NuDE: Development Environment for Safety-Critical Software of Nuclear Power Plant

Jong-Hoon Lee, Junbeom Yoo
Dependable Software Laboratory
KONKUK University
Overview of NuDE

NuDE: Development Environment for Safety-Critical Software of Nuclear Power Plant
Development Process in NuDE

Requirements Analysis
- NuSRS
- NuSCRtoSMV (Embedded)

Design Synthesis
- NuSCRtoFBD
- FBDtoVerilog (VIS/SMV)

Implementation
- FBDtoC
- FBDtoVerilog (FPGA/CPLD)
NuDE

NuDE (Nuclear Development Environment)

- Integration of Existing Tools
  - NuSRS, NuSCRtoFBD, FBDtoVerilog, FBDtoC

IDE for Nuclear-Domain Software

- Requirement Analysis
  - Formal Requirement Specification (NuSCR)
  - Formal Requirement Verification via SMV
    - SMV Code Generation

- Design Synthesis
  - Automatic Translation from Requirement Specification (FBD)
  - Design Verification via VIS, SMV and HW-CBMC
    - Verilog Code Generation

- Implementation
  - C Code Generation
  - Verilog Code Generation for FPGA/CPLD
NuDE

Eclipse Plug-in 기반 통합
Requirements Analysis – NuSRS
Requirements Verification – NuSCRtoSMV
Design Synthesis – NuSCRtoFBD
Design Verification – FBDtoVerilog
Design Verification – FBDtoVerilog (Con’td)

- Formal Verification via SMV, VIS and HW-CBMC
  1) SMV Model Checking
  2) VIS Equivalence Checking between FBDs
  3) HW-CBMC E.C. between FBD and translated C program
Implementation – FBDtoC
Considerations for FPGA/CPLD

NPP Software based on PLC
• Implementation: FBD or C Code
  • 기존 PLC 기반 SW에서는 FBD나 C Code를 구현으로 사용

NPP Software based on FPGA/CPLD
• Implementation: Verilog HDL
  • FPGA/CPLD 기반 시스템에 대한 연구들이 진행 중
  • FPGA/CPLD는 Verilog HDL을 구현으로 사용
Considerations for FPGA/CPLD (Cont’d)

- NuSCR to FBDs
- FBDs to C
- C to Verilog
- Verilog to FPGA/CPLD
- Verification: HW-CBMC
- Verification: VIS, SMV

NuSCR

FBDs

C

Verilog

FPGA/CPLD

PLC
NuDE: Development Environment for Safety-Critical Software of Nuclear Power Plant

Improvements of NuDE
Not Yet Integrated

NuFTA

• FTA for Requirements Specification

VIS Analyzer

• Automated VIS Equivalence Checking

FBD Tester

• Generate Test Cases for FBDs Automatically
Not Yet Integrated (Cont’d)

Development Process

- NuSCR Formal Specification
- NuSRCtoSMV
- NuSRS

Verification & Validation

- SMV
  - Model Checking
  - NuFTA
    - FTA
- FBDtoVerilog
- FBDtoC
- C Program
- VIS Analyzer
- VIS Analyzer
  - Equivalence Checking

NuFTA (FTA for Requirements)

FBD Tester
(from Dr. Jee)

VIS Analyzer
Not Yet Developed

Development Process

Requirements

NuSCR
NuSCR Formal Specification
Automatic Translation
Nu_SCRtoFBD

Design

FBD Program
Automatic Translation
FBDtoC
C Program

Implementation

NuSRS
Automatic Translation
NuSCRtoSMV

Verification & Validation

Automatic Translation
NuFTA
FTA

Simulation for Requirements & Design

FBDFTA
(FTA for Design)

Our Own Testing Tool

Traceability Analyzer

SMV
Model Checking

VIS Analyzer
Equivalence Checking
NuDE: Development Environment for Safety-Critical Software of Nuclear Power Plant

Future NuDE
### Consideration for Future NuDE

#### FBD Programming
- A Guide for Safe FBD Programming
  - How to Design FBD Program Safe?

#### IDE for NPP Software based on FPGA/CPLD
- Seamless Transition from PLC to FPGA/CPLD
  - Automatic Translation from FBD to Verilog (FBDtoVerilog)
- Dependable Development
  - Dependability Demonstration for FBDtoC and FBDtoVerilog
- Verification for FPGA/CPLD
  - Verification Techniques (Simulation, Testing, etc.)
- A All-New Formal Requirements Specification Method
  - Formal Requirements Specification for Verilog HDL
Future NuDE

NuSCR → NuSCRtoFBD → FBDs → FBDtoC → C → FPGA/CPLD

NuSRS

FBDtoVerilog

Verification: HW-CBMC

Verilog → ? to Verilog

Verification: VIS, SMV

PLC
NuDE: Development Environment for Safety-Critical Software of Nuclear Power Plant

Conclusion
Conclusion

Our Goal

• SCADE를 능가할 수 있는 원자력 도메인 SW용 국산 IDE 개발
  • Dependable Development
    • Development life-cycle based on Formal Methods
    • Dependability Demonstration for Our Tools

Expectation

• 진화하는 원자력 SW 개발 환경을 선도
  • FPGA/CPLD기반의 SW 개발을 지원
  • PLC기반의 개발 산출물을 재사용
  • Natural Language Specification -> Formal Specification