# Programming Guidelines for FBD programs in Reactor Protection System

Sejin Jung, Dong-Ah Lee, Eui-Sub Kim, JunBeom Yoo and Jang-Soo Lee
Dependable Software Laboratory
Konkuk University, Republic of Korea

# Contents

- Introduction
- Background
- Guidelines for FBD programming
  - Guidelines
  - FBDChecker
    - Case study
- Conclusion

# Introduction

- Safety critical systems are using FBD (Function Block Diagram) to design software
  - It used PLC (Programmable Logic Controller) programming language in plant automation industry

- FBD has several elements of making errors by human errors
  - Guidelines for reducing errors is needed

- Several guidelines for FBD programming exist
  - There are Some kinds of elements which need to modify and specify
  - We propose refine and added guidelines for FBD programming

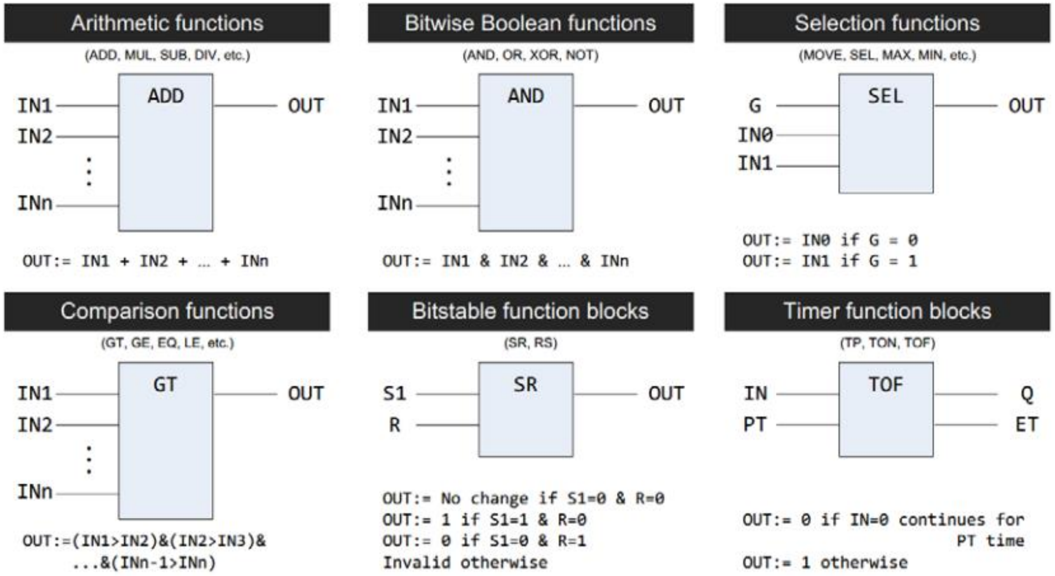
# Introduction

- CASE tool : FBDChecker
  - It check FBD programs for finding violations about guidelines
  - It uses standard input format of FBD
    - Standard XML format of FBD (PLCopen)
- Case study about FBDChecker
  - Example : 5 logics in BP of RPS
    - Finding violations in programs

# Background – Function Block Diagram

- Function Block Diagram defined in IEC 61131-3 standard
  - Defined all function blocks and 10 categories

- FBD consists of number of function blocks
  - Interconnections between function blocks



```
OUT:= IN1 + IN2 + ... + INn
```

```
OUT:= IN1 & IN2 & ... & INn
```

```
OUT:= IN0 if G = 0
OUT:= IN1 if G = 1
```

```
OUT:=(IN1>IN2)&(IN2>IN3)&
    ...&(INn-1>INn)
```

```
OUT:= No change if S1=0 & R=0
OUT:= 1 if S1=1 & R=0
OUT:= 0 if S1=0 & R=1
Invalid otherwise
```

```
OUT:= 0 if IN=0 continues for
                      PT time
OUT:= 1 otherwise
```

# Background – safe programming guidelines

- Safe Programming Guidelines
  - Programming guidelines for achieving safety of software
  - MISRA-C for development in automotive industry
  - DO-178B for airborne systems
  - NUREG/CR-6463 for development in nuclear domain
    - Contains IEC 61131-3 programming language, c/c++, Ada, Pascal, PL/M

# Guidelines for FBD programs

- Making rules with two categories
  - Reliability
  - Maintainability

- Reliability
  - Rules about improving dependability and to guarantee correctness about simulation or action of a program

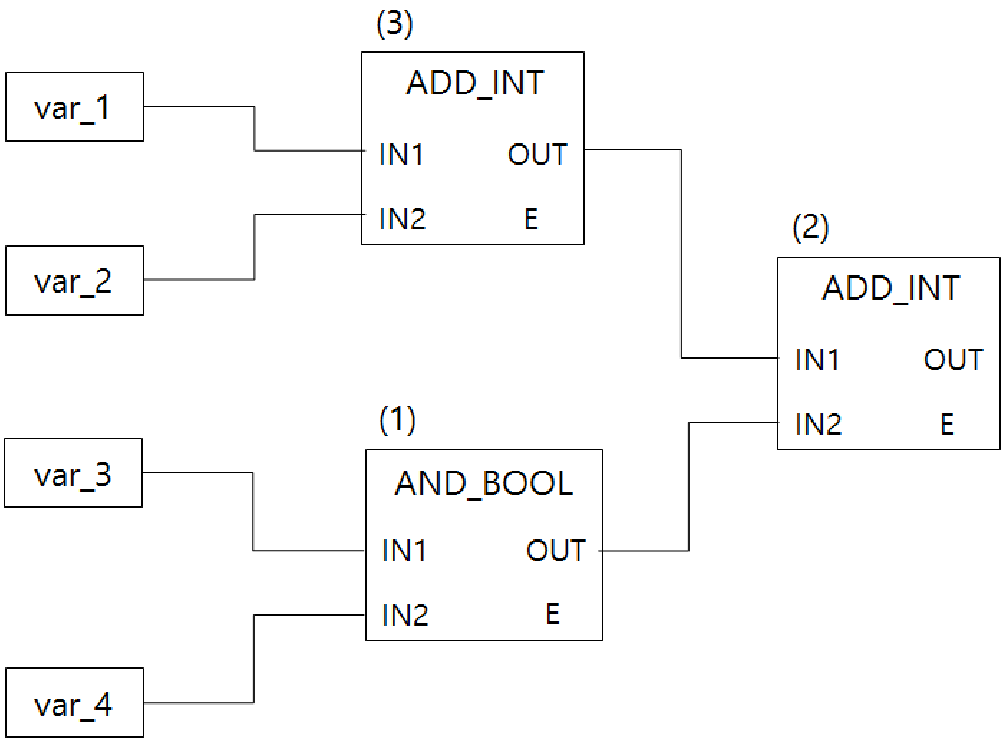
- Maintainability
  - Rules about increasing readability and decreasing complexity

# Guidelines for FBD programs

- Reliability
  - Execution order
    - Using correct execution order
  - Eliminating incorrect move block
    - Connection between move block and function
  - Implicit/explicit type conversion
  - Variable initialization
    - Variable must be initialization before uses
  - Etc.

# Guidelines for FBD programs

- Examples
  - Incorrect execution order

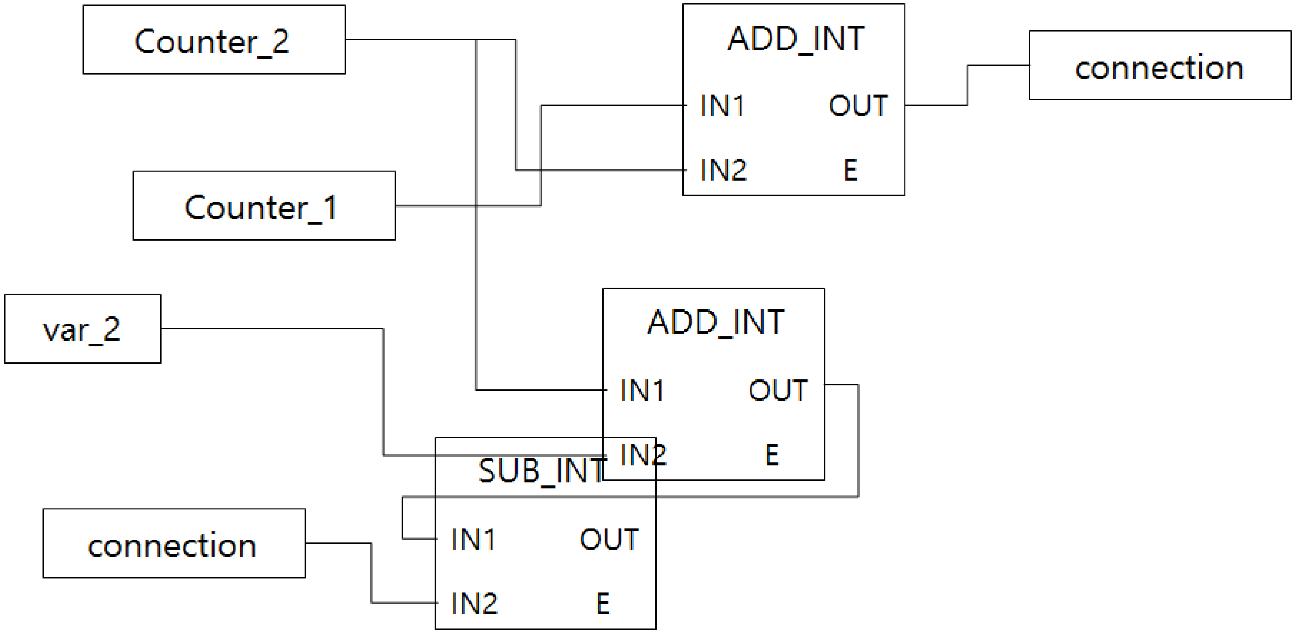


# Guidelines for FBD programs

- Maintainability
  - Naming convention
    - Recommend additional identifier
    - Length – too short, too long
  - Diagram
    - Eliminating crossed lines
    - Eliminating overlapped blocks
  - Etc.

# Guidelines for FBD programs

- Examples
  - Illegible diagram



# Guidelines for FBD programs

- Comparison with existing guidelines and researches

| | FBDChecker | NUREG/ CR-6463 | Research1 | Research2 |
|---|---|---|---|---|
| Target | FBD | FBD | FBD | IEC 61131-3 |
| Diagram | O | O | X | X |
| Data Type | O | O | O | △ |
| Function using | O | △ | X | X |
| Automation | O | - | X | X |
| note | | Need specify | 5 case of guidelines | Target is not just FBD |

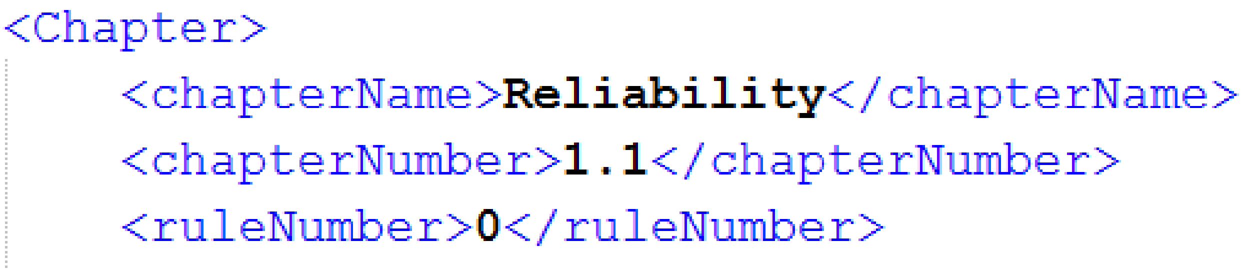
Research1 : Guidelines for the Use of Function Block Diagram in Reactor Protection Systems, accepted APSEC 2014
Research2 : Restricting IEC 61131-3 Programming Languages for use on High Integrity Applications ETFA 2008

# Guidelines for FBD programs

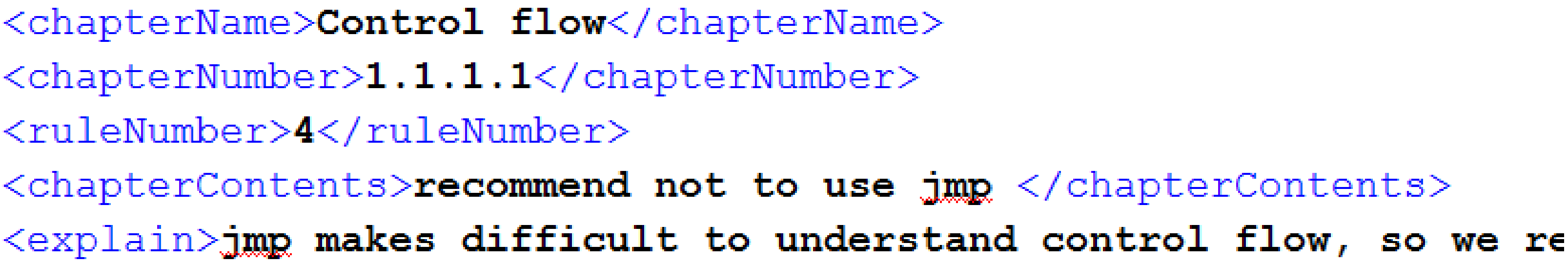
- Classification of rules
  - Two kinds of classification
  - Warnings
    - Rules may have possible to errors
    - Illegible diagram
    - Explicit type conversion
    - Etc.
  - Errors
    - Rules may make critical errors directly
    - Execution order
    - Initialization
    - Implicit type conversion
    - Etc.

# Guidelines for FBD programs

- Compiling a list about guidelines using XML



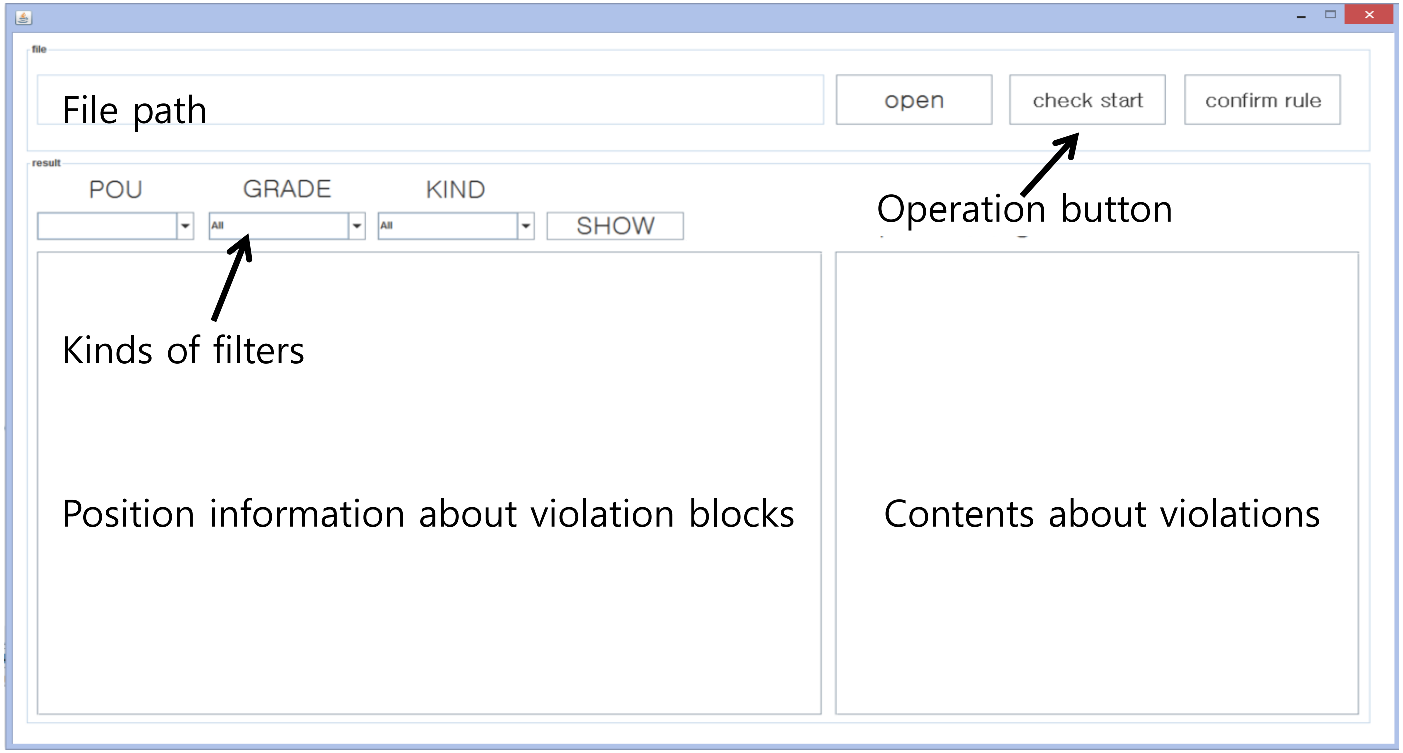
```
<Chapter>
    <chapterName>Reliability</chapterName>
    <chapterNumber>1.1</chapterNumber>
    <ruleNumber>0</ruleNumber>
```



```
<chapterName>Control flow</chapterName>
<chapterNumber>1.1.1.1</chapterNumber>
<ruleNumber>4</ruleNumber>
<chapterContents>recommend not to use jmp </chapterContents>
<explain>jmp makes difficult to understand control flow, so we re
```

# FBDChecker

- CASE tool : FBDChecker
  - Automation tool for checking FBD programs about our guidelines
  - uses standard input format of FBD(PLCopen)
  - checks FBD programs



Operation button

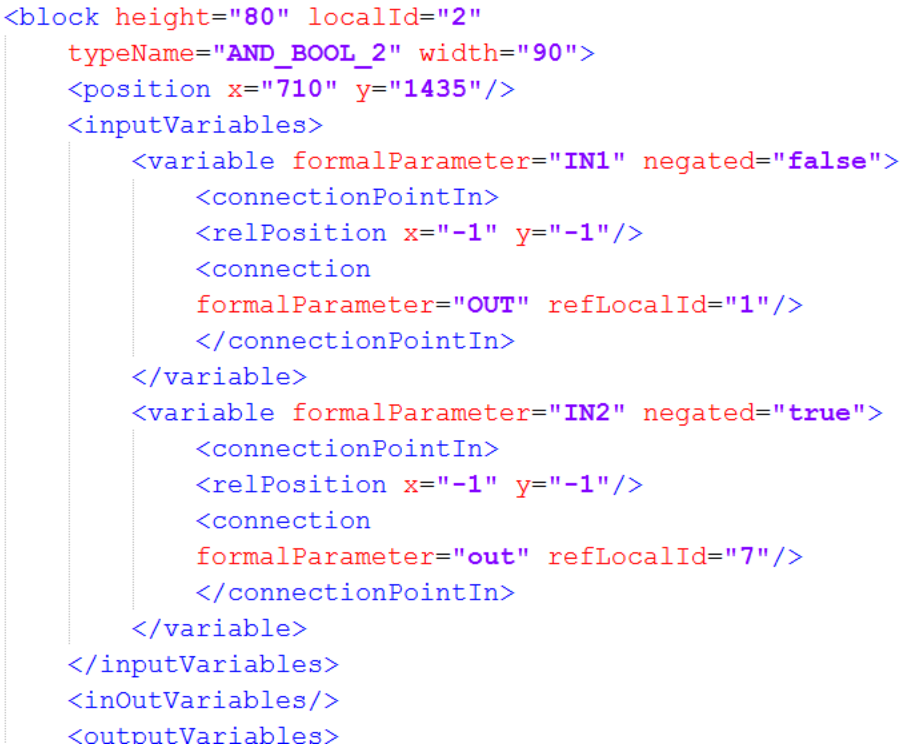
Kinds of filters

Position information about violation blocks

Contents about violations

# FBDChecker

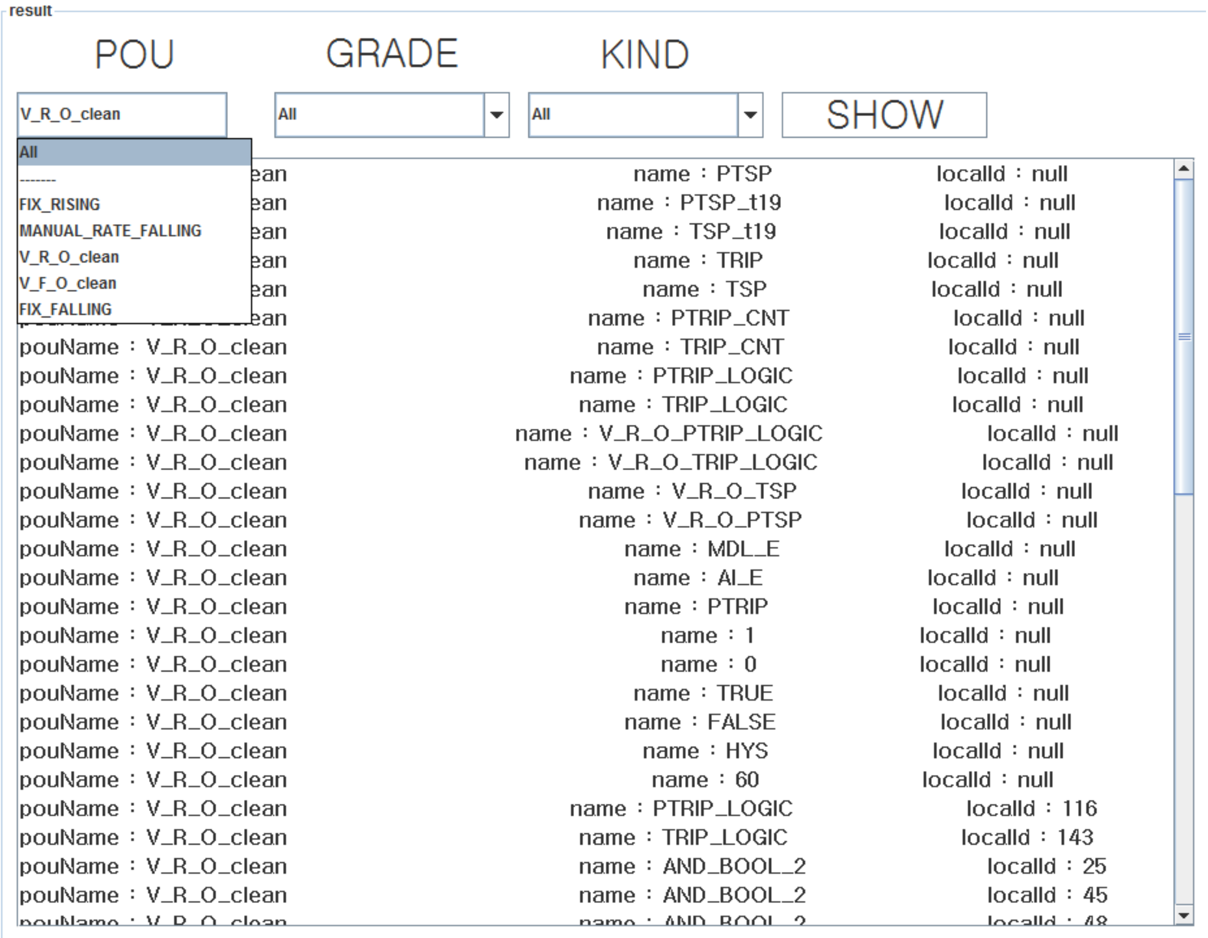
- FBDChecker uses information of FBD programs in XML proposed by PLCopen
  - Parsing xml and searching violations using information about position, type, connection, etc.



```
<block height="80" localId="2"
    typeName="AND_BOOL_2" width="90">
    <position x="710" y="1435"/>
    <inputVariables>
        <variable formalParameter="IN1" negated="false">
            <connectionPointIn>
            <relPosition x="-1" y="-1"/>
            <connection
            formalParameter="OUT" refLocalId="1"/>
            </connectionPointIn>
        </variable>
        <variable formalParameter="IN2" negated="true">
            <connectionPointIn>
            <relPosition x="-1" y="-1"/>
            <connection
            formalParameter="out" refLocalId="7"/>
            </connectionPointIn>
        </variable>
    </inputVariables>
    <inOutVariables/>
    <outputVariables>
```

# Case study

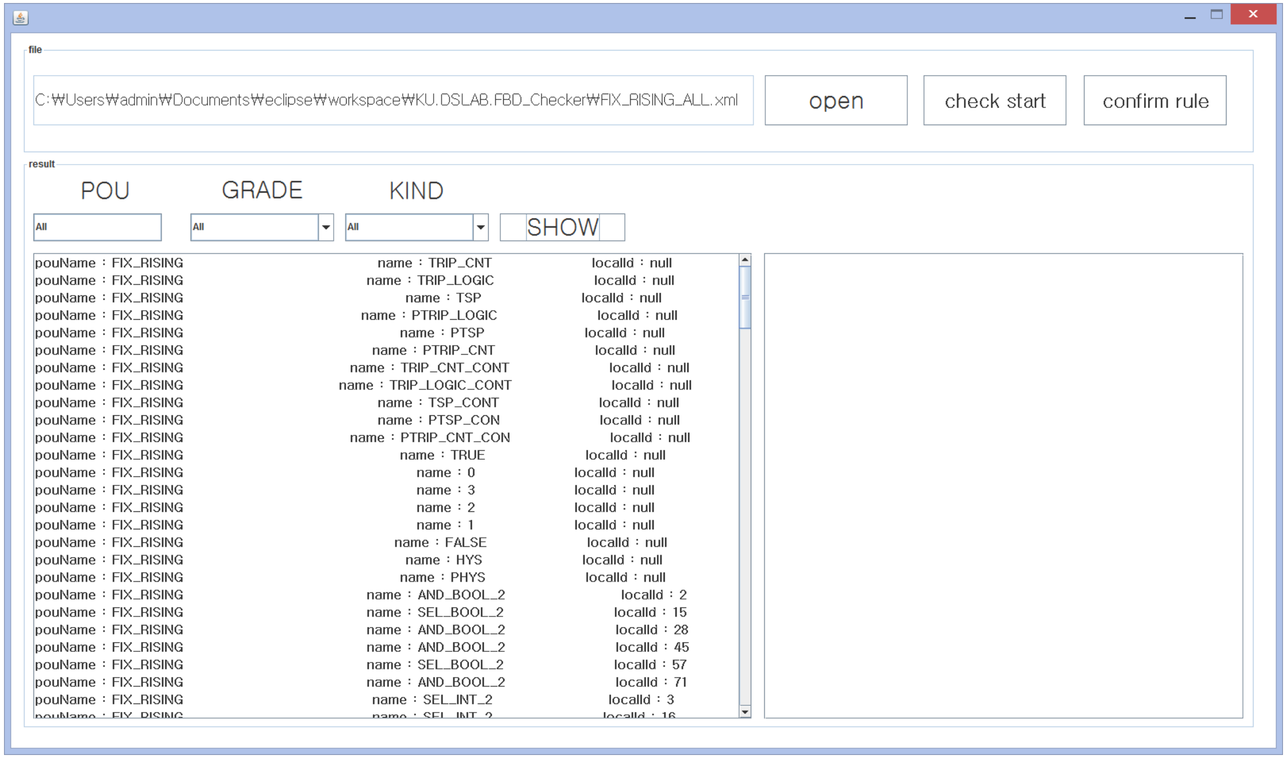
- Filtering screen of POU



```
ean                           name : PTSP                 localId : null
ean                           name : PTSP_t19             localId : null
ean                           name : TSP_t19              localId : null
ean                           name : TRIP                 localId : null
ean                           name : TSP                  localId : null
ean                           name : PTRIP_CNT            localId : null
pouName : V_R_O_clean         name : TRIP_CNT             localId : null
pouName : V_R_O_clean         name : PTRIP_LOGIC          localId : null
pouName : V_R_O_clean         name : TRIP_LOGIC           localId : null
pouName : V_R_O_clean         name : V_R_O_PTRIP_LOGIC    localId : null
pouName : V_R_O_clean         name : V_R_O_TRIP_LOGIC     localId : null
pouName : V_R_O_clean         name : V_R_O_TSP            localId : null
pouName : V_R_O_clean         name : V_R_O_PTSP           localId : null
pouName : V_R_O_clean         name : MDL_E                localId : null
pouName : V_R_O_clean         name : AI_E                 localId : null
pouName : V_R_O_clean         name : PTRIP                localId : null
pouName : V_R_O_clean         name : 1                    localId : null
pouName : V_R_O_clean         name : 0                    localId : null
pouName : V_R_O_clean         name : TRUE                 localId : null
pouName : V_R_O_clean         name : FALSE                localId : null
pouName : V_R_O_clean         name : HYS                  localId : null
pouName : V_R_O_clean         name : 60                   localId : null
pouName : V_R_O_clean         name : PTRIP_LOGIC          localId : 116
pouName : V_R_O_clean         name : TRIP_LOGIC           localId : 143
pouName : V_R_O_clean         name : AND_BOOL_2           localId : 25
pouName : V_R_O_clean         name : AND_BOOL_2           localId : 45
pouName : V_R_O_clean         name : AND_BOOL_2           localId : 48
```

# Case study

- Did case study about 5 logics in BP of RPS
  - Finds 18 kinds of and 264 numbers of violations
    - Type conversion
    - Illegible diagram
    - Naming
    - Etc.



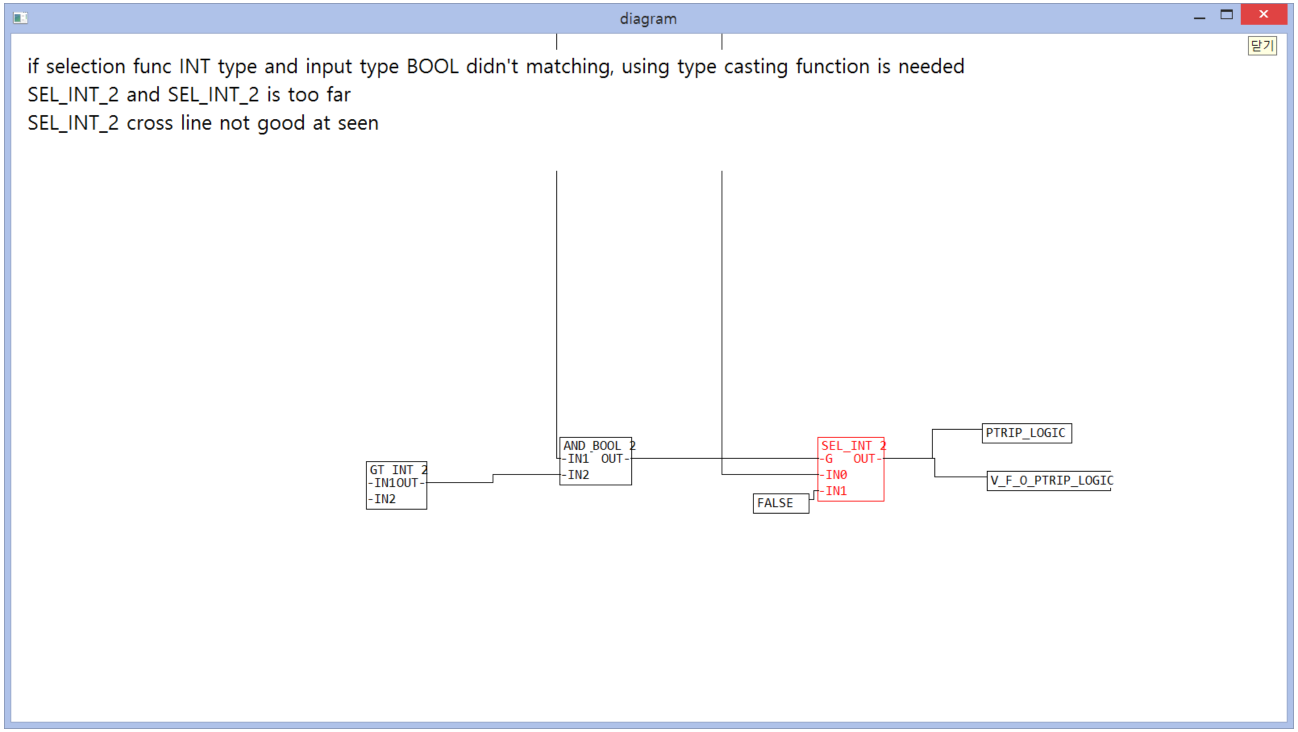
file: C:₩Users₩admin₩Documents₩eclipse₩workspace₩KU.DSLAB.FBD_Checker₩FIX_RISING_ALL.xml

POU: All
GRADE: All
KIND: All

```
pouName : FIX_RISING              name : TRIP_CNT            localId : null
pouName : FIX_RISING              name : TRIP_LOGIC          localId : null
pouName : FIX_RISING              name : TSP                 localId : null
pouName : FIX_RISING              name : PTRIP_LOGIC         localId : null
pouName : FIX_RISING              name : PTSP                localId : null
pouName : FIX_RISING              name : PTRIP_CNT           localId : null
pouName : FIX_RISING              name : TRIP_CNT_CONT       localId : null
pouName : FIX_RISING              name : TRIP_LOGIC_CONT     localId : null
pouName : FIX_RISING              name : TSP_CONT            localId : null
pouName : FIX_RISING              name : PTSP_CON            localId : null
pouName : FIX_RISING              name : PTRIP_CNT_CON       localId : null
pouName : FIX_RISING              name : TRUE                localId : null
pouName : FIX_RISING              name : 0                   localId : null
pouName : FIX_RISING              name : 3                   localId : null
pouName : FIX_RISING              name : 2                   localId : null
pouName : FIX_RISING              name : 1                   localId : null
pouName : FIX_RISING              name : FALSE               localId : null
pouName : FIX_RISING              name : HYS                 localId : null
pouName : FIX_RISING              name : PHYS                localId : null
pouName : FIX_RISING              name : AND_BOOL_2          localId : 2
pouName : FIX_RISING              name : SEL_BOOL_2          localId : 15
pouName : FIX_RISING              name : AND_BOOL_2          localId : 28
pouName : FIX_RISING              name : AND_BOOL_2          localId : 45
pouName : FIX_RISING              name : SEL_BOOL_2          localId : 57
pouName : FIX_RISING              name : AND_BOOL_2          localId : 71
pouName : FIX_RISING              name : SEL_INT_2           localId : 3
pouName : FIX_RISING              name : SEL_INT_2           localId : 16
```

# Case study

- An example of a part of diagram in a logic
  - Too far block
  - Crossed line
  - Type conversion



if selection func INT type and input type BOOL didn't matching, using type casting function is needed
SEL_INT_2 and SEL_INT_2 is too far
SEL_INT_2 cross line not good at seen

# Conclusion & Future Work

- Guidelines
  - We make guidelines which are refined and added
- CASE tool : FBDChecker
  - It uses standard XML format of FBD
  - It finds violations about guidelines which we proposed
- Future Work
  - Implement the improved FBDChecker for expansion easily about guidelines
  - Perform the Case Study about other logics

Q & A

# THANK YOU