디지털 원자로 보호 시스템을 위한 정형 소프트웨어 요구사항 명세

(Formal Software Requirements Specification for Digital Reactor Protection Systems)

유 준 범 † 차 성 덕 †† 김 창 회 ††† 오 윤 주 ††††
(Junbeom Yoo) (Sungdeok Cha) (Chang Hwoi Kim) (Younju Oh)

요 약 원자력 발전소의 디지털 제어 시스템에 적용되는 소프트웨어는 안전성이 중요시되는 safety-critical 소프트웨어로, 충분한 수준의 안전성을 보장하기 위해서 여러 기법들이 적용되고 있다. 특히, 정형명세 기법은 개발의 초기 단계에서 소프트웨어 요구 사항들을 명확하고 완전하게 명세하도록 유도함으로써 안전성을 크게 향상시킬 수 있는 기법으로 인정 받고 있다. 본 논문에서는 원자력 발전소 디지털 제어 시스템 소프트웨어의 요구 사항 명세에 적합하도록 개발된 정형명세 기법인 NuSCR을 논의한다. 또한, 개발된 NuSCR의 적용성을 검토하기 위해, 현재 KNICS 사업단에서 개발중인 발전소보호계통 소프트웨어의 요구사항을 정형 명세한 경험을 소개하고 있다. 또한, 원자력 도매인에 특화된 정형명세 기법인 NuSCR의 우수성도 실례를 들어 설명하고 있다.

키워드 : 정형기법, 소프트웨어 요구사항 명세, 원자력발전소 제어 소프트웨어

Abstract The software of the nuclear power plant digital control system is a safety-critical system where many techniques must be applied to it in order to preserve safety in the whole system. Formal specifications especially allow the system to be clearly and completely specified in the early requirements specification phase therefore making it a trusted method for increasing safety. In this paper, we discuss the NuSCR, which is a qualified formal specification method for specifying nuclear power plant digital control system software requirements. To investigate the application of NuSCR, we introduce the experience of using NuSCR in formally specifying the plant protection system's software requirements, which is presently being developed at KNICS. Case study that shows that the formal specification approach NuSCR is very much qualified and specialized for the nuclear domain is also shown.

Key words : formal method, software requirements specification, nuclear power plants protection system

1. 서 론

항공·우주 관련 제어 시스템이나 인공위성 시스템, 화학처리공장 제어 시스템, 원자력 발전소 제어 시스템들은 고장이 발생할 경우, 인명이나 재산에 커다란 손실을 야기할 수 있는 안전성이 중시되는 시스템(safety critical system)이다. 이러한 시스템들에 있어 소프트웨어 안전성(software safety)은 가장 중요시되는 요구 사항 중의 하나이다. 특히 안전성이 중시되는 소프트웨어 시스템들의 규모와 복잡성이 점점 증가함에 따라 소프트웨어 안전성은 더욱 더 중요시 되고 있는 추세이다[1].

정형기법(formal method)은 이러한 복잡한 소프트웨어 시스템이 지니는 제반 문제들을 해결하기 위해서 제안된 기법으로서 크게 정형명세(formal specification)와 정형검증(formal verification)으로 구분할 수 있다[2]. 소프트웨어 개발의 초기 단계에 적용되는 정형명세 기법은, 개발자가 모든 요구 사항들을 생략하지 않고 명확

· 본 연구는 첨단과학기술연구센터(AITrc), 소프트웨어프로세스개선센터(SPIC) 및 인터넷 침입대응기술연구센터(IIRTRC)의 지원을 받음
† 비 회 원 : KAIST 전자전산학과
jbyoo@salmosa.kaist.ac.kr
†† 종신회원 : KAIST 전자전산학과 교수
cha@salmosa.kaist.ac.kr
††† 비 회 원 : 한국원자력연구소 계측제어인간공학연구부 책임연구원
chkim2@kaeri.re.kr
†††† 비 회 원 : LG전자 UMTS 연구소 차세대단말팀 연구원
coolcool@lge.com
논문접수 : 2003년 6월 17일
심사완료 : 2004년 3월 26일

하게 명세하도록 유도함으로써 소프트웨어의 안전성을 크게 향상시킬 수 있는 기법으로 인정 받고 있다. 특히, 개발하려는 시스템의 특성에 알맞게 특화된 정형명세 기법들이 다수 제시됨으로써 개발 초기 단계에서 보다 명확하게 모든 요구 사항들을 명세 하려는 노력이 꾸준히 진행되고 있다[3]. 정형검증 기법은 이러한 정형명세를 기반으로 하여 모델체킹(model checking)[4]이나 정리증명(theorem proving)[5] 등의 검증을 통해서 정형명세된 소프트웨어의 안전성을 증명하는 기법이다. 이미 신뢰성이 인정된 다수의 정형검증 기법들이 소개되었으며, 이 기법들은 정형명세를 바탕으로 적합한 기법이 적용될 수 있다.

최근의 10여 년 동안 원자력 발전소에 디지털 제어시스템이 적용되면서 소프트웨어 안전성을 특히 중요하게 요구하고 있다. 이는 기존에 제작되었던 RLL(Relay Ladder Logic) 기반의 아날로그 하드웨어 제어 시스템들을 소프트웨어 기반의 디지털 제어 시스템으로 교체하고 있기 때문이다[6]. 따라서, 정형명세 기법을 통해서 제어 소프트웨어의 안전성을 초기 단계에서 향상시키기 위한 노력이 요구되고 있으며, 원자력 발전소의 제어 시스템 소프트웨어를 명세 하는데 적합하게 수정·보완된 정형명세 기법을 개발하기 위한 노력도 꾸준히 진행중이다[7,8].

본 논문은 원자력 발전소에서 사용되는 디지털 제어 시스템 소프트웨어를 명세 하는데 적합하도록 개발된 정형명세 기법인 NuSCR[9]을 소개하고, 발전소보호계통 응용 프로그램을 명세하면서 얻은 경험들을 소개하고 있다. 또한, 특화된 정형 기법인 NuSCR의 우수성을 실례를 통해 설명하고 있다. 논문의 추후 구성은 다음과 같다. 2장에서는 기존의 정형명세 기법들의 특징을 설명한다. 3장에서는 기존에 제시된 기법들의 단점을 극복할 수 있는 기법으로서 NuSCR을 소개하며, 4장에서는 현재 KNICS[7]에서 개발하고 있는 디지털 발전소보호계통(Digital Plant Protections System)의 원자로보호계통(Reactor Protection System) 중에서 가장 중요한 논리를 수행하는 부분인 BP(Bistable Processor) 및 CP(Coincidence Processor)를 NuSCR을 이용해서 명세한 결과를 소개하고 있다. 5장에서는 정형명세 기법인 NuSCR의 적용을 통해서 안전성이 향상된 사례를 구체적인 예를 이용해 설명한 후에, 6장에서 결론 및 향후 연구 방향에 대해 언급하고 있다.

2. 관련 연구

Z[10], VDM[11], Larch[12]는 원자력 발전소의 제어 시스템과 유사한 순차적인 시스템의 행위를 명세할 때 사용할 수 있는 정형명세 기법들이다. 이 기법들은 set, relation, function과 같은 수학 구조체들을 사용하여 상태(state)를 기술하며, pre-condition과 post-condition을 이용해서 상태간의 전이(transition)를 기술한다. 이러한 기법들은 충분한 표현력이라는 장점을 지니는 반면에, 원자력 발전소의 제어 소프트웨어를 기술하기에는 너무 복잡하다는 단점이 있다.

SCR[13]은 실시간 내장 시스템(real-time embedded system)의 소프트웨어 요구사항을 명세하기 위해서 20여년 전에 개발된 기법으로서 현재까지도 지속적으로 발전·보완되고 있는 기법이다. 최근에는 SCR을 이용해서 timing이나 accuracy와 같은 non-functional requirements도 명세 하려는 노력이 시도되고 있다[14,15]. SCR은 본래 현장의 엔지니어에 의해서 개발되었다. 따라서, 현장의 엔지니어들에게 가장 친숙한 정형명세로 알려져 있으며, A-7 Operational Flight Program[16]이나 잠수함 통신 시스템, Canada Darlington 원자력 발전소의 안전 컴포넌트[17] 등의 실제 시스템에도 꾸준히 적용되어 왔다.

AECL에 의해서 Canada Darlington 발전소에 적용된 [14]의 기법은 원자력 발전소 디지털 시스템에 적용된 최초의 정형명세 기법이며, 또한 한국 월성의 SDS2에도 적용된 정형명세 기법이다[17]. 이 기법은 SCR에 기반하고 있으며 추가적인 확장을 지닌다. 먼저 보다 명확하게 명세하기 위해서 SCR이 지니는 세 종류의 테이블을 한 종류의 테이블 SDT(Structured Decision Table)로 통합했다. 또한, DFD의 일종인 FOD (Function Overview Diagram)를 사용하여 전반적인 자료의 흐름을 볼 수 있도록 했으며, 시간제약(timing constraints) 조건을 명세하기 위해서 섬세한 시간 함수들(timing functions)을 제공하고 있다.

AECL이 제시한 기법의 특징을 정리하면 다음과 같다.

① SDT와 FOD가 도매인 엔지니어들에게 친숙한 표현 방법이다. SDT는 현장의 개발 엔지니어들이 일반적으로 사용하는 테이블 형태이며, FOD는 설비 흐름도(Instrumentation Diagram)와 같은 유형의 flow diagram이므로 쉽게 접하여 사용할 수 있다.

② SDT가 테이블 형태로서 쉽게 사용할 수 있는 반면에, 테이블의 행과 열의 개수가 10*10 이상으로 늘어남으로 인해서 SDT를 해석하는데 어려움을 겪는 경우가 많다. 이는 AECL 방법론이 시스템의 모든 행위를 테이블 형태의 function으로만 기술하기 때문에 생기는 단점으로 해석할 수 있다.

③ 시간제약 조건을 다루는 섬세한 시간 함수들이 도매인 엔지니어들이 쉽게 정의하고 이해하여 사용하기에 부적합하다. 또한, FOD에서 별도의 기호를 사용함으로써 FOD의 복잡도를 증가시킨다.

3. NuSCR

NuSCR은 기존에 제시된 AECL의 SCR에 기반 하는 방법론을 수정, 보완하여 원자력 발전소의 제어 시스템 소프트웨어를 명세 하는데 보다 유용하게 사용될 수 있도록 개발된 정형명세 기법이다[9]. 기본적으로 SCR과 마찬가지로 Parnas' Four-Variable Model[18]에 기초하며, 이 모델에서 정의되는 relations를 보다 명확하게 명세하기 위해서 세 가지의 추가적인 변수 모델을 사용한다. 이는 function variable, history variable, timed-history variable로서 각각 SDT(Structured Decision Table), FSM(Finite State Machine), TTS(Timed Transition System)[19]를 이용해서 표현된다.

각각의 세 변수 모델들은 서로 다른 행위를 명세 하는데 유용하게끔 특징지어져 있다. Function variable은 수학적인 함수 관계를 표현하는데 사용되며 SDT와 같은 테이블을 사용한다. History variable은 수학적인 함수 보다는 상태(state)를 중심으로 명세할 때 보다 쉽게 명세되는 내용을 표현할 때 사용된다. History variable은 오토마타의 일종인 FSM으로 표현된다. 또한, Timed-history variable은 시간제약 조건이 추가된 내용을 명세할 때 사용되며, 시간 개념이 추가된 오토마타의 일종인 TTS를 사용한다. 이와 같이 서로 다른 특성을 지니는 변수들 간의 관계는 DFD의 일종인 FOD에 의해서 표현된다. 각각의 변수들은 FOD 상에서 하나의 독립된 노드(node)로서 표현되며, 각 노드들은 입력과 출력을 갖는다. 또한 FOD는 계층적으로 표현된다. 이와 같은 여러 요소들을 포함하고 있는 NuSCR의 원활한 사용을 위해서 현재 명세 지원 도구인 NuEditor가 개발되어 사용되고 있으며, 이 도구의 특징에 대한 설명은 뒤의 4장에서 자세히 소개하겠다. 각 변수들과 FOD를 포함한 전체 시스템에 대한 자세한 정형 정의(formal semantics)는 [9]에 설명되어 있다. 본 장에서는 논문 전체에서 예제로 사용될 KNICS RPS BP의 일부분에 대한 NuSCR 명세를 소개함으로써 NuSCR을 직관적으로 설명하고 있다. 또한, 기존의 AECL이 제안한 명세 기법을 NuSCR과 비교 설명함으로써 2장에서 제시한 AECL 방법론의 장단점을 객관적으로 설명하고 있다.

Function Overview Diagram 그림 1은 *g_SG1_LVL_Lo_RPS*에 대한 FOD의 일부로서, "g_"는 FOD 상에서 hierarchy를 가지는 중간 단계의 노드임을 의미한다. 좌우의 사각형으로 표현된 노드들은 FOD의 hierarchy 상에서의 입출력을 의미한다. (a)는 NuSCR을 이용해서 작성된 FOD이며, (b)는 AECL의 방법론을 이용해서 작성된 FOD이다. AECL의 FOD에는 시간 함수인 *t_Trip*과 상태 변수인 *s_X_Trip*이 모두 FOD상에 명시되어 있음을 알 수 있다. 반면에 NuSCR FOD에서는 이러한 시간 함수들과 상태 변수들을 모두 *th_X_Trip* 노드 내부에 명시하고 노드의 모양과 이름으로서 (확장자 "th_") 이를 구별할 수 있게 하고 있다.

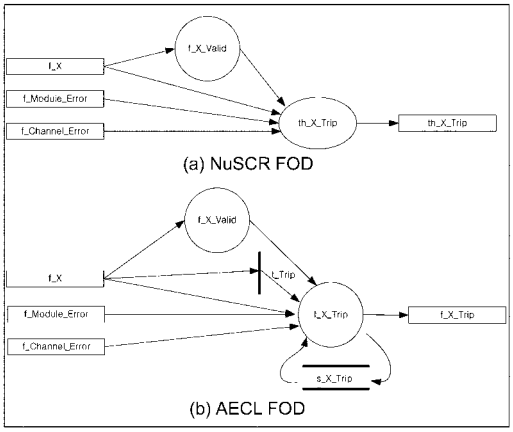

그림 1 Simplified FOD for *g_SG1_LVL_Lo_RPS*

Function variable node 아래의 표 1은 위의 FOD 중에서 function variable에 해당하는 *f_X_Valid* 노드에 대한 SDT이다. 두 종류의 SDT는 모두 다음과 같이 해석된다. "*f_X*의 값이 *k_X_MIN*과 *k_X_MAX* 사이에 존재하면 입력값이 올바르다는 의미의 출력값인 *f_X_Valid*에 0을 지정한다. 만약, 두 상수값 사이에 존재하지 않으면 1을 내보낸다." NuSCR의 SDT와 AECL 방법론의 SDT가 지니는 가장 큰 차이점은 한 줄에 표기하는 조건문의 크기이다. AECL에서는 기본적으로 하나의 단일 조건문 단위로 나누어 있는 반면에, NuSCR에서는 관련이 있는 조건문은 하나의 복합문으로서 한 줄에 표기하도록 하고 있다. NuSCR의 SDT는 AECL SDT가 지니는 테이블 크기 증가 문제를 상당 부분

표 1 SDT for *f_X_Valid*

Conditions		
k_X_MIN <= f_X <= k_X_MAX	T	F
Actions		
f_X_Valid := 0	X	
f_X_Valid := 1		X

(a) NuSCR SDT

Conditions			
f_X >= k_X_MIN	T	F	-
f_X <= k_X_MAX	T	-	F
Actions			
f_X_Valid := 0	X		
f_X_Valid := 1		X	X

(b) AECL SDT

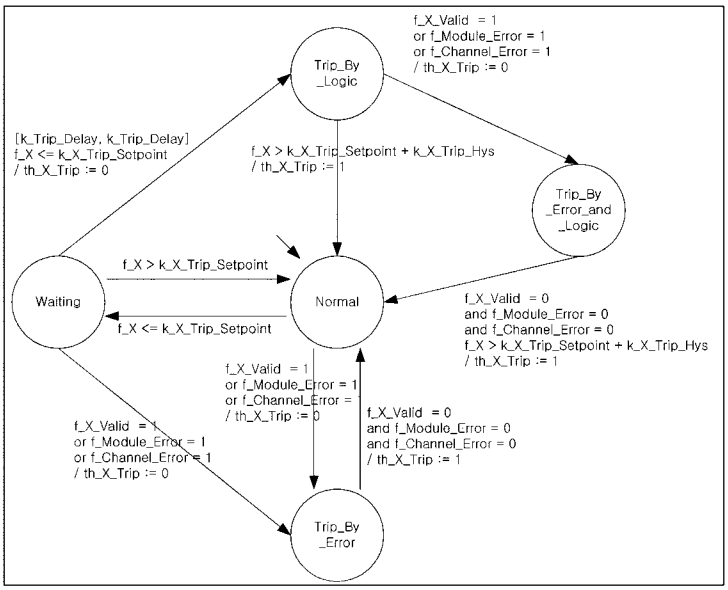

그림 2 Timed history variable definition for *th_X_Trip*

해결할 수 있으며, 또한 테이블의 가독성도 향상시킬 수 있다.

Timed history variable node 그림 1의 (a)에서 *th_X_Trip*은 timed history variable node이며, 변수의 상태 정보와 시간제약 조건을 모두 포함하는 오토마타의 일종인 TTS로 표현된다. TTS는 시간 조건이 [a, b]의 구간 형태로 추가된 FSM이다. TTS는 FSM에서 커다란 의미의 변화 없이, 원자력 발전소 제어 시스템에서 사용되는 시간 조건인 타이머 기능을 충분히 명세할 수 있다는 특징이 있다. (a)의 *th_X_Trip* 노드와 동일한 역할을 하는 (b)의 부분은 SDT 테이블을 이용해서 정의되는 *f_X_Trip* 노드와 수직바 형태로 표현되는 시간제약 함수인 *t_Trip* 및 상태 변수인 *s_X_Trip* 부분이다. 아래의 그림 2는 *th_X_Trip*에 대한 NuSCR 명세이며, 표 2는 이와 동일한 의미를 지니는 AECL 명세이다. 표 2에는 시간제약 함수인 *t_Trip*과 *t_Wait*에 대한 간략한 정의가 포함되어 있으며, 보다 정확한 정의는 생략하였다. 그림 2에서 제시된 *th_X_Trip* 정의는 5장에서 자세히 설명하고 있다.

NuSCR의 오토마타는 정의하는 대상의 행위를 시간 흐름의 순서대로 파악하여 기술하는 형태를 취하고 있으므로, 개별적인 조건들을 나열한 후 이들의 성립 여부를 표시하는 AECL의 기법에 비해서 명세하기가 훨씬 수월하고 해석 또한 용이하다. 이는 원자력 엔지니어들의 KNICSR DPPS RPS 명세 과정에서 직접 확인된 결과이다. 또한, NuSCR에서는 시간제약 조건들이 오토마타 노드의 정의에 포함되므로, 시간제약 함수들의 구별하여 정의할 필요가 없으며, FOD도 간략해져서 FOD에 대한 가독성도 향상시킬 수 있다.

표 2 AECL table for *f_X_Trip* and timing function *t_Trip*

Conditions							
f_X_Valid = 1	F	F	F	T	-	-	F	F
f_Module_Error = 1	F	F	F	-	T	-	F	F
f_Channel_Error = 1	F	F	F	-	-	T	F	F
f_X <= k_X_Trip_Setpoint	T	F	T	-	-	-	-	-
f_Trip_Basing function	F	-	T	-	-	-	-	F
f_X> k_X_Trip_Setpoint + k_X_Trip_Hys	-	-	-	-	-	-	T	-
t_th_Trip = 0	F	F	F	-	-	-	T	F
Actions								
f_X_Trip := 0				X	X	X	X		
f_X_Trip := 1	X	X						X	X
s_X_Trip := 0 (state variable)				X					
s_X_Trip := 1 (state variable)	X	X			X	X	X	X	X

t_Trip = t_Wait(0, k_Trip_Delay, k_Trip_Delay_Tol)
where
0 = f_X <= k_X_Trip_Setpoint

t_Wait(0, -, -) : System 's wait timer function

4. KNICS RPS의 BP & CP에 대한 NuSCR 정형명세

KNICSR RPS KNICS에서 현재 개발하고 있는 원자력 발전소의 발전소보호계통(Plant Protection System)은 크게 원자로를 보호하는 계통인 RPS(Reactor Protection System)와 냉각제 상실사고(LOCA) 등의 사고 발생시 사고 완화를 위해 작동되는 ESF-CCS (Engineering Safety Features - Component Control

System)로 구성된다. 본 연구에서 NuSCR을 사용해서 정형명세를 수행한 부분은 RPS의 BP와 CP로서 DPPS를 구성하는 가장 중요한 부분이다. RPS는 동일한 기능을 수행하는 4채널의 BPs(Bistable Processor A,B,C,D)를 통해서 안전변수가 트립 설정치를 초과하는지를 감시하고, CP(Coincidence Processor)에서 voting을 실시하여 4 채널의 BP 중 2개 이상이 안전하지 않다고 판단되면 최종적으로 트립(즉, shutdown) 신호를 발생시키는 기능을 수행한다. 각 채널의 BP와 CP는 모두 이중화되어 있으며, 일정한 주기에 따라 동작한다. 또한 이들 간의 통신은 안전 데이터 링크(Safety Data Link)를 통해서 보장된다. RPS에 대한 개략적인 구성도는 다음의 그림 3과 같다. 각각의 BP 및 CP는 PLC (Programmable Logic Controller)로 구현된다.

NuSCR Specification for BP 각각의 BP는 모두 18개의 센서 입력을 주기적으로 받아서, 각 입력에 따라 필요한 트립 논리 비교 연산을 수행하여 원자로의 안전 상태를 판단하게 된다. 각 입력들은 크게 4 가지의 트립 논리로 구분할 수 있으며, 각각은 모두 독립적으로 수행되는 특성이 있다. 그림 4는 RPS BP에 대한 NuSCR 명세의 일부로서 가장 상위의 FOD인 *g_BP*를 표현하고 있다. NuSCR 명세를 지원하는 도구로서 개발된 NuEditor를 이용해서 명세하는 화면으로서, NuEditor의 기능 및 특징에 대해서는 후에 보다 자세히 설명하겠다. 노드 *g_BP*의 왼쪽에 있는 노드들은 BP로의 입력을 의

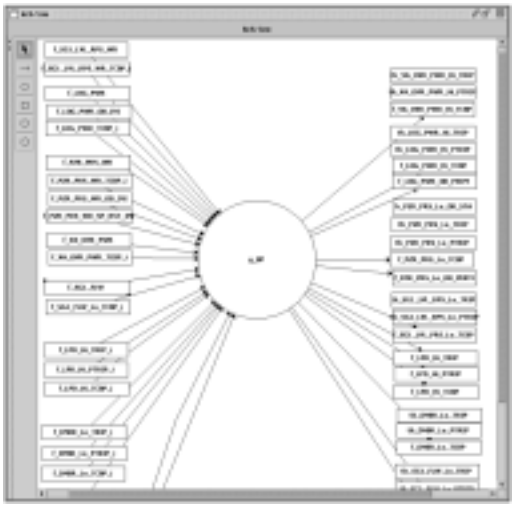

그림 4 FOD for *g_BP*

미하며, 오른쪽에 있는 노드들은 BP로부터 계산되어 발생되는 출력들을 의미한다.

그림 5는 최상위 노드인 *g_BP*에 대한 한 단계의 상세 명세를 수행한 결과에 대한 NuEditor 화면이다. NuSCR 지원 도구인 NuEditor의 좌측 부분은 FOD의 계층구조를 표현하고 있으며, 오른쪽에는 편집 화면이 위치한다. 앞에서 언급한 BP의 특성에서 유추할 수 있듯이 "g_"로 표현된 각 노드들은 모두 각각의 입력 변

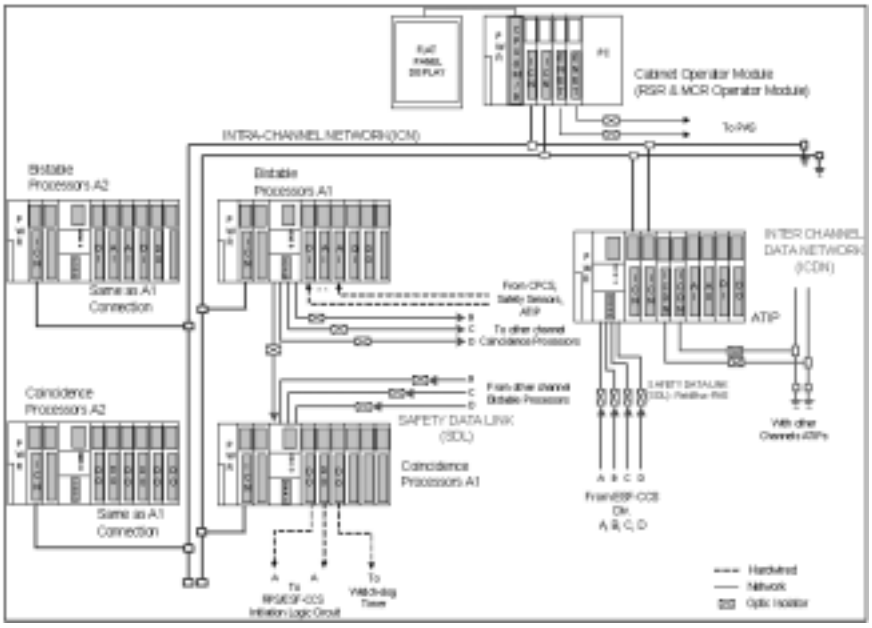

그림 3 DPPS RPS 구성도

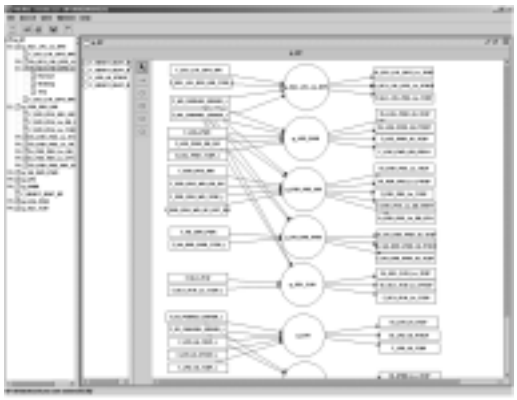

그림 5 NuEditor Screen-dump for *g_BP*

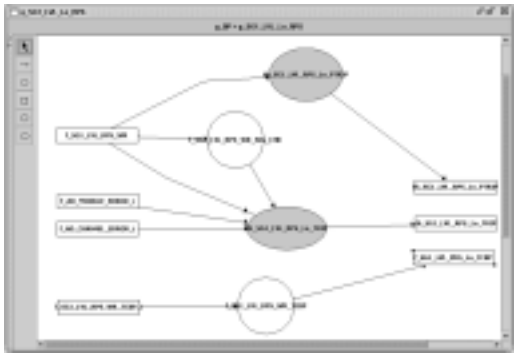

그림 6 FOD for *g_SG1_LVL_Lo_RPS*

수에 대한 트립 논리를 대표하고 있으며, 이들간에는 상호교류가 없음을 알 수 있다. 이러한 독립적인 특성은 각 트립 논리 부분을 모듈화 시킬 수 있으며, 이는 후에 각 논리가 주어진 자연어 명세대로 정확하게 동작하는가를 증명하기 위해서 정형검증 기법을 적용할 때에도 유용하게 사용될 수 있다. 정형검증과 관련해서는 결론부에서 다시 언급하겠다.

그림 5의 BP를 구성하는 18개의 독립적인 모듈 중에서 가장 상위에 위치한 *g_SG1_LVL_Lo_RPS*는 8가지의 트립 논리 중에서 고정설정치 하강 트립에 해당된다. 이는 트립 설정치가 고정되어 있으며, 입력값 즉 입력받은 센서의 값이 트립 설정치 보다 밑으로 하강하게 되면 트립이 발생되는 논리를 의미한다. 이 모듈에 대한 FOD는 다음의 그림 6이며, 2장에서 소개되었던 그림 1(a)가 그림 6의 한 부분이 된다. 그림 6에서 원으로 표현된 function variable 노드인 *f_SG1_LVL_RPS_WR_SIG_CHK*는 표 1(a)에서 정의되어 있으며, 타원으로 표현된 timed history variable 노드인 *th_SG1_LVL_RPS_Lo_TRIP*는 그림 2에서 정의되어 있다.

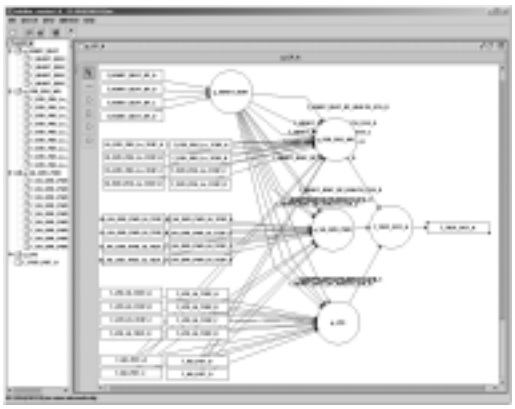

그림 7 FOD for *g_CP*

Timed history variable node인 *th_SG1_LVL_RPS_Lo_TRIP*의 정의에 대한 자세한 설명은 다음 장에서 다루도록 하겠다.

NuSCR Specification for CP 다음의 그림 7은 CP에 대한 NuSCR 명세 중 일부이다. FOD의 형태에서 유추할 수 있듯이 CP에서는 4개의 독립된 BP들로부터 18개의 입력 변수들의 트립 여부를 의미하는 입력들을 받아서 voting을 실시한 후 최종적으로 트립의 여부를 나타내는 출력값 하나만을 출력하고 있다. CP 부분은 voting을 실시하는 논리가 중심이 되며, BP에 비해서 비교적 수월하게 명세할 수 있다.

NuEditor NuEditor는 NuSCR의 효과적인 사용을 위해서 개발된 명세 지원 도구이다. 단순한 NuSCR editing 기능 외에도 작성된 NuSCR 명세의 완전성(completeness)과 일관성(consistency)을 일정수준까지 보장해 줄 수 있는 기능이 내재되어 있다. 그림 5의 NuEditor 화면에서 좌측의 트리 구조는 FOD 전체의 계층 구조를 표현하는 트리로서 각 노드들간의 상하관계를 명확하게 알 수 있게 도와주는 역할을 한다. 현재 NuEditor에서 제공되는 기능들과 앞으로 추가될 기능들을 정리하면 다음과 같다.

(1) NuSCR editing 기능 (기본 기능)

(2) FOD의 계층구조 내에서 입력/출력에 대한 완전성/일관성 유지 기능

① 상위 노드에서 입/출력으로 명시된 변수들의 list가 하위 FOD에 표시된다(그림 5의 좌측 두 번째 list 부분).

② 입출력 변수들의 list에서 직접 drag해서 사용할 수 있다.

③ 입출력 변수들의 list에서 drag해서 사용되면 list에서 해당 변수가 사라지며, 사용되지 않으면 계속 잔재하게 된다.

④ 그룹 노드가 아닌 단일 노드들로부터 발생한 출력이 이름이 노드의 이름과 동일하게끔 강제한다 (그림 6의 노드의 이름과 우측의 출력 노드의 이름).

(3) 3 가지 단일 노드들의 입력/출력에 대한 완전성/일관성 유지 기능

① 입력으로 명시된 입력 데이터들의 사용유무가 편집 화면 좌측에 명시됨(그림 8의 *th_SG1_LVL_RPS_Lo_TRIP* 노드에 대한 편집창 좌측).

② 3 종류의 노드들 즉, SDT, FSM, TTS 정의에 대한 완전성/일관성 검사 기능(구현 중)

(4) 기존의 정형검증 도구들과의 연계를 통한 정형 검증

① 정형검증 도구들의 입력으로 사용될 XML 출력 발생(그림 8의 우측).

② 구조적인 완전성/일관성과 관련된 사항들을 정리증명기인 PVS를 이용하여 증명할 수 있도록 자동 입력 변환기 개발

③ SMV, SPIN 등의 모델 체커를 이용해서 정형검증을 수행할 수 있도록 자동 입력 변환기의 개발(구현 중)

그림 8 Screen-dump of NuEditor

5. NuSCR 정형명세를 통한 안전성 향상 사례연구

정형명세 기법을 적용함으로써 얻을 수 있는 가장 큰 이점은 바로 개발자로 하여금 개발하고 있는 시스템에 대해서 충분히 명확하게 고려하도록 유도해 주는 점일 것이다. 이는 모든 정형명세 기법이 지니는 공통적인 특징이다. 정형명세 기법은 개발자로 하여금

① 모든 입력과 출력을 명확하게 정리하게끔 유도한다.

② 개발자의 생각으로만 있었던 구체적인 수행 논리들을 명확하게 기술하게끔 함으로써 시스템에 대한 전반적인 이해를 향상시켜 준다.

③ 정형명세는 시스템이 지니고 있는 내부적인 가정들 즉, 일반적으로 인식되므로 생략되던 것들을 모두 명확하게 기술하게끔 함으로써 명세를 보다 완전(complete)하게 도와준다. 이렇게 공개된 가정들은 후에 정형검증을 수행할 때 매우 중요한 자료로서의 역할을 담당하게 되며, 디자인 단계에서도 매우 중요한 자료로서 사용되게 된다.

이와 같은 정형명세 기법의 공통적인 특징 외에, NuSCR은 다음과 같은 고유한 특징을 지닌다.

① 대부분의 정형명세 기법들은 수학적인 기호나 특수한 의미의 diagram을 사용하므로 일반적인 도메인 엔지니어들이 쉽게 배워서 사용하기에는 무리가 있다. 반면에, NuSCR은 도메인 엔지니어들에게 친숙한 테이블과 흐름도 만을 사용함으로써 현장의 개발자들이 쉽게 이해하고 사용할 수 있다. 이는 KNICS DPPS를 개발하고 있는 KAERI의 개발자들로부터 검증된 결과이며, 단 기간 내에 KNICS RPS 전체에 대한 NuSCR SRS가 완성된 사실이 이를 실증하고 있다.

② NuSCR은 원자력 발전소에서 사용하는 제어 소프트웨어를 명세하기에 쉽도록 정의되어 있다. 즉, 명세하려는 내용의 특성에 따라 3 종류- SDT, FSM, TTS로 분류하여 명세하도록 함으로써 보가 쉽고 간결(compact)한 명세가 가능하다.

③ NuSCR은 지원 도구인 NuEditor를 통해서 입력/출력에 대한 기본적인 완전성/일관성을 보장해 주고 있으며, XML 출력을 이용한 자동 검증도구와의 연계 또한 쉽게 할 수 있다.

본 논문에서는 RPS의 BP 부분에 대한 NuSCR 명세를 통해 개발자의 시스템에 대한 이해를 향상시킴으로써 논리적인 오류를 찾아 소프트웨어의 안전성을 향상시킨 사례를 자세히 소개하려 한다.

예제: 고정설정치 트립 결정 논리 본 연구에서 소개하고자 하는 부분은 BP의 고정설정치 하강 트립에 대한 논리 부분으로서, 그림 5,6의 *g_SG1_LVL_Lo_RPS*이 이에 해당된다. 이 트립 논리에 대한 NuSCR 명세는 앞의 표 1(a)와 그림 2에서 축약된 변수명을 사용하여 제시되었다. 그림 2의 *th_X_Trip* 변수는 그림 6의 *th_SG1_LVL_RPS_Lo_TRIP*에 해당한다. *th_SG1_LVL_RPS_Lo_TRIP* 변수의 트립 논리에 대한 요구사항은 다음과 같다.

이와 같이 제시된 요구 사항을 만족하는 트립 결정 논리에 대한 NuSCR SRS는 다음과 같다. 해석의 편의를 위해서 각 변수의 전체 이름을 사용하지 않고 생략된이름을 사용하겠다. 다음의 그림 9는 timed-history variable인 *th_X_Trip*를 정의하기 위해서 사용되는 입

력 변수들과 상수값들에 대한 정의이다. 변수 *th_X_Trip*은 트립 논리를 만족하면 0을 출력하며, 만족하지 않으면 1을 출력하는 변수이다.

Requirement for *th_SG1_LVL_RPS_Lo_TRIP*

① 공정 변수 값이 미리 결정된 설정치 보다 감소할 경우에 트립 신호를 발생한다.
② 트립이 발생하기 위해서는 일정 시간 동안 트립 상황이 지속 되어야만 트립 신호를 발생할 수 있다.
③ 트립 상태가 되면 트립 설정치는 정해진 히스테리시스만큼 증가하고, 트립 상태가 해제되면 처음의 트립 설정치를 다시 유지한다.
④ 채널 에러, 모듈 에러, 입력값 에러 등이 발생하면 바로 트립 신호를 발생한다.

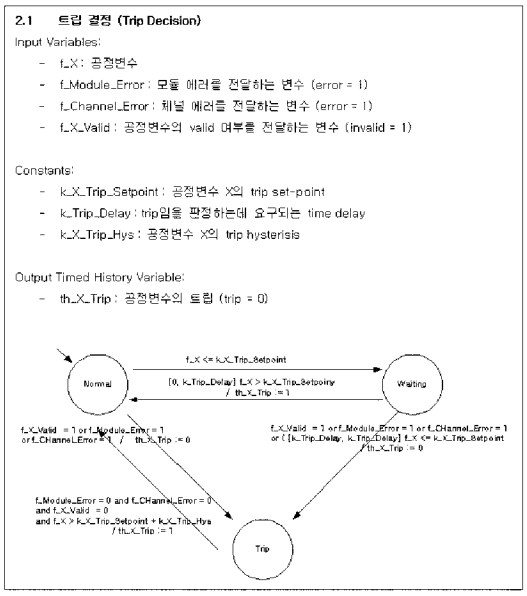

그림 9 NuSCR SRS for *th_X_Trip*

그림 9에서 정의된 *th_X_Trip*의 내용을 살펴보면 다음과 같다. 초기 상태는 *Normal*이며, 공정값이 트립 설정치 보다 작게 되면 *Waiting* 상태로 전이한다. 이 상태에서는 *[k_Trip_Delay, k_Trip_Delay]*으로 표현된 것과 같이 일정 시간 동안 이 조건이 계속 만족할 경우에만 출력값을 0으로 내보내며 *Trip* 상태로 전이한다. *Normal*과 *Waiting* 상태에서는 채널 에러나 모듈 에러, 공정값의 에러 등의 외부 에러가 발생하면 바로 *Trip* 상태로 전이하면서 출력값을 트립을 의미하는 0으로 바꿔준다. 일단 트립이 발생한 후에는 모든 외부 에러들이 정상적이고 공정값이 원래의 트립 설정값에서 히스테리시스 값을 더한 값 보다 상승할 경우에만 출력값을 정상인 1로 바꿔주면서 *Normal* 상태로 복귀하게 된다.

이와 같이 NuSCR을 이용해서 SRS를 작성하면서 개발자들은 다음과 같은 문제점을 발견했다. 이는 채널 에러나 모듈 에러, 공정값 에러 등이 발생해서 *Trip* 상태가 된 경우에도 다시 *Normal* 상태로 복귀하기 위해서는 히스테리시스가 추가된 비교연산을 만족시켜야 한다는 것이다. 따라서, 공정값 f_X가 원래의 트립 설정값과 히스테리시스를 더한 값 사이에 있는 경우에는 즉, $k_X_Trip_Setpoint < f_X < k_X_Trip_Setpoint + k_X_Trip_Hys$인 경우에는 외부 에러들이 모두 정상으로 돌아왔다 하더라도 다시 *Normal* 상태로 복귀할 수 없다. 개발자들이 이와 같은 문제점을 발견할 수 있었던 이유는 NuSCR이 테이블과 같은 나열식으로 명세하는 것이 아니라, 필요에 따라서 적합한 형태로 명세할 수 있도록 하는 유동성에 있다. 개발자는 *Normal*, *waiting*, *Trip* 등의 여러 상태들 간의 전이를 오토마타를 이용해서 작성함으로써 트립 논리의 상태 흐름을 자연스럽게 파악할 수 있었으며, 이를 통해 숨겨져 있던 오류 발생 상황을 발견할 수 있었다.

이러한 문제점이 발생하는 이유는, 공정값에 대한 트립 비교 논리에 의해서 트립이 발생하지 않은 경우에도, 이 논리에 의해서 트립이 발생한 것과 동일한 후속 조치를 적용했기 때문이다. 따라서 이 문제를 해결하기 위해서는 3장에서 제시된 그림 2와 같이 외부 에러에 의한 *Trip*과 내부의 트립 논리에 의한 *Trip*을 구분해서 정의하여야 한다. 하지만, *th_X_Trip*에 대한 정의를 그림 2와 같이 수정하는 과정에서, 개발자들은 다음과 같은 소프트웨어 구동 환경(operational environment)에 대한 중요한 가정을 발견했다.

Operational Assumption 1 채널 에러나 모듈 에러, 공정값 에러 등과 같은 외부 에러가 발생한 경우에는, 에러를 유발시킨 원인을 제거하기 위해서 PLC 전체를 다시 Reset한 후 재가동하는 작업이 항상 수반된다.

일단 외부 에러에 의한 Trip이 발생한 경우에는 전체 PLC 시스템을 Reset 한 후 재가동하므로 초기 상태부터 다시 시작하게 된다. 따라서 경우를 분리하여 *th_X_Trip*을 정의하지 않고 처음의 정의를 사용하더라도 결과적으로는 문제가 발생하지 않음을 예측할 수 있다.

하지만, 문제의 소지가 될 수 있는 상황이 아직 남아 있다. 현재는 **Operational Assumption 1**에 의해서 두 명세가 동일한 행위를 한다는 것이 보장되지만, 만약 **Operational Assumption 1**이 충분한 정보에 기반하지 않았거나 또는 다른 가정으로 인해서 수정될 수 있다면, 이를 기반으로 작성된 그림 9의 명세는 오동작을 일으킬 수 있다. 만약, 채널 에러나 모듈 에러, 공정값 에러 등과 같은 외부 에러가 노이즈 등의 이유로 인해서 일

시적으로 발생할 수 있다면, 이러한 외부 에러를 처리하는 운전원의 행위를 고려해야 한다는 판단을 내릴 수 있다. 이러한 판단으로부터 제어 소프트웨어의 구동 환경에 대한 추가적인 조사를 통해 개발자들은 다음과 같은 추가 보완적 성격의 가정을 추가할 수 있었다.

Operational Assumption 2 채널 에러나 모듈 에러, 공정값 에러 등과 같은 외부 에러가 발생한 경우에, 운전원은 이 에러가 일시적인 문제일 수도 있으므로 일정 시간 동안 에러가 지속되는 가를 확인한 후에야 후속 조치를 취한다

따라서, 위의 **Operational Assumption1, 2**에 의하여 그림 9와 같이 정의된 *th_X_Trip*은 처음에 예측했던 오동작을 일으킬 수 있으므로, 그림 2에서의 수정된 정의를 사용해야 한다고 결론 지을 수 있다. 또한, 앞에서 제시된 4가지 외에 위의 두 가정을 추가해서 다음과 같은 보다 완전한 고정설정치 하강 트립의 트립 결정 논리에 대한 요구사항을 완성할 수 있었다.

Refined Requirement for *th_SG1_LVL_RPS_Lo_TRIP*

① 공정 변수 값이 미리 결정된 설정치 보다 감소할 경우에 트립 신호를 발생한다.
② 트립이 발생하기 위해서는 일정 시간 동안 트립 상황이 지속 되어야만 트립 신호를 발생할 수 있다.
③ 트립 논리에 의해서 트립 상태가 되면 트립 설정치는 정해진 히스테리시스 만큼 증가하고, 트립 상태가 해제되면 처음의 트립 설정치를 다시 유지한다.
④ 채널 에러, 모듈 에러, 입력값 에러 등이 발생하면 바로 트립 신호를 발생하며, 이 신호를 보고 운전원이 에러를 발생시킨 원인들이 해결되고 PLC 시스템이 다시 리셋을 시키면, 정상 상태로 돌아온다.
⑤ 트립 논리에 의한 트립 신호와 에러에 의한 트립 신호가 동시에 들어온 경우에, 일시적인 문제로서 잠시 후에 에러에 의한 트립 신호가 해제된 경우에는 ③에 의해서 동작한다. 에러에 의한 트립 신호가 지속될 경우에 운전원에 의해서 ④와 같이 처리된다.

본 연구에서는 정형명세 기법인 NuSCR을 DPPS RPS BP에 적용함으로써, 위와 같은 논리적인 오류를 발견할 수 있었을 뿐만 아니라, 숨겨진 가정들을 명확하게 드러냄으로써 보다 명확하고 완전한 소프트웨어 요구사항 명세를 작성할 수 있었다. 제시된 예 외에도 개발자의 시스템에 대한 이해를 높임으로써 보다 명확한 요구사항을 작성할 수 있도록 도와준 사례를 다수 발견할 수 있었다. 이러한 경험을 통해 우리는 정형명세 기법인 NuSCR이 원자력 발전소의 디지털 제어 시스템 소프트웨어를 명세 하는데 유용하게 사용되었음을 확인할 수 있었으며, NuSCR이 정형명세 기법으로서의 장점 외에도, 원자력 분야의 소프트웨어를 명세 하는데 적합하게 개발된 기법이라는 점을 실증할 수 있었다.

6. 결론 및 향후 연구 계획

본 논문에서는 원자력 발전소의 디지털 제어 소프트웨어를 대상으로 개발된 정형명세 기법인 NuSCR을 소개하고, KNCIS RPS를 대상으로 하여 직접 적용해 본 경험을 소개하고 있다. 정형기법으로서 NuSCR은 개발자의 시스템에 대한 이해를 높여줄 뿐만 아니라, 생략되기 쉬운 가정들을 명확하게 드러냄으로써 명확하고 완전한 소프트웨어 요구 명세를 작성할 수 있도록 지원해 준다. 뿐만 아니라, NuSCR은 원자력 분야에 종사하는 엔지니어들에게 익숙한 표기 방법을 채택함으로써, 현장의 개발자들이 큰 무리 없이 약간의 교육만 받으면 바로 사용할 수 있다는 장점을 지니고 있다. 이러한 사용의 편의성은 KNICS DPPS RPS를 NuSCR을 이용해서 명세 하는 과정을 통해서 현장의 엔지니어들에 의해 인정받고 있다.

NuSCR은 정형명세 기법으로서의 역할을 충실히 수행하기 위해서 NuSCR 명세를 지원하는 지원 도구인 NuEditor를 개발하여 효과적인 정형명세를 지원하고 있다. 정형명세 도구인 NuEditor은 명세 하려는 시스템의 전체를 체계적으로 명세할 수 있게끔 지원하며 또한, 입력/출력간의 기본적인 완전성/일관성 문제도 도구 상에서 충족시켜 줄 수 있다. 이와 같이 NuEditor를 통해서 작성된 NuSCR 명세는 XML 포맷의 데이터로 변환되어 사용될 수 있는데, 이는 NuSCR 정형명세를 대상으로 기존의 정형검증 기법을 적용하기 위함이다. 모델 체킹과 같은 자동 정형검증 기법을 적용하기 위한 연구가 현재 진행 중이며, NuSCR 정의상의 특성으로 인해 정적(static)으로 미리 검증할 수 있는 내용 즉, FOD의 완전성이나 FSM의 determinism 등의 내용에 대해서는 지원 도구인 NuEditor 상에서 검사를 수행해 줄 수 있도록 하기 위한 연구도 또한 진행 중에 있다. 또한, 작성된 NuSCR 명세로부터 디자인 및 구현 단계의 내용인 FBD(Function Block Diagram)로의 원활한 변환을 위한 연구도 함께 진행중이다.

참 고 문 헌

[1] Nancy G. Leveson, SAFEWARE, System safety and Computers, Addison Wesley, 1995.
[2] Doron A. Peled, SOFTWARE RELIABILITY METHODS, Springer, 2001.
[3] Edmund M. Clarke and Jeannette M. Wing, "Formal Methods: State of the Art and Future Directions," ACM Computing Survey, 1996.
[4] E. A. Emerson, Edmund M. Clarke and A. P. Sistla, "Automatic verification of finite-state con-

current system using temporal logic specification," ACM Trans. Programming Languages and Systems, 8(2):244-263, 1986.
[5] D. vam Dalem, Logic and Structure, Springer-Verlag, 3 edition, 1994.
[6] U.S. NRC, "Digital Instrumentation and Control Systems in Nuclear Power Plants: safety and reliability issues," National Academy Press, 1997.
[7] KNICS, Korea nuclear instrumentation and control system research and development center, http://www.knics.re.kr
[8] UK MoD, The procurement of safety critical software in defense equipment, Define Standard 00-55, 1997.
[9] J. Yoo, T. Kim, S. Cha, J. Lee, and H. S. Son, "A Formal Software Requirements Specification Method for Digital Nuclear Plants Protection Systems," Journal of Systems and Software, accepted.
[10] J. M. Apivey, Introducing Z: a Specification Language and its Formal Semantics, Cambridge University Press, 1988.
[11] C. B. Jones, Systematic Software Development Using VDM, Prentice-Hall International, 1986.
[12] J. Gutting and J. Horning, Larch: Languages and Tools for Formal Specification, Springer-Verlag, 1993.
[13] K. L. Heninger, "Specifying software requirements for complex systems: New techniques and their application," IEEE Trans. Software Engineering, SE-6(1):2-13, 1980.
[14] D. Parnas, A. J. Schouwen Van, and J. Maday, "Documentation of requirements for computer systems," In RE'93: IEEE International Symposium on Requirements Engineering, 198-207, 1993.
[15] D. L. Parnas and J. Madey, "Functional documentation for computer systems," Science of Computer Porgramming, 25(1):41-61, 1995.
[16] K. H. Britton, R. A. Parker, D. L. Parnas, et, al., "Software requirements for the A-7E aircraft," NRL 9194, Naval Research Laboratory, Washington, D.C., 1992.
[17] Wolsong NPP 2/3/4, Software requirements specification for shutdown system 2 PDC, 86-68350-SRS-001, June 1993.
[18] D. Parnas and J. Madey, "Functional documentation for computer systems engineering," CRL 237, Telecommunications Research Institute of Ontario(TRIO), McMaster Univ., Hamilton, Ontario, 1991.
[19] Zphar Manna, Thomas A. Hensinger, and Amir Pnueli, "Timed transition systems," In REX Workshop, 226-251, 1991.

유 준 범
1999년 홍익대학교 컴퓨터공학과 학사 2001년 KAIST 전자전산학과 전산학전공 석사. 2001년~현재 KAIST 전자전산학과 전산학전공 박사과정. 관심분야는 소프트웨어공학, 안전성분석, 정형검증 및 명세

차 성 덕
1983년 University of California, Irvine 전산학 학사. 1986년 University of California, Irvine 전산학 석사. 1991년 University of California, Irvine 전산학 박사. 1994년~현재 KAIST 전자전산학과 전산학전공 부교수. 관심분야는 정형기법 및 명세, 정보보호, 침입탐지

오 윤 주
2002년 연세대학교 기계전자공학부 컴퓨터공학전공 학사. 2004년 KAIST 전자전산학과 전산학전공 석사. 2004년~현재 LG전자 UMTS 연구소 차세대단말팀 연구원. 관심분야는 소프트웨어공학, 안전성분석

김 창 회
1995년 충남대학교 전자공학과 박사. 현재 한국원자력연구소 계측제어인간공학 연구부 책임연구원. 연구세부분야는 원자로보호계통 개발