Data-Flow Analysis for PLC Programs

Presented by : Sreeja S. Nair
Guided By : Dr. Raoul Jetley, Dr. Anil R. Nair, Dr. S. Ashok

July 2, 2015

Problem Definition

Specified run-time errors need to be detected during development time with the help of static analysis techniques

Introduction

IEC 61131-3 - Standard for the five PLC programming languages

- Structured Text (ST)
- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Ladder Diagram (LD)

Introduction(Contd..)

Probable run-time errors

- Division by zero
- Array access out of bounds
- Use of uninitialized variables
- Unused Variables
- Invariant if condition
- Unreachable code
- Infinite loops

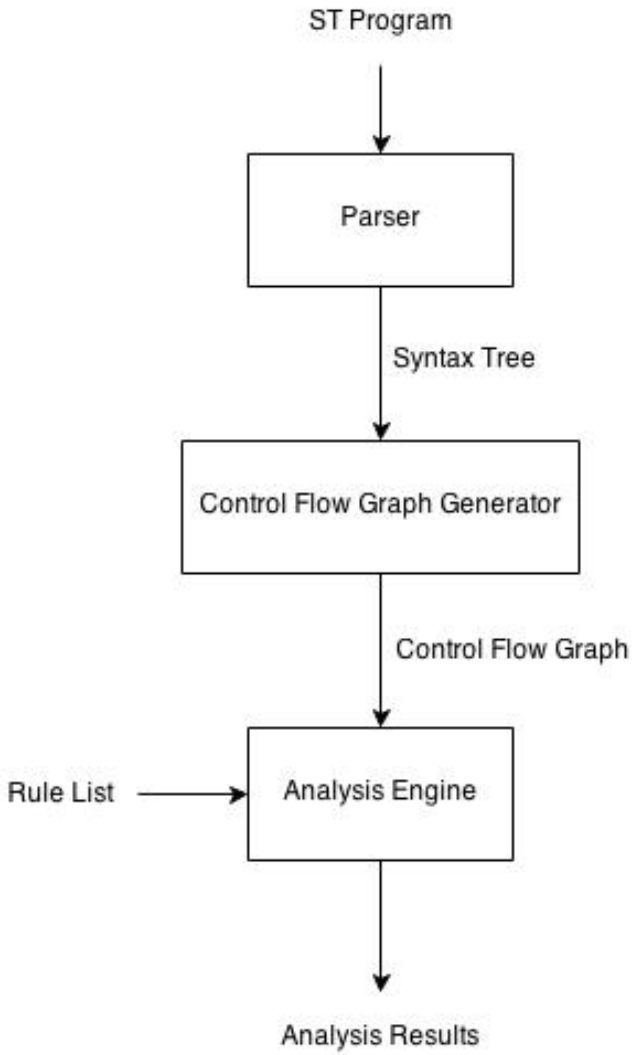
Data-Flow Analysis

- A technique for gathering information about the possible set of values calculated at various points in a computer program.
- A sample data-flow analysis :

P1 : x := 1;
P2 : while (x < 100) do
P3 : x := x + 1;
P4 : end_while;

Program point	Range of x
P1	[1,1]
P2	[1,99]
P3	[2,100]
P4	[100,100]

Tool Structure



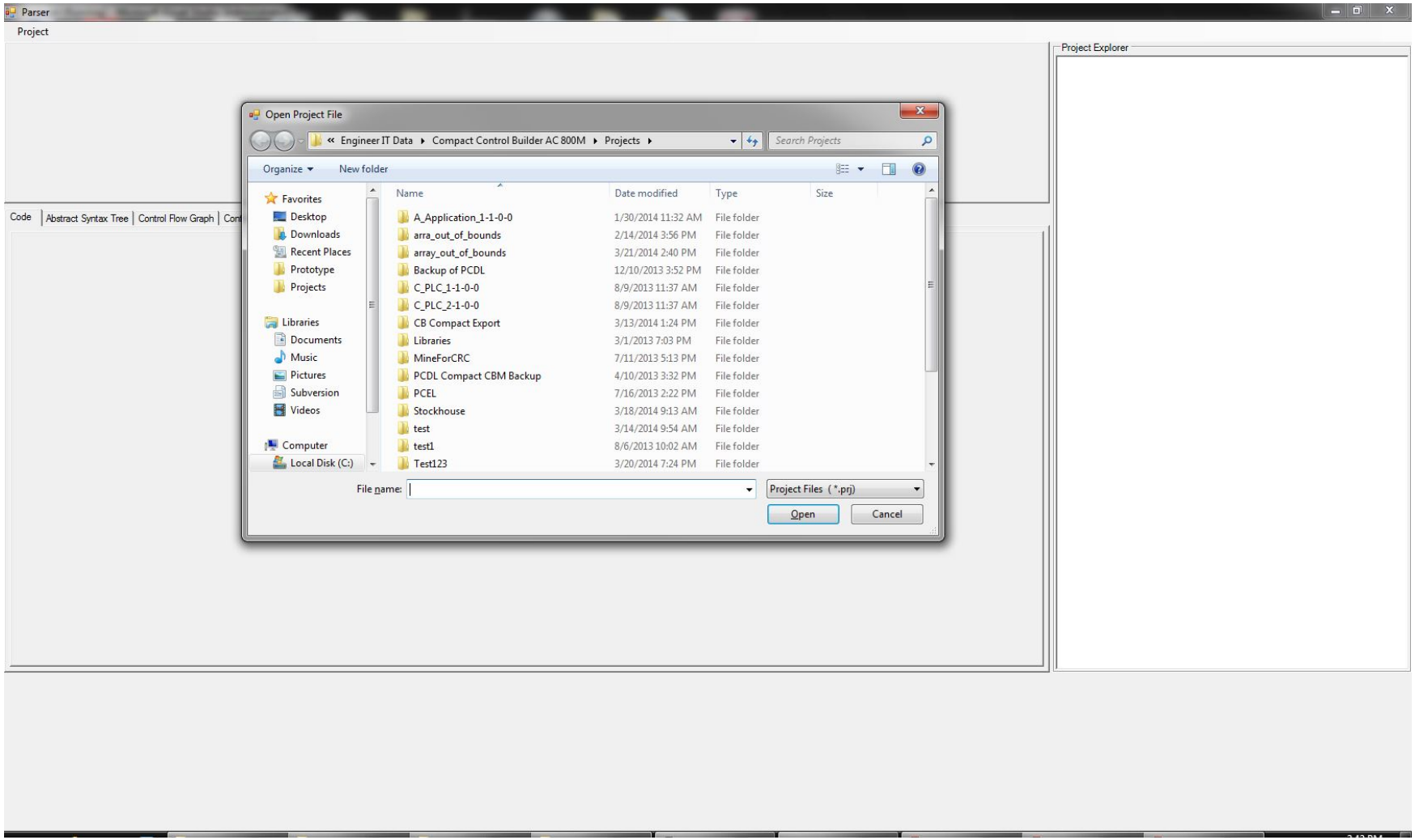
Mile stones

- Interview control system engineers to list critical run-time errors
- Finalize errors according to priority
- Implement grammar
- Generate Parse Tree
- Generate Abstract Syntax Tree
- Generate Control Flow Graph
- Design and implement analysis framework
- Design and implement algorithms for error rules
- Implement error handler

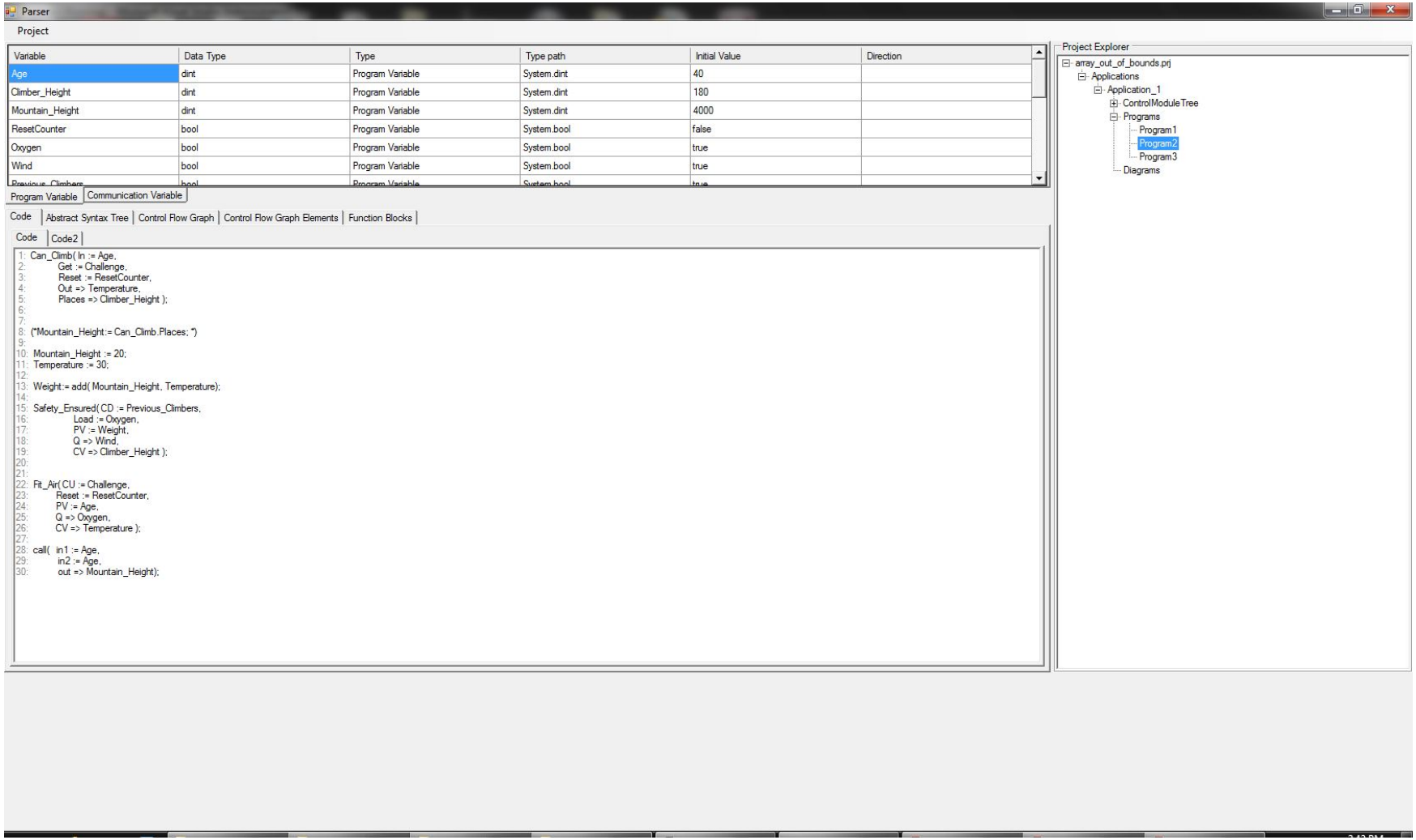
Control Builder Integration

- Control Builder
  - ABB's product for building Control System projects
  - Supports all five languages specified by IEC 61131-3 standard
- CBOpenInterface
  - Interface provided by ABB, for project details extraction

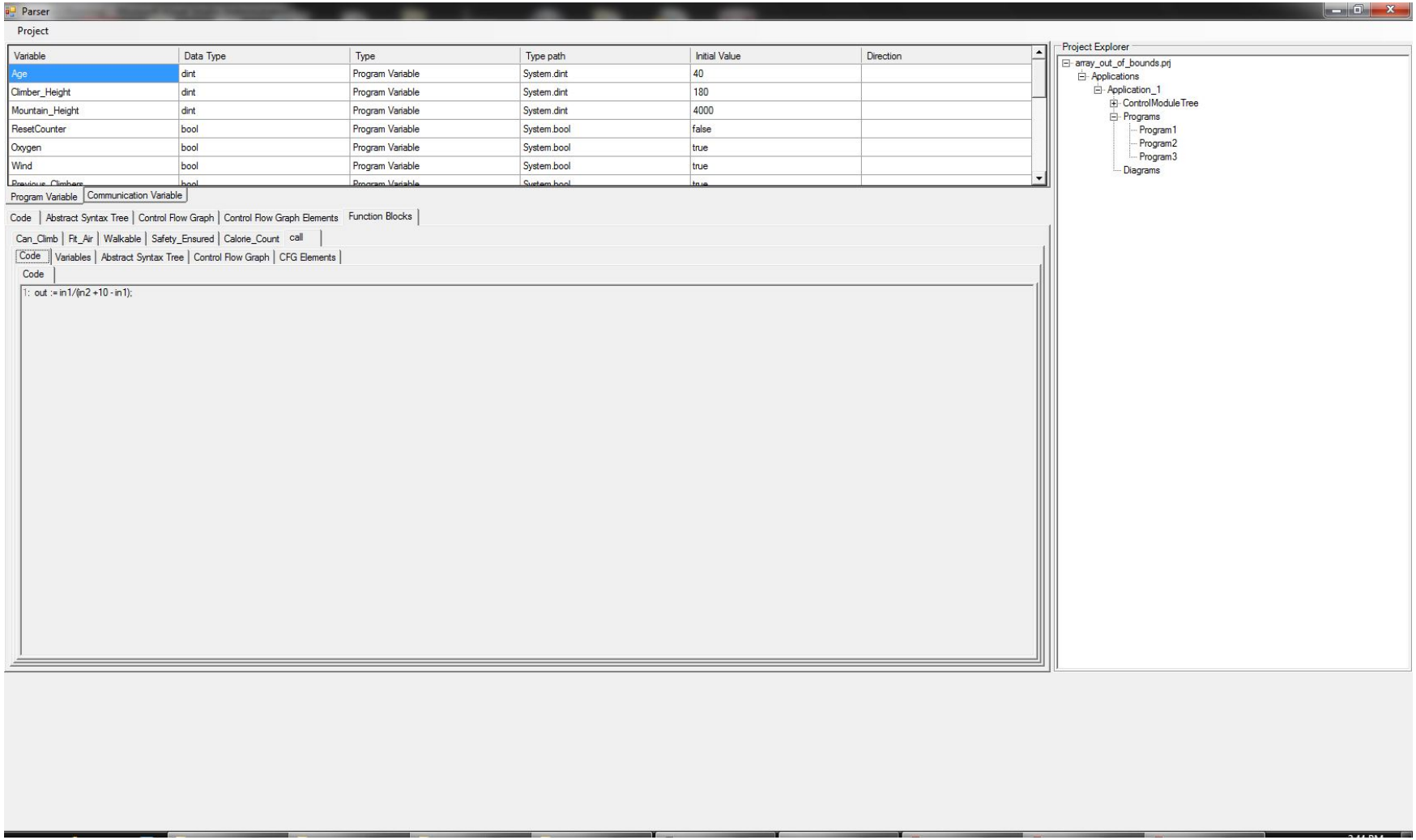
Screenshot


Parser
Project
Project Explorer
Code
Abstract Syntax Tree
Control Flow Graph
Open Project File
Engineer IT Data ▸ Compact Control Builder AC 800M ▸ Projects ▸
Search Projects
Organize
New folder
Favorites
Desktop
Downloads
Recent Places
Prototype
Projects
Libraries
Documents
Music
Pictures
Subversion
Videos
Computer
Local Disk (C:)
Name
Date modified
Type
Size
A_Application_1-1-0-0 1/30/2014 11:32 AM File folder
arra_out_of_bounds 2/14/2014 3:56 PM File folder
array_out_of_bounds 3/21/2014 2:40 PM File folder
Backup of PCDL 12/10/2013 3:52 PM File folder
C_PLC_1-1-0-0 8/9/2013 11:37 AM File folder
C_PLC_2-1-0-0 8/9/2013 11:37 AM File folder
CB Compact Export 3/13/2014 1:24 PM File folder
Libraries 3/1/2013 7:03 PM File folder
MineForCRC 7/11/2013 5:13 PM File folder
PCDL Compact CBM Backup 4/10/2013 3:32 PM File folder
PCEL 7/16/2013 2:22 PM File folder
Stockhouse 3/18/2014 9:13 AM File folder
test 3/14/2014 9:54 AM File folder
test1 8/6/2013 10:02 AM File folder
Test123 3/20/2014 7:24 PM File folder
File name:
Project Files (*.prj)
Open
Cancel

Screenshot (Contd..)



Screenshot (Contd..)



Grammar

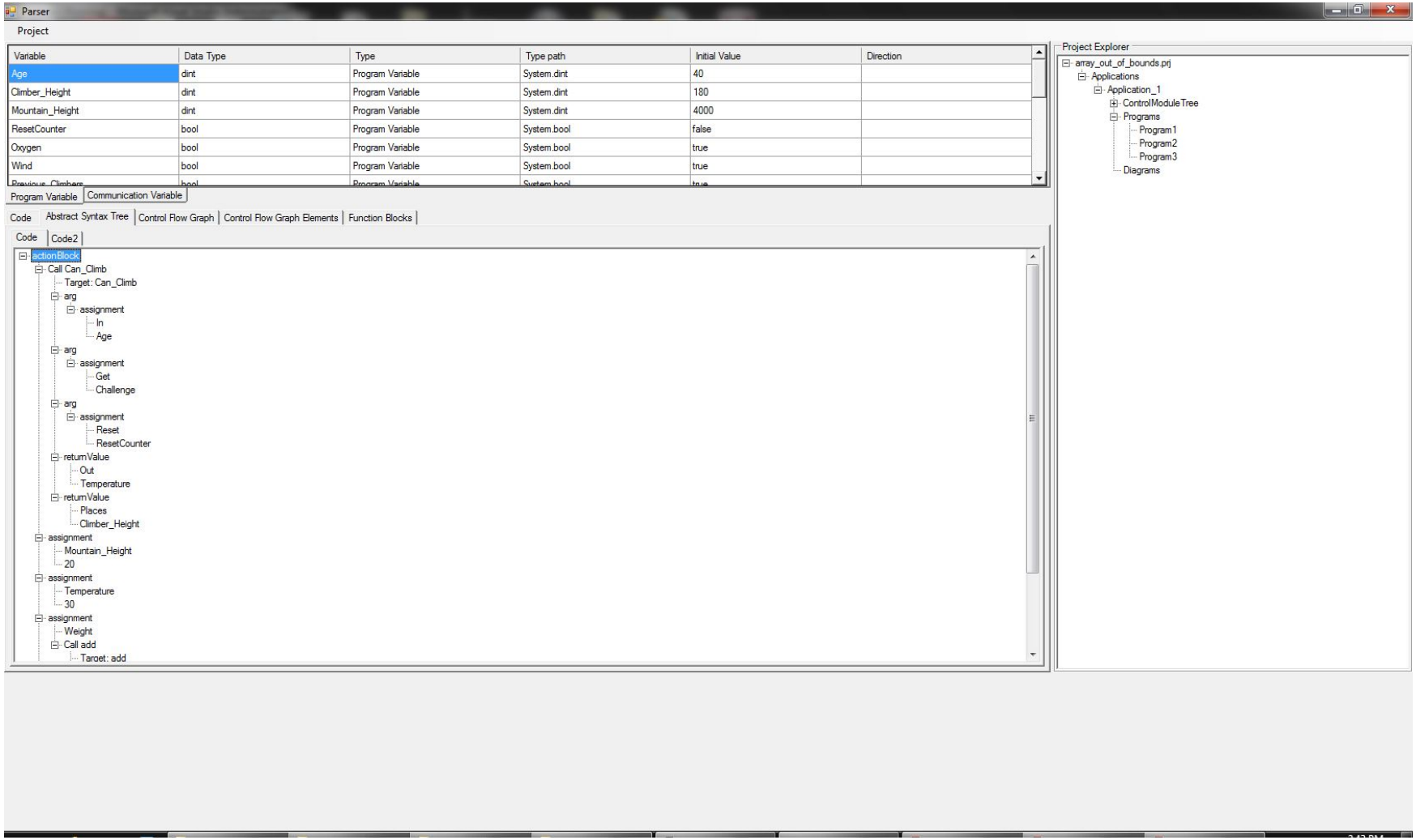
- Describes the hierarchical structure of a program
- Terminals - elementary symbols
- Non-terminals - variable symbols
- Productions - rules
- Start symbol - one non-terminal
- Example

```
expr.Rule = term | unExpr | binExpr | relExpr;
term.Rule = N | parExpr | funCall | identifier;
parExpr.Rule = LEFT_PAREN + expr + RIGHT_PAREN;
unExpr.Rule = unOp + term + ReduceHere();
unOp.Rule = ToTerm("-") | "+" | NOT;
binExpr.Rule = expr + binOp + expr;
binOp.Rule = ToTerm("+") | "-" | "*" | "/" | "**" | "mod";
relExpr.Rule = expr + relOp + expr;
relOp.Rule = AND | OR | XOR | "=" | "<" | "<=" | ">" | ">=" | "<>" |"&";
```

Parser

- Lexical Analyzer (Scanner)
- Tokens
- Syntactic Analyzer (Parser)
- Parse Tree
- Abstract Syntax Tree
- INPUT : ST program
- OUTPUT : AST

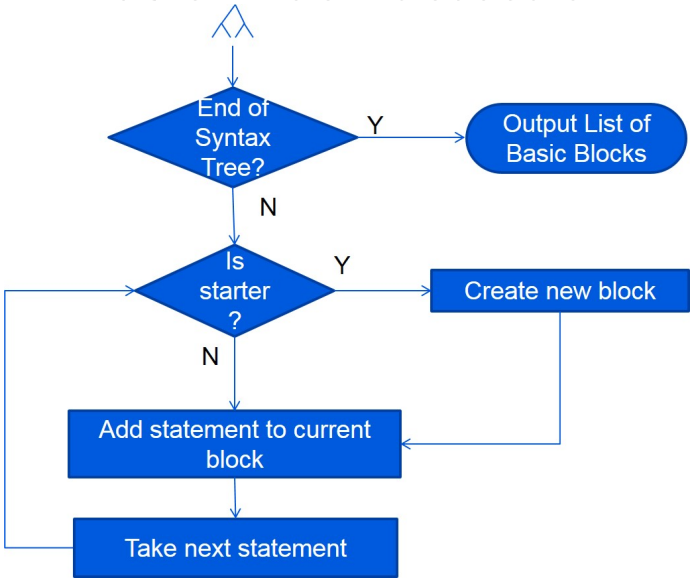
Screenshot


Parser
Project
Variable
Data Type
Type
Type path
Initial Value
Direction
Age
dint
Program Variable
System.dint
40
Climber_Height
dint
Program Variable
System.dint
180
Mountain_Height
dint
Program Variable
System.dint
4000
ResetCounter
bool
Program Variable
System.bool
false
Oxygen
bool
Program Variable
System.bool
true
Wind
bool
Program Variable
System.bool
true
Program Variable
Communication Variable
Code
Abstract Syntax Tree
Control Flow Graph
Control Flow Graph Elements
Function Blocks
Code
Code2
actionBlock
Call Can_Climb
Target: Can_Climb
arg
assignment
In
Age
arg
assignment
Get
Challenge
arg
assignment
Reset
ResetCounter
returnValue
Out
Temperature
returnValue
Places
Climber_Height
assignment
Mountain_Height
20
assignment
Temperature
30
assignment
Weight
Call add
Target: add
Project Explorer
array_out_of_bounds.prj
Applications
Application_1
ControlModuleTree
Programs
Program1
Program2
Program3
Diagrams

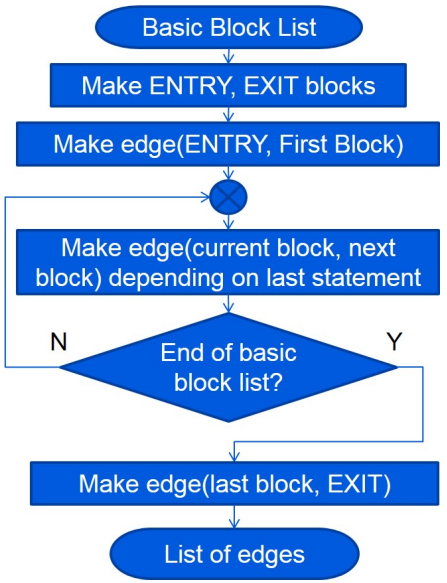
Control Flow Graph Generator

- Control Flow Graph
- INPUT : AST
- OUTPUT : CFG

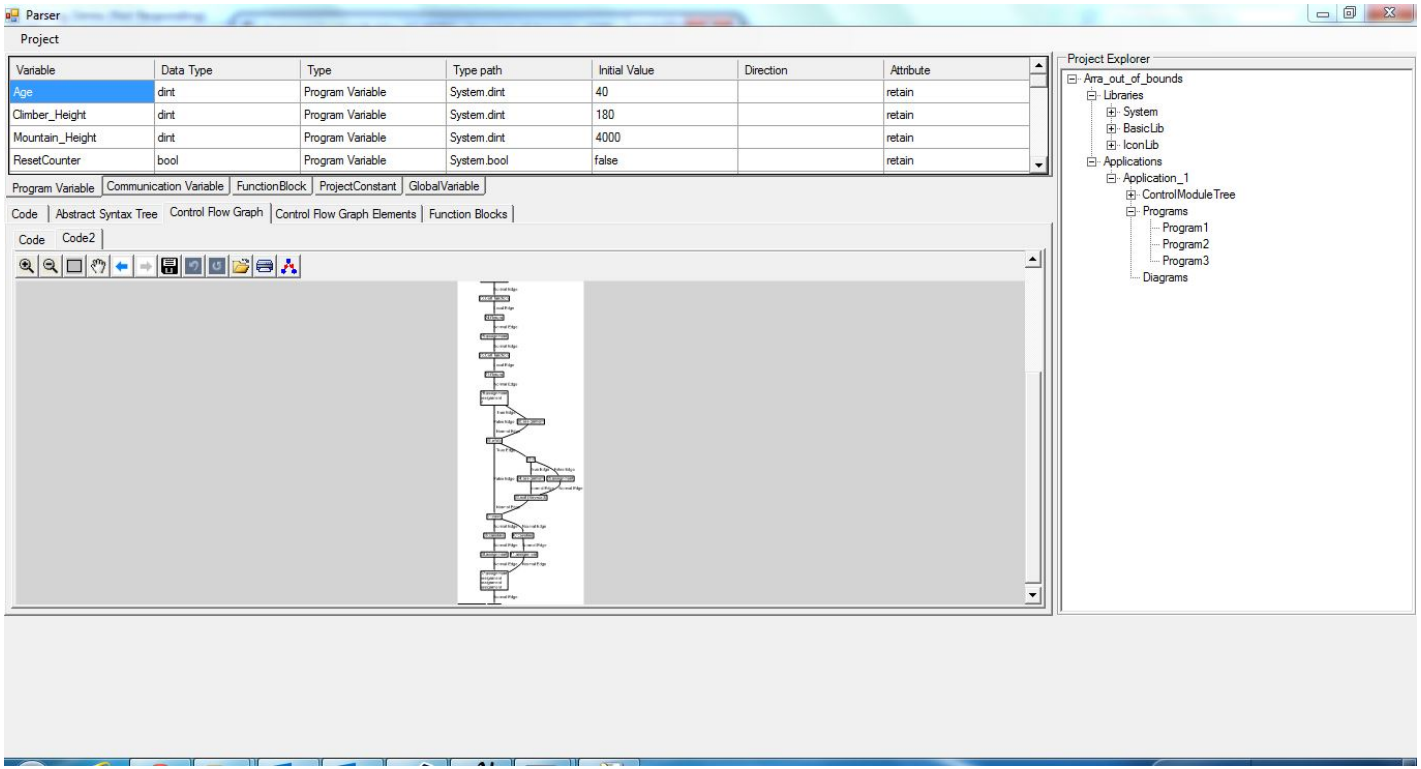
Basic Block detection



Connecting Basic Blocks



Screenshot


Parser
Project
Variable
Data Type
Type
Type path
Initial Value
Direction
Attribute
Age
dint
Program Variable
System.dint
40
retain
Climber_Height
dint
Program Variable
System.dint
180
retain
Mountain_Height
dint
Program Variable
System.dint
4000
retain
ResetCounter
bool
Program Variable
System.bool
false
retain
Program Variable
Communication Variable
FunctionBlock
ProjectConstant
GlobalVariable
Code
Abstract Syntax Tree
Control Flow Graph
Control Flow Graph Elements
Function Blocks
Code
Code2
Project Explorer
Arra_out_of_bounds
Libraries
System
BasicLib
IconLib
Applications
Application_1
ControlModuleTree
Programs
Program1
Program2
Program3
Diagrams

Data-Flow framework

- Data-Flow Analysis
- Data-Flow Framework - (D, V, $\bigcap$, F)
  - D - Direction
  - (V, $\bigcap$) - Semilattice
    - V - Domain
    - $\bigcap$ - Meet Operator
  - F - Transfer Function

Data-Flow Algorithms for Rules

Error Rules

- Division by zero
- Array access out of bounds
- Use of uninitialized variables
- Unused variables
- Invariant if condition
- Unreachable code
- Infinite loops

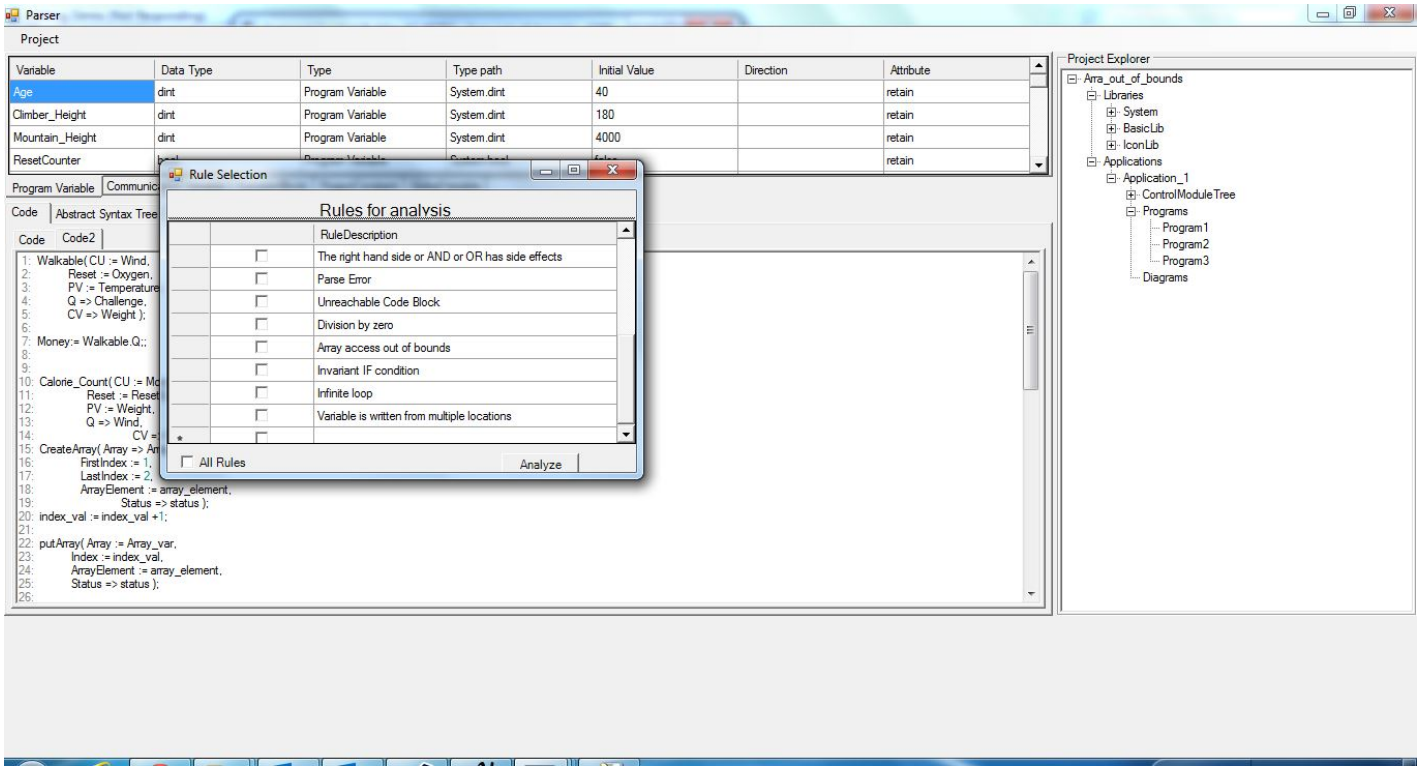
Analysis Algorithms

- Reaching Definitions
- Live Variable Analysis
- Interval Analysis

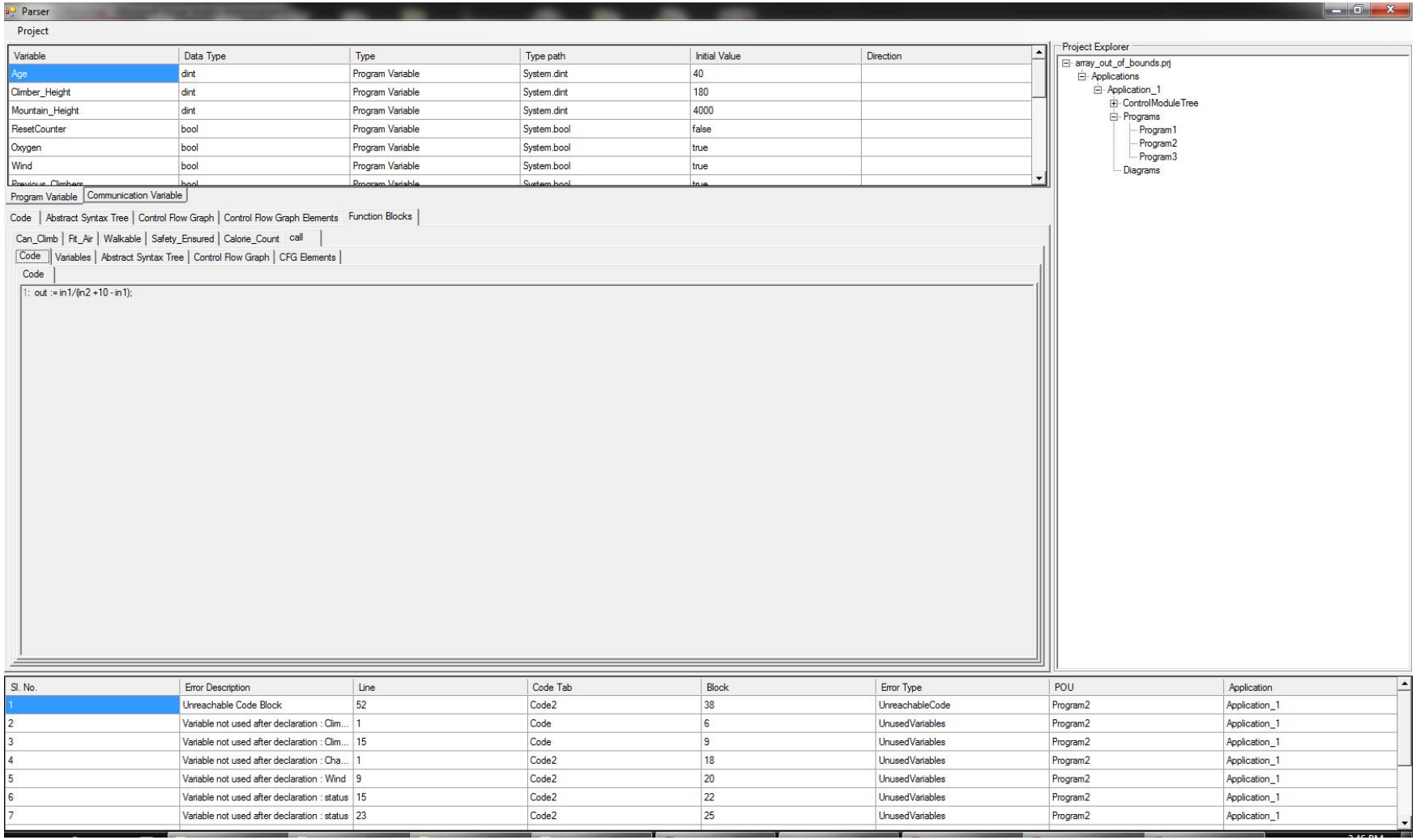
Error Handler

- Violation of rules
- Display message
  - Error name
  - Error description
  - Source of error

Screenshot


Parser
Project
Variable
Data Type
Type
Type path
Initial Value
Direction
Attribute
Age
dint
Program Variable
System.dint
40
retain
Climber_Height
dint
Program Variable
System.dint
180
retain
Mountain_Height
dint
Program Variable
System.dint
4000
retain
ResetCounter
retain
Rule Selection
Rules for analysis
RuleDescription
The right hand side or AND or OR has side effects
Parse Error
Unreachable Code Block
Division by zero
Array access out of bounds
Invariant IF condition
Infinite loop
Variable is written from multiple locations
All Rules
Analyze
Project Explorer
Arra_out_of_bounds
Libraries
System
BasicLib
IconLib
Applications
Application_1
ControlModuleTree
Programs
Program1
Program2
Program3
Diagrams

Screenshot (Contd..)



Conclusion

The results of testing on actual Control System Code

Project	LOC	Errors detected	Time taken(in seconds)
1	24	12	8.703
2	82	172	121.298
3	123	252	115.970

Acknowledgement

- Prof. Vineeth Paleri, Professor, NIT Calicut
- ABB Global Industries and Services Limited

Thank You