NuSCR 정형 요구사항 명세로부터 FBD 프로그램 자동생성을 위한 CASE 도구

(A CASE Tool for Automatic Generation of FBD Program from NuSCR Formal Specification)

백 형 부 † 유 준 범 ††
(Hyoungbu Back) (Junbeom Yoo)
차 성 덕 †††
(Sungdeok Cha)

요 약 정형명세기법은 안전최우선시스템 소프트웨어의 안전성을 일정 수준 이상 보장할 수 있는 기법으로서, 원자력 발전소의 디지털 제어시스템의 개발에 사용되고 있다. 정형명세기법 NuSCR로부터 Programmable Logic Controller(PLC) 시스템을 구현하기 위한 소프트웨어인 Function Block Diagram(FBD) 프로그램을 자동으로 생성하는 기법[1]이 개발되었으나, 이를 지원하는 자동화 도구가 없어 이 기법이 널리 사용되지 못하였다. 본 논문에서는 이 자동생성 기법을 지원하기 위하여 개발된 자동화 도구 NuSCRtoFBD를 소개한다. 본 연구에서 제안하는 NuSCRtoFBD 도구를 사용하여 NuSCR로부터 FBD를 자동생성함으로써, 기존의 수동 프로그래밍 작업에서 발생했던 다수의 오류들을 줄일 수 있다.

키워드 : 정형명세, PLC, FBD, 디지털 제어시스템, 자동화 도구

Abstract Formal specification plays important roles in guaranteeing software safety of safety-critical systems such as nuclear power plant's digital control systems. We had developed a technique [1] which synthesizes Function Block Diagram(FBD) programs from NuSCR formal requirements specifications, but it did not be used widely as it had no automatic tool support. FBD is one of the programming languages for Programmable Logic Controllers(PLC) based system. This paper introduces a CASE tool, NuSCRtoFBD, developed to automate the synthesis procedure. The CASE tool NuSCRtoFBD can reduce a number of errors occurred in the process of manual FBD programming.

Key words : Formal Specification, Programmable Logic Controller, Function Block Diagram, Digital Controller, CASE

1. 서 론

NuSCR[1]은 안전성이 요구되는 원자력 발전소의 디지털 제어 시스템 소프트웨어의 요구사항을 명세하는데 적합하게 수정 보완된 정형명세 기법[2]으로 인정받고 있다. 최근의 원자력발전소의 디지털 제어시스템들은 하드웨어로서 PLC(Programmable Logic Controller)[3]를 사용하며, 설계 단계에서는 PLC를 구동하기 위해서 사용되는 프로그래밍 언어인 LD(Ladder Diagram)와 FBD(Function Block Diagram)[4] 등을 이용하여 프로그래밍을 수행한다.

NuSCR 명세를 기준으로 이와 동일한 행위를 하는 PLC 프로그램을 체계적으로 생성할 수 있다면, 기존의 수동으로 하던 명세 작업에서 발생하던 오류들을 크게 줄일 수 있으며, 소프트웨어의 개발 비용과 기간을 크게 줄일 수 있다. 정형명세 기법인 NuSCR로부터 PLC 기반의 FBD 프로그램을 자동으로 생성하는 기법[5]이 개발 되었지만 이를 지원하기 위한 도구가 없어 널리 사용되지 못했다. 본 논문에서는 이를 지원하기 위해서 개발된 자동 CASE 도구 - NuSCRtoFBD를 소개한다.

논문의 구성은 다음과 같다. 2장에서는 관련 지식으로서 NuSCR 정형명세언어와 FBD 프로그래밍에 대해 살펴본 후, 3장에서는 기 개발된 NuSCR 정형명세로부터 FBD 프로그램을 생성하는 절차에 대해 간략하게 설명한다. 4장에서는 본 논문에서 소개하는 NuSCRtoFBD 도구에 대해 설명한 후, 5장에서는 제안하는 도구와 유사한 관련 연구들을 소개한다. 마지막으로 6장에서 본 논문을 마무리한다.

· 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업과 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 학술진흥재단의 지원을 받아 연구되었음(KRF-2008-331-D00524)
· 이 논문은 제35회 추계학술대회에서 'NuSCR 정형 요구사항 명세로부터 FBD 프로그램 자동생성을 위한 도구 (NuSCRtoFBD) 개발'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 건국대학교 컴퓨터공학부
qogudqn@konkuk.ac.kr
†† 정 회 원 : 건국대학교 컴퓨터공학부 교수
jbyoo@konkuk.ac.kr
(Corresponding author임)
††† 종신회원 : 고려대학교 컴퓨터통신공학부 교수
scha@korea.ac.kr

논문접수 : 2008년 12월 11일
심사완료 : 2009년 2월 16일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제4호(2009.4)

2. Background

2.1 NuSCR

NuSCR[1]은 원자력발전소의 제어시스템 소프트웨어를 명세하도록 개발된 정형명세기법으로서 SCR[6]을 원자력발전소 제어시스템에 적합하게 수정 및 보완하였다. NuSCR은 Parnas의 Four-Variable Mode에 기반을 두고 추가적으로 function variable, history variable, timed-history variable의 세가지 모델을 사용한다. Function variable은 테이블 형태인 SDT(Structured Decision Table)에 의해 표현되며 수학적인 함수 관계를 나타내는데 사용된다. History variable은 오토마타 형태인 FSM(Finite State Machine)으로 표현되며 수학적인 함수 관계 보다는 상태의 흐름을 중심으로 명세할 때 보다 효율적으로 명세되는 내용일 때 사용된다. Timed-history variable은 FSM에 시간적 제약이 추가된 TTS (Timed Transition System)로 표현되며 시간적인 조건이 필요한 내용을 명세 할 때 사용된다.

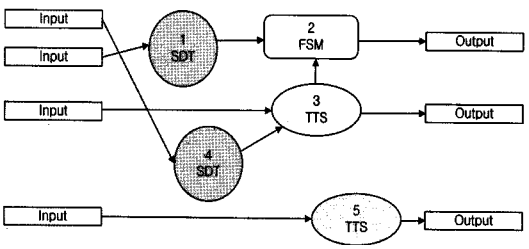

그림 1 NuSCR FOD의 예

이와 같이 NuSCR은 각각 다른 특징을 가지는 변수들을 혼용함으로써 대상을 보다 명확하게 명세할 수 있으며, 이를 위하여 데이터 흐름 다이어그램의 한 종류인 FOD (Function Overview Diagram)를 사용한다. 그림 1에서 보는 바와 같이 각 변수들은 실행 순서에 의한 상관관계를 지니는 노드로 표현되며, 각 노드들은 입력, 출력을 가진다. 보다 자세한 정의는 [1]에서 확인할 수 있다.

2.2 FBD

FBD(Function Block Diagram)[4]는 PLC를 구동하기 위한 소프트웨어 프로그램 작성에 널리 사용되는 프로그래밍 언어이다. FBD는 각각의 기능을 수행하는 FB (Function Block)들과 이들의 연결로 표현된다. 따라서 FBD는 정보의 흐름을 표현하는 프로그래밍 언어로 볼 수 있다. 기본적으로 산술연산, 논리연산, 비교연산, 선택연산, 시간연산 기능을 수행하는 FB들이 있다. 그림 2에서 보는 바와 같이 FB들은 FB의 상단에 위치한 고유번호에 의해 실행 순서를 가지고 선들에 의해 연결되어 절차적인 흐름을 표현한다.

3. NuSCRtoFBD 생성 과정

정형명세 NuSCR로부터 FBD 프로그램을 자동으로 생성하는 과정[5]은 다음과 같다.

(STEP 1) 완전성 및 일관성 분석 NuSCR 명세 상의 모든 변수들은 문법상의 오류 없는 FBD를 생성하기 위해 먼저 완전성과 일관성[1] 분석을 수행해야 한다. SDT의 경우 조건들이 모든 가능한 경우를 포함하고 있는지 판별해야 하며, 중복된 조건에 대해서도 유사한 확인 작업이 필요하다. 또한 한 조건에 대해 서로 다른 출력 값을 가지는지도 분석해야 한다. FSM과 TTS는 오토마타의 일종이므로 전이의 조건이 만족하지 않으면 현재 상태를 유지하게 된다. 따라서 이러한 경우 적절한 반응액션을 추가 해주어야 하며, 한 상태에서 두 가지 이상의 전이 조건을 만족하는 경우가 있는지도 살펴보

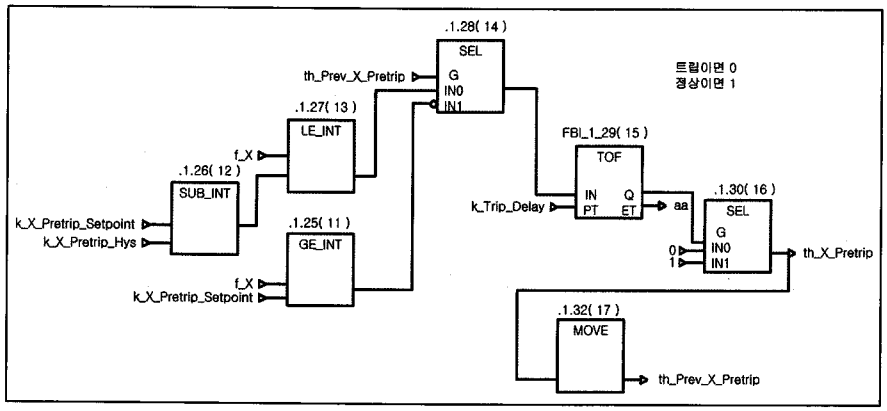

그림 2 FBD의 예(원자력발전소 원자로보호시스템에서 발췌함)

아야 한다. FBD 생성 알고리즘에 대한 수학적인 증명과, 본 연구에서 개발한 도구 NuSCRtoFBD를 실제 프로젝트에 적용한 실례는 [7]에 자세히 소개되어 있다.

(STEP 2) FSM과 TTS에 대한 2C-Table생성 FSM과 TTS는 완전성과 일관성을 확인한 후에 2C-Table형태로 변환 된다. 2C-Table은 그림 3과 같이 각 상태에 존재하는 전이조건들과 그 조건이 만족했을 때 가지는 출력 값과 다음 상태의 정보로 구성된다. 완전성과 일관성이 확인된 2C-Table은 상호 배타적인 특성을 지니므로 적은 수의 FB들만으로 정의된 FBD를 구성할 수 있다. SDT는 이미 이러한 테이블 형태이므로 동일한 작업을 수행할 수 있다.

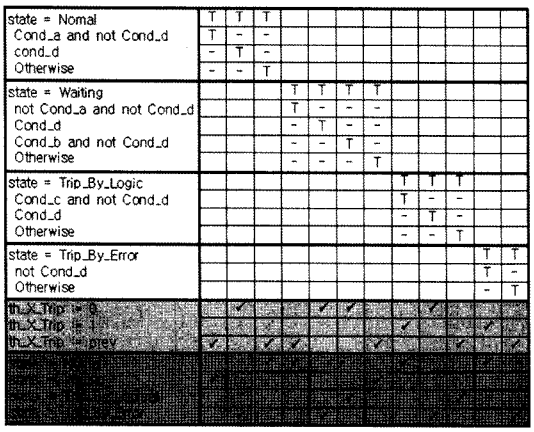

그림 3 상호배타성인 특성을 지니는 2C-Table

(STEP 3) 기본 FBD생성 다음 단계에서는 수정된 SDT나 추가적으로 생성된 2C-Table을 기반을 FBD를 생성한다. SDT는 조건들만을 분류해서 미리 계산해 놓은 "전처리 FBD"와 구체적인 계산을 수행하는 "출력 계산 FBD"로 구분되어서 작성되며, FSM과 TTS는 전처리 FBD와 출력 계산 FBD외에도 오토마타의 상태를 계산하기 위한 "상태변수 계산 FBD"가 추가 작성된다. 이와 같은 과정이 완료되면, FOD 상에 존재하는 모든 function variable node 및 history variable node와 timed history variable node에 대한 개별적인 FBD가 완성된다.

(STEP 4) FBD의 실행순서 결정 마지막 단계에서는 각각의 FBD에 실행 순서를 지정해 주는 작업이 진행된다. 그림 1과 같이 5개의 노드로 구성된 FOD를 대상으로, 가능한 모든 실행 순서를 고려해 보면, 먼저 다음과 같은 세 종류의 부분적인 순서 관계가 있음을 알 수 있다.

Partial execution order 1: 1 → 2
Partial execution order 2: 3 → 2
Partial execution order 3: 4 → 3 → 2

5번 노드는 다른 노드들과 연관 관계가 없어 독립적으로 수행할 수 있다. 이러한 부분적인 순서 관계와 독립적인 노드를 이용하여 전체적인 순서를 정하면 다음과 같다.

Execution order : 입력 → (1,4) → 3 → 2 → (5)→ 출력

각 부분적인 순서 관계에서 3번 노드과 2번 노드가 중복 되므로 그 노드를 기준으로 나머지 노드들의 순서를 결정하고 5번 노드는 독립적이므로 어느 위치에 오더라도 무관하다.

4. NuSCRtoFBD CASE Tool

NuSCRtoFBD는 앞 장에서 소개한 생성과정을 이용하여 NuSCR 정형 요구사항명세로부터 FBD를 자동으로 생성하는 CASE 도구이다. 본 장에서는 NuSCRtoFBD의 기능에 대해 자세히 설명한다. NuSCRtoFBD는 FBD를 생성하여 시각화하는 기능 외에도, 일정 수준의 완전성과 일관성 분석 기능과 FBD 생성 과정에서 얻어지는 정보들을 한눈에 볼 수 있는 기능을 제공한다. 전체적인 화면 구성은 그림 4와 같다. 상단에 위치한 메뉴들을 통해 NuSCR 읽기, 완전성과 일관성 여부 확인 등의 기능들을 제공한다. 좌측에 위치한 'Hierarchy Window'를 통해 노드들의 상하 관계를 tree 형태로 파악 할 수 있고, 'Description Window'를 통해 조건들의 논리오류 여부를 파악할 수 있다. 또한 우측에 위치한 각각의 탭을 통해 좌측의 'Hierarchy Window' 에서 선택한 노드에 대한 2C-Table과 FBD의 정보들을 파악할 수 있다. 현재 NuSCRtoFBD가 제공하는 주요 기능들을 자세히 정리하면 다음과 같다.

4.1 2C-Table 및 Combined 2C-Table 생성

NuSCR 정형명세의 FSM과 TTS는 완전성과 일관성 분석을 거친 후 2C-Table 형태로 표현된다. SDT의 경우 테이블 형태로 표현되므로 내용 그대로 우측의 2C-Table Tab에 생성된다. NuSCRtoFBD는 기본적인 2C-Table 외에 Combined 2C-Table을 생성한다. Combined 2C-Table은 전처리 FBD를 위해 생성된 것으로서, 각 조건들 중에서 공통적으로 사용할 수 있는 작은 단위의 조건들을 추출하여 새로운 이름을 할당한 후 작성된 테이블이다. 그림 4에서 보는 바와 같이 TTS로부터 생성된 2C-Table은 Combined 2C-Table로 변환되어 새롭게 생성되며 'Combined Text' 메뉴를 통해 새롭게 할당된 이름의 원래 조건들을 확인할 수 있다.

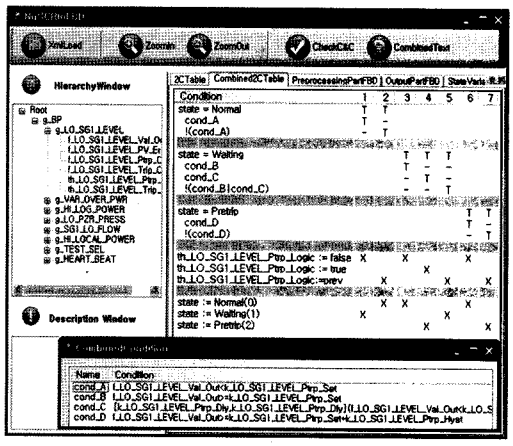

그림 4 TTS에 대한 Combined 2C-Table

또한 NuSCRtoFBD는 일정 수준의 완전성과 일관성 분석 기능을 제공하는데 'Check C&C' 메뉴를 통해 그 결과를 확인 할 수 있다. 그림 5에서 보이는 SDT의 경우 두 조건에 대한 모든 경우의 수에 대해 행위가 정의되어 있지만 1열과 3열에서 중복된 조건들의 조합이 정의되어 있다(오류가 발생하지는 않음). 만약 1열과 3열이 서로 상이한 행위를 가진다면 오류 메시지를 출력하게 된다.

4.2 Base FBD 생성

NuSCRtoFBD는 SDT, FSM, TTS로부터 생성된 2C-Table과 Combined 2C-Table을 통해 전처리 FBD, 출력 계산 FBD, 상태변수 계산 FBD를 생성한다. 우선 FB들을 선으로 연결하고 FB의 상단에 숫자를 할당함으로써 FB들의 흐름을 한눈에 파악 할 수 있게 도와준다. 또한 SDT에 한하여 FBD 생성에 대한 최적화 기능을 제공한다.

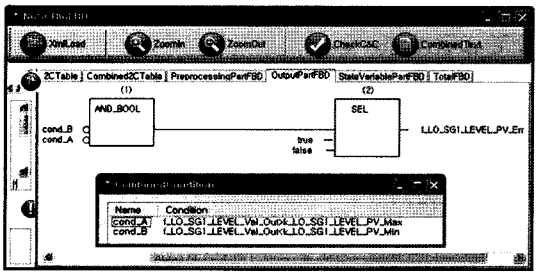

그림 5 SDT에 대한 출력 계산 FBD

4.3 FBD의 실행순서 결정 및 Total FBD 생성

NuSCRtoFBD는 FOD에 포함되는 모든 노드들의 순서를 결정하기 위해서 변형된 그래프 넓이 우선 탐색을 이용한다. 기존의 넓이 우선 탐색의 레벨(Level) 부여 방법 대신 모든 입력 노드를 하나의 출발점으로 묶은 후 한 노드의 이전 노드 중 레벨이 가장 높은 노드에 1을 더하는 방식을 사용한다. 그림 1의 FOD에 적용시킨 후 기준을 분석하면 다음과 같다.

Level 1 : (1, 4, 5) , Level 2 : (3) , Level 3 : (2)

단 노드3과 노드2의 경우에 각각 레벨 1과 레벨 2로부터 입력을 받으므로 각각 레벨 2와 레벨 3을 부여한다. 기준의 순서인 (1, 4, 5) → (3) → (2) 순서로 실행하게 되면 존재하는 모든 부분적인 실행순서(Partial

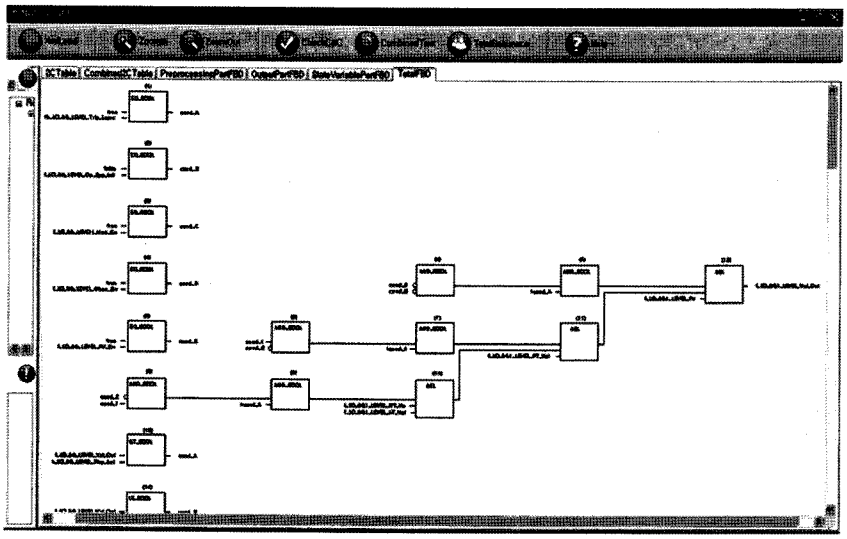

그림 6 FOD에 포함된 모든 노드에 대한 FBD

Execution Order)를 만족하게 된다. NuSCRtoFBD는 이와 같은 방법으로 각 노드에 대한 레벨을 부여하고 레벨의 크기에 따라 전체 순서를 결정한다. 그림 6은 NuSCRtoFBD에서 FOD의 노드들에 대해 레벨을 부여하고 전체 순서를 결정한 후, FOD 내에 포함된 모든 노드들의 FB들은 한꺼번에 생성한 예이다.

4.4 NuSCRtoFBD 도구의 정확성 확보 방안

NuSCRtoFBD 지원 도구는 MS .Net 기반으로 개발되었으며, NuSCR 정형 요구사항명세 지원 도구인 NuSRS 2.0에서 저장된 XML 기반의 NuSCR 정형명세를 입력으로 사용한다. 현재 버전은 입력 받은 NuSCR 정형명세를 동일한 행위를 지니는 FBD 프로그램으로 변환하여 화면에 출력하는 기능만을 제공하며, 이를 추후에 다시 사용하기 위하여 저장하는 기능을 제공하지는 않는다. 현재 개발 중인 NuSCRtoFBD 지원 도구 Ver.2.0은 FBD 프로그램의 저장 포맷(XML 기반)을 정의한 국제 표준에 맞추어, 생성된 FBD 프로그램을 XML 파일로 저장하는 기능을 추가로 지원할 계획이다. 이 기능이 지원되면, [8]에서 제시된 원자력발전소 디지털제어시스템 소프트웨어를 정형기법 기반으로 개발하기 위한 방법론을 바탕으로 모델체킹과 일치성검증 기법을 이용하여 구현된 도구의 정확성을 보다 정확하고 자세하게 검증할 계획이다.

5. 관련 연구

PLC를 구동하기 위한 소프트웨어 프로그램을 자동으로 또는 체계적으로 생성하기 위한 기존의 연구들은 소개하면 다음과 같다. ProCoS-method[9] Duration Calculus[10]로 작성된 요구사항명세를 입력으로 여러 단계의 변환 과정을 거쳐 하드웨어 개발에 사용되는 OCCAM과 유사한 프로그램을 생성한다. 후속 연구로 수행된 UniForM-project[11]에서는 전 프로젝트의 단점으로 지적되던 자동화 도구 지원 부분을 보완하였다. 또한, 복잡한 변환 단계들의 자동화를 지원하는 UniForM-Workbench를 개발하였으며, 이 도구는 OCCAM 기반의 프로그램이 아닌 PLC에서 사용되는 ST(Structural Text)을 최종적으로 변환한다.

6. 결론 및 향후 연구계획

원자력발전소의 NuSCR 정형명세로부터 PLC를 위한 FBD 프로그램을 자동으로 생성하는 기법이 개발 되었지만 이를 지원하기 위한 도구가 없어 널리 사용되지 못하였다. 본 논문에서는 이를 지원하는 도구인 NuSCR-toFBD를 소개하였다. 소개한 도구를 이용하여 NuSCR로부터 FBD의 자동생성 과정에서 얻어지는 정보들과 FBD를 한눈에 볼 수 있고, 명세 작업 중 발생한 오류들도 파악 할 수 있다. 또한 이 도구에 의해 FBD 자동생성기법이 좀 더 널리 사용될 것으로 예상된다.

향후 계획으로는 자동 생성된 FBD를 표준 XML 포맷으로 저장함으로써, 추후 FBD programming tools에서 읽어 사용할 수 있는 interface를 추가로 개발할 계획이다. 또한, 기발행된 원자력발전소의 NuSCR 정형명세와 FBD 명세를 사용하여, 제안된 도구로부터 자동생성된 FBD와 공식적으로 발행된 FBD가 동일한 행위를 가지는 가를 지속적으로 실험함으로써, 제안된 도구 NuSCRtoFBD의 정확성과 건전성을 확보하기 위한 노력도 꾸준히 진행될 예정이다.

참 고 문 헌

[1] Jumbeom Yoo, Taihyo Kim, Sumgdeok Cha, Jangsu Lee, and Han Seong Son, "A Formal Software Requirements Specification Method for Digital Nuclear Plants Protection Systems," Journal of Systems and Software, Vol.74, No.1, pp. 73-83, 2005.

[2] Doron A. Peled, SOFTWARE RELIABILITY METHODS, Springer-Verlag, 2001.

[3] Henning Dierks, "PLC-Automata: A new class of implementable real-time automata," Theoretical Computer Science, Vol.253, No.1, pp. 61-93, 2001.

[4] IEC, International standard for programmable controllers: Programming languages 61131- Part 3, 1993.

[5] Junbeom Yoo, Sungdeok Cha, Chang Hwoi Kim, and Duck Yong Song, "Synthesis of FBD-based PLC design from NuSCR formal specification," Reliability Engineering and System Safety, Vol.87, No.2, pp. 287-294, 2005.

[6] K.L. Heninger, "Specifying software requirements for complex systems: New techniques and their application," IEEE Trans. Software Engineering, SE-6(1):2-13, 1980.

[7] Junbeom Yoo, Sungdeok Cha, and Eunkyoung Jee, "Automatic Synthesis of Function Block Diagrams from NuSCR Requirements Specification," submitted to Information and Software Technology, 2008.

[8] Junbeom Yoo, Eunkyoung Jee, and Sungdeok Cha, "Formal Modeling and Verification of Safety-Critical Software implemented in PLC," IEEE Software, to be published, 2009.

[9] ProCos: Hardware Compilation, http://archive.comlab.ox.ac.uk/hwcomp/procos.html.

[10] Zhou Chaochen, C. Hoare, and A. Ravn, "A Calculus of Duration," Information Processing Letter, Vol.40, No.5, pp. 269-276, 1991.

[11] Uniform - Universal Formal Methods Workbench, http://www.informatik.uni-bremen.de/uniform/.