기 위해 report를 내보내는 프로세스이다. |

# SD – structured charts

1.basic

2.advanced

a : code data  
 b : isValidFile  
 c : isValidCommand  
 d : isSuccess  
 e : parsed data



a : code data  
 b : isValidFile  
 c : isValidCommand  
 d : isSuccess  
 e : parsed data

