# NuSCR 정형 요구 명세에서 UML2.0 Activity Diagram으로의 변환 규칙

손준익O 정세진 유준범

건국대학교 컴퓨터-정보통신 공학부

{sji6227, jsjj0728, jbyoo}@konkuk.ac.kr

# Transformation Rules from NuSCR Formal Requirement Specification to UML2.0 Activity Diagram

Junik SonO Sejin Jung Junbeom Yoo

Division of Computer Science and Engineering, Konkuk University

**요 약**

NuDE 2.0 환경에서 요구사항 단계에서는 NuSCR 정형명세를 대상으로 모델 체킹을 수행할 수 있는 검증 환경을 제공해주는데 모델 체킹의 기본적인 한계로 인해 시스템 전체를 대상으로 특정 조건들을 검증하기 위해서는 테스팅 기법이 추가적으로 필요하다. 그러나, NuSCR의 경우 시스템의 동기화와 병행성 및 데이터 흐름과 제어 흐름 간의 관계에 대한 명확한 정의가 없어 직접 테스팅 하는 것은 불가능하다. UML 2.0 Activity Diagram은 제어 흐름과 데이터 흐름 그리고 병행성을 명시 할 수 있어 소프트웨어 시스템의 동작을 모델링하는데 적합하다. 본 논문에서는 NuSCR을 테스팅 하기 위한 목적으로 NuSCR로 작성된 명세를 UML2.0 Activity Diagram으로 변환하는 규칙을 제안한다. NuSCR의 각 구성에 대해 동일한 의미를 가지는 Activity Diagram 대응 파트로 변환하는 규칙을 정의하고 변환 절차에 대해서 설명한다.

## 1. 서론

원자력 발전소 제어 시스템, 항공 시스템과 같은 안전 필수 시스템은 오류 발생시 단순한 미션 실패 뿐만 아니라, 대규모의 인명피해와 물적 손실을 야기할 수 있다. 따라서, 이러한 소프트웨어 시스템에 대해서는 철저한 확인 및 검증 활동이 수행되어야 한다.

NuDE2.0[1]은 원자력 발전소의 디지털 계측제어 시스템을 위한 정형 기법 기반 소프트웨어 개발, 검증 및 안전성 분석 환경이다. NuDE2.0은 검증 활동을 위해 소프트웨어 개발 각 단계별 테스팅 및 모델 체킹 환경을 제공한다. 요구사항 단계에서는 NuSCR[2] 정형명세를 SMV 입력 값으로 변환하여 모델 체킹을 수행할 수 있는 검증 환경을 제공한다. 그러나, NuSCR로 작성된 시스템 전체를 대상으로 모델 체킹을 수행할 경우 상태 폭발 문제가 발생할 수 있다. NuSCR로 작성된 원자로보호계통 시스템의 비교논리프로세서(Bistable Processor, 이하 BP로 표시)와 동시논리프로세서를 정형검증한 사례에서도 위의 문제가 발생하였다[3].

NuDE2.0 환경에서 NuSCR로 명세 된 시스템 전체를 대상으로 특정 조건들을 검증하기 위해서는 모델 체킹 검증 기법 이외에도 테스팅 기법이 필요하다. 기본적으로 테스팅을 수행하기 위해서는 대상 코드 또는 모델의 실행이 필요하다. NuSCR의 경우 데이터 흐름을 나타내는 FOD와 제어 흐름을 나타내는 여러 구성의 의미와 구문이 정형적으로 정의 되어 있지만 시스템의 서로 다른 모듈 간의 동기화 및 병행성(concurrent)에 대한 정의가 없다. 또한, 데이터의 흐름 관계와 제어 흐름 관계 간의 명확한 정의가 없어 NuSCR을 직접 실행하여 테스팅을 수행하는 것은 불가능하다.

UML 2.0 Activity Diagram[4](이하 AD로 표시)은 제어 흐름과 데이터 흐름 그리고 병행성을 명시 할 수 있어 다양한 소프트웨어 시스템의 동적인 동작을 모델링하는데 적합하다. 이러한 특징으로 인해 AD를 이용하여 병행 시스템을 테스팅 하는 연구[5], DFD (Data Flow Diagram)를 AD로 변환하는 연구[6], UML의 sequence diagram을 AD로 합성하는 연구[7] 등 여러 연구들이 진행되고 있다.

본 논문에서는 NuSCR을 테스팅 하기 위한 목적으로 NuSCR로 작성된 명세를 AD로 변환하는 규칙을 제안한다. NuSCR의 각 구성에 대해 동일한 의미를 가지는 AD의 대응 파트로 변환하는 규칙을 정의하고 변환 절차에 대해서 설명한다. 또한, 제안하는 방법의 유효성을 확인하기 위하여 원자로 보호계통의 비교논리프로세서를 이용하여 사례연구를 수행하였다.

## 2. NuSCR

NuSCR은 안전성과 정확성이 요구되는 원자력 발전소의 디지털 계측제어 시스템을 명세 하는데 적합하게 보완된 정형명세기법이다. NuSCR은 입출력 변수 이외에 3개의 기본 구성인 function variable, history variable, timed history variable 으로 구성이 되며 이러한 모든 구성의 관계는 FOD (Function Overview Diagram)로 표현된다. 또한, FOD는 그룹 노드를 이용하여 계층적으로 모델링 할 수 있다. 그림 1은 NuSCR로 작성된 BP와 g_LO_SG1_LEVEL의 FOD이다.

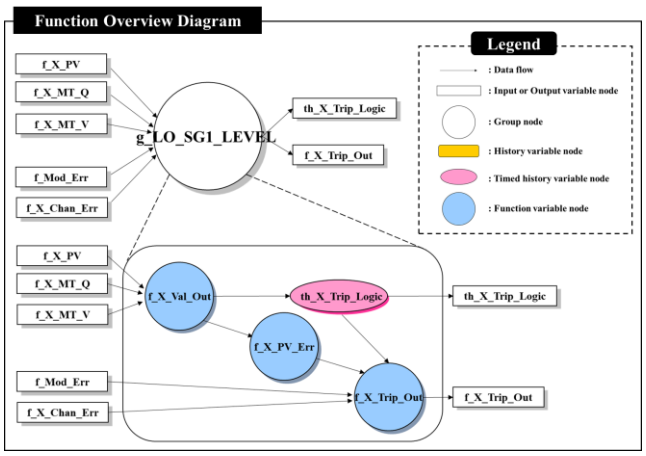
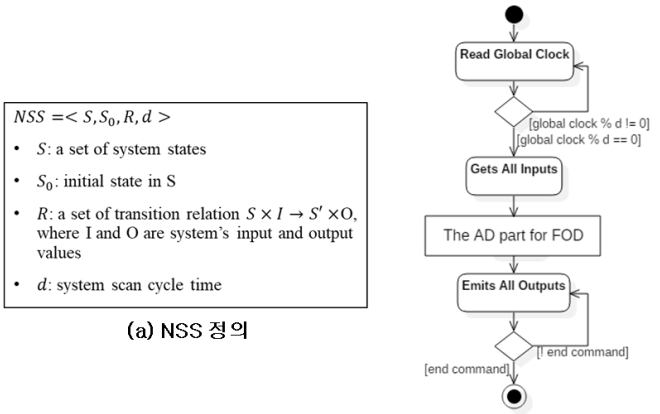



그림 1. g_LO_SG1_LEVEL만 있는 비교논리프로세서의 NuSCR 명세

## 3. Transformation from NuSCR to Activity Diagram

NuSCR의 AD로의 변환요소는 NuSCR로 작성되는 시스템을 나타내는 NSS (NuSCR Software System), FOD, SDT, FSM, TTS 이다.

**[Mapping Rule 0 (NSS)]** NSS는 시스템을 표현하는데 사용되며 그림2(a)는 NSS의 정의이다. NSS는 일정 주기마다 외부 환경에서 입력 값을 얻은 후 값을 이용하여 내부 계산 후 출력 값을 내보내는 동작을 한다. 그림 2(b)는 해당 동작에 대응하는 AD 파트이다. 내부 계산은 FOD에 의해 수행된다.

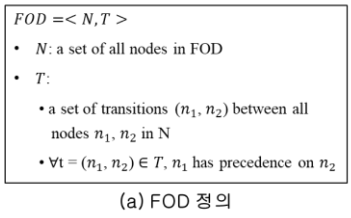



(a) NSS 정의

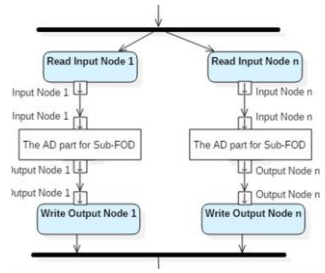
(b) NSS에 대응하는 AD 파트

그림 2. NSS 정의 및 대응하는 AD 파트

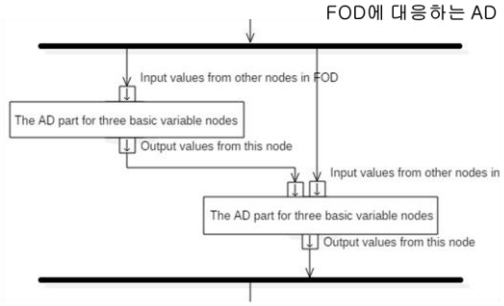
**[Mapping Rule 1 (FOD)]** FOD는 DFD의 일종으로 FOD 간의 계층을 표현하는데 사용되거나 NuSCR의 기본 구성들의 관계를 표현하는데 사용된다. 전자의 경우 그룹 노드가 사용된다. 그림 3(a)는 FOD의 정의이다. 그림 3(b)는 FOD 간의 계층을 표현하기 위하여 그룹 노드로만 구성된 FOD에 대응하는 AD 파트이다. 그림 3(c)는 3개의 기본 구성의 관계를 표현하기 위한 FOD에 대응하는 AD 파트이다. 파란색으로 채워져 있는 액션은 최상위 FOD를 변환할 때 한 번만 생성되는 액션이다. FOD는 데이터의 흐름과 병행성을 나타내므로 2개의 AD 파트는 동일하게 AD의 요소 중 데이터의 흐름을 나타내는 Pin과 병행성을 나타내는 Fork, Join 노드를 이용하여 구성되어 있다.




(a) FOD 정의



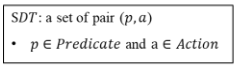

(b) 그룹 노드로만 구성된 FOD에 대응하는 AD 파트




(b) 3개의 기본 구성으로만 구성된 FOD에 대응하는 AD 파트

그림 3. FOD 정의 및 대응하는 AD 파트

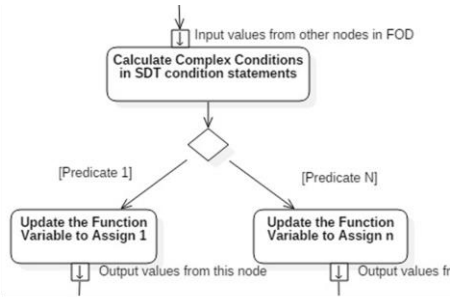
**[Mapping Rule 2 (SDT)]** SDT (Structured Decision Table)는 function variable을 정의하기 위해 사용되며 그림 4(a)는 SDT의 정의이다. 그림 4(b)는 그림1에 있는 f_X_Val_Out에 대한 SDT이다. SDT는 그림 4(b)에 있는 f_X_MT_Q=true 와 같은 조건들을 가지고 있으며 Predicate는 이들 간의 논리곱을 의미한다. SDT는 들어온 입력 값을 가지고 SDT에 있는 조건들을 계산하여 가능한 Predicate를 찾아 해당하는 Action으로 function variable의 값을 변경하는 식으로 동작하며 해당하는 AD 파트는 그림 4(c)이다.




(a) SDT 정의

| Conditions | | |
|---|---|---|
| f_X_MT_Q = true | T | F |
| **Actions** | | |
| f_X_V_O := f_X_MT_V | O | |
| f_X_V_O := f_X_PV | | O |

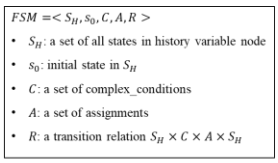
(b) SDT for f_X_Val_Out



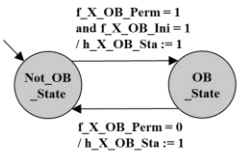

(c) SDT에 대응하는 AD 파트

그림 4. SDT 정의, 예제 및 대응하는 AD 파트

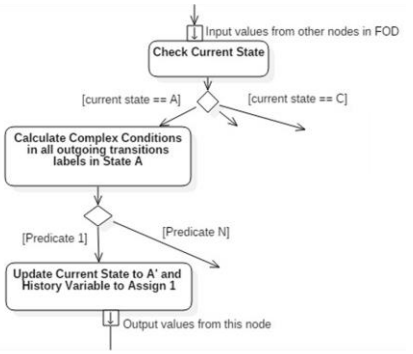
**[Mapping Rule 3 (FSM)]** FSM (Finite State Machine)은 history variable을 정의하기 위해 사용되며 그림 5(a)는 FSM의 정의이다. 그림 5(b)는 FSM의 예제이다. FSM은 상태, 전이, 전이의 라벨로 구성 되어 있으며 라벨은 SDT의 Predicate와 Action으로 구성되어 있다. FSM은 입력이 들어오면 현재 상태에서 나가는 모든 전이의 라벨에 있는 조건들을 계산하여 가능한 전이를 찾는다. FSM의 현재 상태를 전이에 목적 상태로 변경하고 해당하는 Action으로 history variable의 값을 변경하는 식으로 동작하며 해당하는 AD 파트는 그림 5(c)이다.




(a) FSM 정의




(b) FSM for h_X_OB_Sta




(c) FSM에 대응하는 AD 파트

그림 5. FSM 정의, 예제 및 대응하는 AD 파트

**[Mapping Rule 4 (TTS)]** TTS (Timed Transition System)는 timed history variable을 정의하기 위해 사용되며 그림 6(a)는 TTS의 정의이다. TTS는 FSM의 확장으로 전이에 시간제약조건이 포함된 것을 제외하고는 FSM과 동일하다. 그림 6(b)는 그림 1에 있는 th_X_Trip_Logic에 대한 TTS이다. TTS의 동작은 시간제약조건이 있는 전이를 제외하고는 FSM의 동작과 동일하다.

시간제약조건이 있는 전이의 동작은 전이의 Predicate가 만족했을 때 시간제약조건이 만족되었을 경우와 만족되지 않은 경우 2가지로 나뉘게 된다. 만족한 경우는 전이가 발생하며 TTS의 local clock을 0으로 초기화 한다. 만족하지 않은 경우는 전이가 발생하지 않으며 TTS의 local clock을 시간주기만큼 더하고 timed history variable 의 값을 이전 주기의 값으로 보존한다. TTS에 대응하는 AD 파트는 그림 6(c)이며 시간제약조건이 있는 전이에 해당하는 파트 이외에는 FSM에 대응하는 AD 파트와 동일하다.

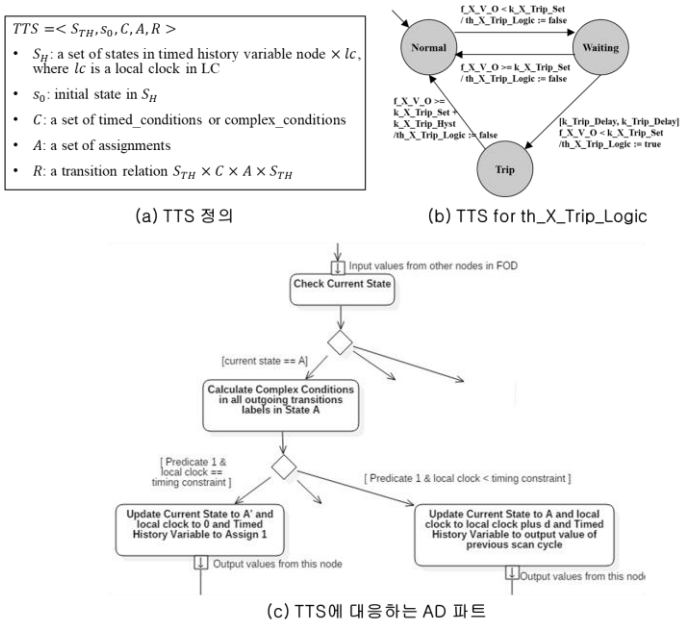



그림 6. TTS 정의, 예제 및 대응하는 AD 파트

[Transformation Process] 위의 규칙들을 이용하여 NuSCR를 AD로 생성하는 절차는 Top-down 방식을 이용한다. 먼저, 시스템을 나타내는 NSS를 이용하여 NSS 변환 규칙에 따라 AD파트를 생성하고 FOD 변환 규칙에 따라 최상위 FOD부터 말단 FOD까지 AD 파트를 생성한다. 말단 FOD의 경우 3개의 기본 구성에 대한 변환 규칙을 이용하여 FOD에 정의된 순서에 맞추어 AD 파트를 생성한다.

## 4. 사례연구

이 절에서는 제안하는 방법에 유효성을 확인한다. AD로의 변환 규칙을 적용한 시스템은 원자로보호계통 시스템의 BP이다. BP는 18개의 모듈을 가지고 있지만 사례연구에서는 고정 하강 트립 논리인 g_LO_SG1_LEVEL 모듈만 사용한다. 그림 1은 g_LO_SG1_LEVEL만을 포함함 BP의 FOD이다. BP의 FOD는 계층적으로 구성되어 있으며 하위 FOD로 g_LO_SG1_LEVEL를 가지고 있다. g_LO_SG1_LEVEL FOD는 5개의 입력 노드와 3개의 function variable node, 1개의 timed history variable node 그리고 2개의 출력 노드로 구성되어 있다. 해당 시스템의 스캔 주기는 20ms 이다.

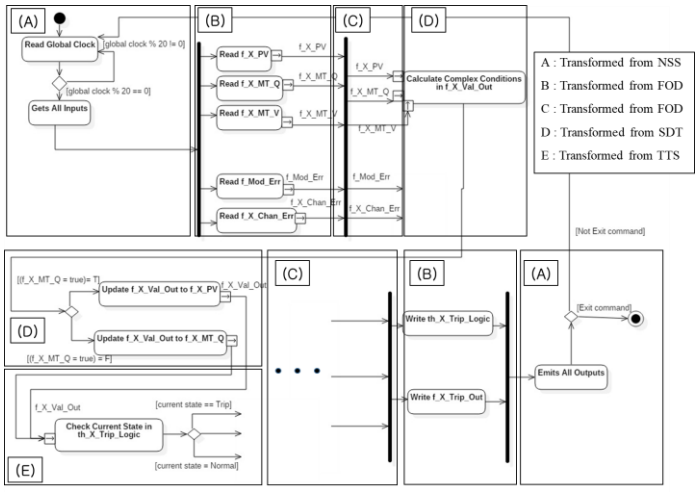



그림 7. AD for 비교논리프로세서

그림 7은 NuSCR로 작성된 BP를 변환 규칙 및 절차에 맞추어 변환한 AD이다. (A)는 NSS 규칙에 의해 생성된 AD 파트이고 (B)는 최상위 FOD인 BP에 의해 생성된 AD 파트이다. (C)는 BP의 하위 FOD인 g_LO_SG1_LEVEL의 FOD에 의해 생성된 AD 파트이다. (D)는 f_X_Val_Out에 대한 AD 파트로 SDT 변환 규칙에 의해 생성되었다. (E)는 th_X_Trip_Logic에 대한 AD 파트로 TTS 변환 규칙에 의해 생성이 되었고 (D)와 (E)는 g_LO_SG1_LEVEL의 FOD에 정의된 순서에 따라 생성이 되었다. 생성된 AD는 20ms 마다 입력을 받아 내부 계산이 실행이 되고 fork 와 join 노드로 인해 전체 실행에 대한 동기화가 수행되는 것을 확인할 수 있다.

## 5. 결론 및 향후 연구

본 연구에서는 NuSCR 정형 명세 언어를 UML2.0 Activity Diagram으로의 변환 규칙 및 절차를 제안하였다. AD로의 변환 규칙은 NuSCR의 각 구성들의 정의 및 동작을 이용하여 작성되었다. 또한, 제안한 방법의 유효성을 확인하기 위해 원자로보호계통 시스템의 BP의 한 모듈에 대해서 변환을 수행해 보았다.

향후 연구로는 제안한 방법의 변환 규칙을 정형적으로 증명하고자 한다. 제안한 방법이 정형적으로 증명이 된다면 변환을 기계적으로 수행하고 AD를 실행시켜 테스팅이 가능한 자동화 도구를 구현하고자 한다.


## Acknowledgement

이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A1B03030065)


## 참고 문헌


[1] Junbeom Yoo, et al. "NuDE 2.0: A model-based software development environment for the PLC & FPGA based digital systems in nuclear power plants." Integrated Circuits (ISIC), 2014 14th International Symposium on. IEEE, 2014.

[2] Junbeom Yoo, et al. "A formal software requirements specification method for digital nuclear plant protection systems." Journal of Systems and Software, vol. 74, No. 1, p73-83, 2005.

[3] 지은경 외 2인. “원자로보호계통 소프트웨어 안전 확보를 위한 모델 체킹 및 테스팅 적용 사례 분석과 발전방향 제고.” 정보과학회지 제 33권 제 7호, p15-26, 2015

[4] Object Management Group (OMG). Unified Modelling Language: Superstructure v2.0 (formal/05-07-04), July 2005.

[5] Chang-ai Sun, et al. “A transformation-based approach to testing concurrent programs using UML activity diagrams.” Software: Practice and Experience, vol. 46, No. 4, p551-576, 2016

[6] Fanchao Meng, et al. “Transformation from Data Flow Diagram to UML2.0 activity diagram.” Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on. vol. 2, p1010-1014, IEEE, 2010

[7] Sungwon Kang, et al. “Transformation rules for synthesis of UML activity diagram from scenario-based specification.” Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual. IEEE, 2010.