2000 년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

원자력 발전소 제어시스템의 정형 명세와 검증+

김일곤, 김진현, 남원홍, 최진영
고려대학교 컴퓨터학과
e-mail : {igkim, jhkim, wnam, choi}@formal.korea.ac.kr

Formal Specification and Verification of Nuclear Power Plant Control System

Il-Gon Kim, Jin-Hyun Kim, Won-Hong Nam, Jin-Yong Choi
Department of Computer Science & Engineering Korea University

요 약

원자력 발전소와 같이 시스템 오작동으로 인하여 엄청난 재난을 불러 올 수 있는 시스템은 시스템을 구축하기 이전에 완전한 설계 및 검증이 절대적으로 필요하다. 특히 긴급성을 요하는 원자력 발전소의 긴급 차단 시스템과 같은 실시간 safety-critical 시스템은 시스템 행위의 유기적인 측면뿐만 아니라, 시간적 제약을 고려하여 엄격하게 명세하고 분석해야 한다. 본 논문에서는 시각적 기반의 설계 명세 언어인 STATECAHRT 를 이용하여 원자력 발전소 제어 시스템을 명세하고 이를 모델 체킹 검증 도구인 SMV 로 검증함으로써 시스템의 신뢰성을 높이고 실시간 safety-critical 시스템의 설계 및 검증에 대한 방법론을 제시한다. 본 연구에서는 [6]의 논문의 명세 오류를 수정하여 명세 및 검증을 수행하였다.

1. 서론

컴퓨터로 제어되는 시스템의 효용이 증가하는 가운데 시간적 제약에 따른 시스템의 정확성이 요구되는 실시간 safety-critical 시스템에 대한 필요성이 산업체의 공정제어 및 공장 자동화, 첨단 군사 무기제어, 원자력 발전소의 통제, 항공관제 시스템, 멀티미디어 서비스와 같은 분야에서 대두되고 있다. 특히 원자력 발전소와 같이 시스템 오작동으로 인하여 환경적인 큰 재난을 불러 올 수 있는 시스템에 대한 완전한 설계 및 검증은 절대적으로 필요한 과제라 하겠다. 원자력 발전소의 긴급 차단 시스템과 같은 실시간 safety-critical 시스템은 시스템의 행위의 유기적인 측면 뿐만 아니라 시간적 제약을 고려하여 엄격하게 명세하고 분석해야 하며 그와 같은 대규모적인 시스템은 그 복잡성으로 인해 예상치 못한 잠재적 에러와 큰 재앙을 막기위해 많은 기술을 필요로 한다. 국내. 외적으로 원자력 발전소의 실시간 safety-critical 시스템을 설계 명세한 다양한 연구들이 진행되어 왔다. 많은 나라에서는 원자력 발전소에서 소프트웨어 기반의 비상 차단 시스템(Emergency shutdown system)에 소프트웨어 공학의 일종인 정형기법[1]을 이용하여 예기치 못한 오류를 검증하고 시스템의 효율성을 증가시키는 좋은 결과를 가져왔다. 하지만 정형기법을 도입한 설계는 수학적 기호나 전문용어로 기술된 사용자의 요구명세를 검증한 경우로, 전문가의 설명이나 지식이 없이는 이해하기 쉽지 않아 그 효용도가 낮은 편이였다. 본 논문에서는 시각적 언어인 STATECHART[2]언어로 명세하여 설계의 편이를 도모하고 설계자와 사용자 사이의 일관성을 유지시킴은 물론, 정형 검증 기법인 모델 체킹[3] 기법을 사용하여 명세된 시스템을 검증함으로 실시간 safety-critical 시스템의 일종인 원자력 발전소 일부분의 설계를 보다 안정적이고 신뢰성 있는 시스템으로 설계하고자 한다. 본 논문에서는 i-Logix 에서 만든 STATEMATE[4]을 이용해 원자력 발전소의 제어시스템을 STATECHART 로 명세하고 시뮬레이션 하며, STATECHART 명세를 모델 체커의 일종인 SMV[5]로 바꾸어서 검증한다. 본 논문에서 목표로 하는 전체 설계 기법의 세부 단계 방법은 <그림-1>과 같다. 본 논문은 2 장에서 STATECHART 에 대하여 설명하고 3 장에서는 정형 기법의 일종인 모델 체킹과 모델 체킹의 자동화 도구인 SMV 에 대하여 기술한다. 4 장에서는 [6]의 논문에 나와 있는 명세의 오류를 수정하여 실제 원자력 발전소의 차단 시스템[6]의 일부를 STATECHART 로 명세하고 이를 SMV 로 바꾸어 검증한다. 5 장에서는 본 논문에서 소개된 기법의 특징과 효용성을 기술하며 6 장에서는 결론 및 향후 연구 방향을 제시한다.

2. 시각적 명세 언어 STATECHART

STATECHART 는 원자력 발전 시스템과 같은 대규모의 reactive 시스템의 명세를 위하여 개발된 정형명세 언어로써

+본 연구는 기초 전력 공학 공동 연구소 지원으로 수행되었음

기존의 유한 상태 오토마타와 상태 전이 다이어그램의 확장된 형태의 상태 기반 정형 명세 언어이다. STATECHART 의 문법을 간략히 살펴보면 다음과 같다. STATECHART 는 관련된 이벤트가 발생할 때, 상태 전이를 표현한 것으로 상태(State)와 전이(Transition) 액션(Action), 변수(variable)로 이루어진다. 상태는 특정 시간동안 유지되고 구별되는 상태를 가리키고 전이는 이벤트에 응답하여 객체가 상태를 변화시키는 것과 같은 방법을 의미하며 액션은 어떤 상태로 진입되거나 진출할 때 이벤트가 전이를 발생시키는 것과 같은 단위적인 행동을 말한다. STATECHART 는 복잡한 상태를 표현하기 위해 상태의 계층구조와 병렬성을 AND 상태와 OR 상태를 표현하도록 하였으며 AND 상태에 속하는 하부 상태들은 병렬적으로 존재하는 상태로서 동시성(Concurrency)을 표현하는 수단으로 사용된다. 반면 OR 상태에 있다는 것은 그 상태의 전이는 그 상태의 하부 상태 중 특정한 하나의 상태에 있다는 것을 의미한다. 이러한 상태

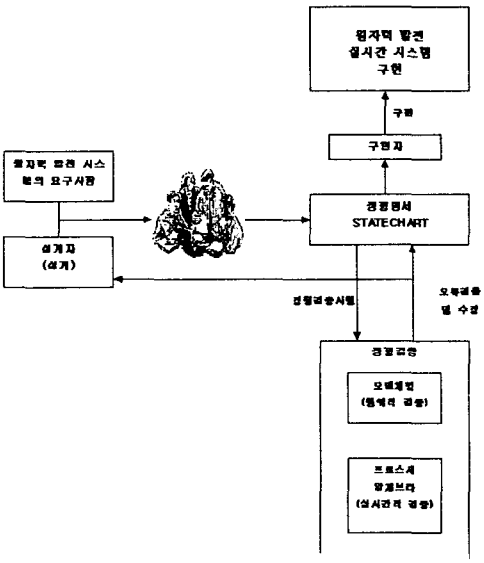

그림 1. 정형기법을 이용한 설계 및 구현

전이는 내부 이벤트를 발생시킬 수 있으며, 이렇게 발생된 이벤트는 이 이벤트와 관련된 시스템의 다른 모든 STATECHART 의 전이를 유도할 수 있다. 또한 STATECHART 는 동시성을 지니고 있어 이벤트에 의해 상태 전이가 일어나면, 그 상태 전이에 의한 내부 이벤트가 또 다른 상태 전이를 일으켜 연쇄 반응을 일으킨다.

STATECHART 는 State diagram 의 골격에다 계층과 동시성 그리고 Broadcasting 의 개념을 더한 것이라 할 수 있다.

<그림-2>은 간단한 STATECHART 를 보여준다. 이 다이아그램에서 사각형은 각 상태를 나타내고 화살표는 전이를 나타낸다. STATECHART 에서 전이 레이블(label)의 문법은 e(c)/a 이다. e 는 전이를 발생시키는 이벤트이고 (c)는 이벤트가 발생했을 때 전이를 가능하게 하는 조건문이다. a 는 전이가 일어날 때 발생되는 액션을 가리킨다. 각 상태간의 계층은 encapsulation 을 사용한다. <그림-2>에서 상태 A 와 상태 D 는 서로 자식과 부모간의 계층관계를 가지고 있고 서로 exclusive-or 관계로 같은 시간에 서로 다른 상태에 제어가 있어서는 안 된다. <그림-3>에서 각 상태 붙어있는 작은 화살표는 default 상태가 된다.

<그림-4>에서는 동시성을 보여주고 있다. 즉 가운데 점선은 상태 F 와 G 가 "AND"관계에 있음을 보여준다. 즉 상태 F 와 G 가 동시에 있는 것을 의미한다.

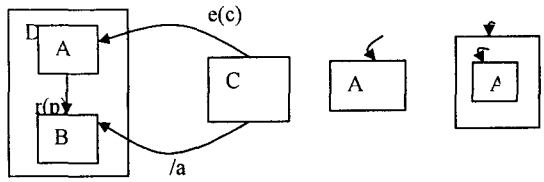

그림 2 : STATECHART

그림 3 : default state

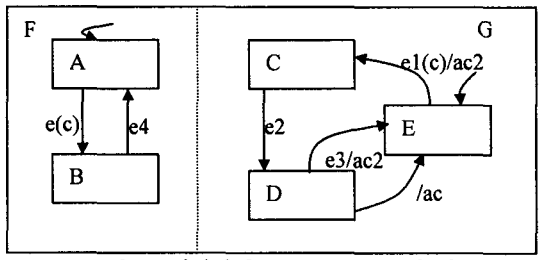

그림 4 : 병렬적인 STATECHART 의 예

3. 모델 체킹과 모델 체커 SMV

모델 체킹(Model Checking)은 유한 상태 동시 시스템을 검증하기 위한 오토마타 기반의 기술이다. 이것은 모의실험, 테스팅 및 연역적 추리(deductive reasoning)를 기반으로 문제를 해결하는 기존의 방법보다 나은 몇 가지 이점을 가지고 있다. 이것은 회로 설계나 통신 프로토콜을 검증하는 실용적인 방법 가운데 하나이다. 모델 체킹은 시스템을 오토마타의 형태로 표현할 수 있는 언어로 표현하고, 이 시스템이 만족해야 하는 요구 사항을 시제 논리로 표현하여 이 시스템의 도달할 수 있는 모든 상태에서 요구된 시제 논리를 만족하는가를 알아내는 검증 기법이다

3.1 SMV

SMV 시스템은 유한 상태 시스템이 CTL(Computational Tree Logic)로 표현된 요구 명세를 만족하는지를 검증하는 정형검증 도구이다. SMV 의 입력 언어는 유한 상태 시스템을 명세하기 위해 만들어 졌다. 시스템은 입력 언어를 통해서 동기적인 Mealy machine 이나 비동기적인 네트워크로 손쉽게 명세 될 수 있다. SMV 언어가 유한 상태 시스템을 위해 만들어진 것이기 때문에 언어가 제공하는 유한한 것(Boolean, scalar, fixed array)만을 제공한다. CTL 은 safety, liveness, farness, dead lock freedom 을 포함한 풍부한 종류의 시간적 특성(temporal property)들을 간단한 문법을 이용하여 표현하는 것이 가능하다. SMV 는 입력 언어로 나타내어진 모델을 CTL 로 표현된 요구 명세를 만족하는지의 여부를 효율적으로 검사하기 위해 OBDD(Ordered Binary Decision Diagram)을 기반으로 하는 symbolic 모델 체킹 알고리즘을 사용한다. SMV 를 이용하여 대형 하드웨어나 소프트웨어 시스템 명세를 검증하는 예가 연구되고 있다.

4. CASE STUDY

4.1 기존 명세의 오류

기존의 명세에서는 초기 상태에 만일 i_Pressure ≤ 950 kPa 이고 i_Power ≥ 80%FP 이여서, Belowsp, High 스테이트

상태가 될 때 function 부분에서 S2 스테이트로 전이하지 못하는 경우가 발생하게 된다. 이런 경우가 발생하면 trip signal 을 발생시키지 못해 output 부분에서 open 하지 못하는 경우가 발생하게 되는 것이다. 따라서 본 논문에서는 어떤 경우라도 Belowsp, High 상태가 되었을 때는 , S2 스테이트로 전이할 수 있도록 수정하였다.

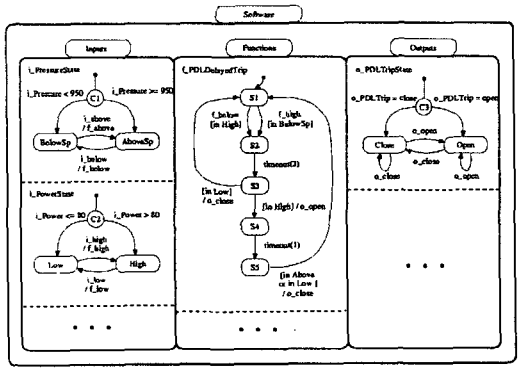

그림 5 수정 전 STATECHART 명세

4.2 차폐 시스템(Shutdown System)의 수정된 명세

이 논문에서 명세할 월성 SDS2 의 원자력 발전 제어시스템의 일부[6]이다. 원자력 발전소 제어시스템의 구조는 간략히 다음과 같은 세 개의 프레임으로 나눌 수 있다. 첫 번째 입력 부분 두 번째 입력에 따른 function 부분, 마지막으로 output 부분이다.

만일 i_Pressure ≤ 950 kPa 이고 i_Power ≥ 80%FP 이면 3 초동안 o_PDLTrip 을 열지 않고 일반적인 역할을 수행한다.

3 초가 지난 다음, i_Power ≥ 80%FP 이면, o_PDLTrip 의 값은 Open 이 되며, 그렇지않을 경우, o_PDLTrip 의 값은 Close 가 된다.

일단 o_PDLTrip 이 open 되면 1 초동안 Open 상태를 유지한다.

i_Pressure > 950kPa 또는 i_Power 가 < 80%가되면 o_PDLTrip 의 값이 Close 가 된다

위에서 살펴본 바와 같이 시스템의 구성은 두 개의 input 모듈- i_Pressure, i_Power-과 입력을 받아 출력을 내어주는 function 모듈, function 모듈의 출력을 받아 시스템의 상태를 조정하는 PDLTrip 모듈, 마지막으로 하나의 입력 상태를 받아 출력 상태의 이벤트를 내어주는 function 상태와 시간을 측정하는 모듈로 구성되어 있다. Inputs 모듈에서의 전이는 외부 이벤트에 의해 발생되고, 이 이벤트는 Functions 모듈로 broadcasting 된다. i_Pressure 의 초기 상태는 입력 변수, 즉 i_Pressure 의 통신상에 있는 predicates 를 포함하는 조건 연결자(conditional connective)에 의해 결정된다. Outputs 은 trip 매개변수가 open 인지 close 인지를 나타내는 o_PDLTrip 로 구성되어 있다. f_PDLDdelayedTrip 의 행동(behavior)은 다음과 같다. 초기 상태는 S1 은 어떠한 일도 하지 않은 상태로 대기하고 있다. 그러다가 상태 S2 로 들어가게 되면, 3 초간 대기하게 된다. 3 초가 지난 후, i_Pressure 와 i_Power 의 값을 확인해서, trip signal 을 발생시킬 것인지 아닌지를 결정하게 된다. 위의 내용을 바탕으로 원자력 발전소 제어시스템을 STATECHART 로 명세해 본 결과는 <그림-5>과 같다.

STATECHART 의 명세는 크게 네 부분으로 나누어져 있다. 첫번째 부분은 외부 입력이 들어오는 입력 부분과 두 번째 부분은 이러한 입력 이벤트에 따라 현재의 상태를 파악함으로 위험 혹은 안전을 판단하여 출력을 내 보냄으로 위험한 상황 즉 I_PRESSURE 가 950 이상 I_POWER 가 80 이하인 경우는 TRIP 을 가동시키도록 출력 부분을 특정 O_OPEN 이벤트를 출력으로 보내내는 부분이다. 그리고 안정화 상태가 되면 O_CLOSE 를 내보냄으로 위험상태에서 벗어나게 한다. 세 번째 부분은 현재 시스템의 상태를 가리킨다. 즉 안정화 상태에서 O_OPEN 을 받게 되면 시스템은 불안정화 상태이고 O_CLOSE 를 받게 되면 안정화 상태이다. 마지막 네 번째 부분은 위의 명세를 SMV 검증을 위한 명세로써 외부 입력을 내부입력으로 바꾸어 임의의 입력이 들어오는 것을 표현하기 위한 부분이다.

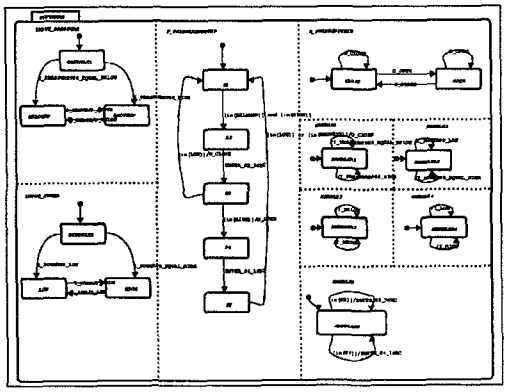

그림 6 수정 후 STATECHART 명세

4.3 차폐 시스템의 검증

본 논문에서는 검증코드를 만들기 위해 원래의 STATEMATE 의 STATECHART 의 문법에 얼마의 제약을 가했다. 특히 STATECHART 의 시간적 문법은 배제하고 이를 임의의 이벤트(ENTER_S2_3SEC, ENTER_S4_1SEC)로 바꾸었다. 또한 조건문을 배제하고 이를 특정 이벤트로 전환하여 적용시켰다. 이를 SMV 로 바꾸면 다음과 같은 코드를 만들 수 있다. <그림-5>는 SMV 로 전환된 코드의 일부이다. 이러한 설계 명세에 사용자의 요구명세가 첨가되어 함께 운용되게 되는데 사용자가 요구하는 요구 명세는 다음과 같다.

1. 어떤 경우에 있어서도 high 이고 belowsp 인 상태에서 3 초가 지나면 바로 S3 상태가 된다.

2. 어떤 경우에 있어서도 S3 상태에서 high 이면 pdltrip 이 open 상태가 된다.

3. 어떤 경우에 있어서도 S3 상태에서 low 이면 pdltrip 이 close 상태가 된다.

그리고 이를 시제논리(CTL)로 바꾸면 <그림-6>과 같이 바뀌게 되며 이를 SMV 로 운용 시키면 아래의 결과와 같이 검증할 수 있다. 위의 경우를 시제논리로 바꾸어 검증한 결과가 모두 pass 임을 볼 수 있다. 즉, 요구명세의 1 의 경우 high 상태에서 belowsp 상태에 있다면, open 의 전단계를 가리키며 이 상태에서 3 초가 지나면 반드시 pldtrip 이 open 상태의 준비단계에 있음을 볼 수 있다. 그 다음 요구명세 2 의 경우, in_S3 와 in__LOW 에 있다면 CLOSE 상태에 있음을 볼 수 있으며, 요구명세 3 의 경우, S3 상태에서 in_HIGH 의 경

우라면 pldtrip 이 OPEN 상태에 있음을 볼 수 있다. 세 경우 모두 safety 를 검증한 결과라 볼 수 있다.

```
-----------------------------------------------------
MODULE main
VAR
HANDLE1_arbiter                    : {1, 2};
HANDLE2_arbiter                    : {1, 2};
HANDLE3_arbiter                    : {1, 2};
HANDLE4_arbiter                    : {1, 2};
HANDLE5_arbiter                    : {1, 2};
I_PRESSURE950_EQUAL_BELOW_e : boolean;
I_PRESSURE950_HIGH_e        : boolean;
I_POWER80_LOW_e             : boolean;
I_POWER80_EQUAL_HIGH_e      : boolean;
I_ABOVE_e                   : boolean;
F_ABOVE_e                   : boolean;
I_BELOW_e                   : boolean;
F_BELOW_e                   : boolean;
I_HIGH_e            : boolean;
F_HIGH_e            : boolean;
...
...
-----------------------------------------------------
ASSIGN
next(I_PRESSURE950_EQUAL_BELOW_e) :=
case
     HANDLER1_TO_HANDLER1_2 : 1;
   1 : 0;
esac;
-----------------------------------------------------
ASSIGN
next(I_PRESSURE950_HIGH_e) :=
case
     HANDLER1_TO_HANDLER1_1 : 1;
   1 : 0;
esac;
-----------------------------------------------------
ASSIGN
next(I_POWER80_LOW_e)       :=
case
     HANDLER2_TO_HANDLER2_2 : 1;
   1 : 0;
esac; ----------
```

그림 7. SMV 명세

```
-- specification AG (in_HIGH & in_BELOWSP &
gen_ENTER_S2... is true
-- specification AG (in_S3 & in_LOW -> AF
in_CLOSE) is true
-- specification AG (in_S3 & in_HIGH -> AF
in_OPEN) is true

resources used:
user time: 295.25 s, system time: 0.09 s
BDD nodes allocated: 300809
Bytes allocated: 5963776
BDD nodes representing transition relation:
147308 + 9
```

그림 8. SMV 의 검증

5. 결론 및 향후 연구 방향

본 논문에서는 실시간 safety-critical 시스템인 차폐 시스템을 기존 명세의 오류를 수정하여 정형 명세 및 정형 검증을 수행하였다. 즉 설계단계에서 사용자와 설계자는 시각적 도구를 사용하여 원활하고 일관성 있는 의사소통 통해 시스템을 설계하고 이를 즉시 모델 체킹을 사용하여 시스템이 사용자의 요구를 온전히 충족시키는가를 검증하였다. 검증을 통하여 오류가 발생하였다만 이를 즉시 시각적 도구에서 수정하여 고칠 수 있게 된다. 이러한 기법은 다음 세 가지 효과를 거둘 수 있다. 첫 번째로 정형명세 언어가 가지고 있는 수학적 기호를 배재 하여 시각적이고 이해하기 쉬운 언어를 통해 사용자와 설계자 사이의 일관성 있고 원활한 의사 소통 제공할 수 있게 된다. 두 번째로 safety-critical 시스템에서 가장 중요한 요구 사항이라 할 수 있는 안정성과 신뢰성을 얻을 수 있다. 세 번째로 설계와 검증을 일관성 있게 시행할 수 있다. 즉 시각적 언어를 일관서 있게 정형 검증 언어로 바꿀 수 있게 되어 시각적 명세 후, 바로 검증을 시행할 수 있게 된다. 이것은 원활한 feedback 을 제공할 수 있다.

본 논문에서는 시간적 모델을 단지 이벤트로 처리함으로 시간의 흐름에 따른 검증은 하지는 않았다. 이에 대한 심도 있는 연구가 진행되어야 하며, 또한 전체적인 설계의 자동 역시 연구되어야 할 것이다.

6. 참고문헌

[1] Edmund M.Clarke. jr, Orna Grumberg, Doron A. Peled. Model Checking, 1999, The MIT press

[2] David Harel, STATECHART: A VISUAL FORMALISM FOR COMPLEX SYSTEMS, Science of Computer Programming 8 (1987) pp231-274

[3] Edmund M.Clarke. jr, Orna Grumberg, Doron A. Peled. Model Checking, 1999, The MIT press

[4] David Harel and Ammon Naamad, The STATEMATE Semantics of STATECHARTs, ACM Trans. Soft. Eng. Method. Oct. 1996

[5] K. L. McMillan. Symbolic model checking - an approach to the state explosion problem. PhD thesis, SCS, Carnegie Mellon University, 1992

[6] S.D. Cha and H.S. Hong, `` Specification and Analysis of Real-Time Systems in STATECHARTs," *Second International Workshop on Object-oriented Real-time Dependable Systems (WORDS) '96* Laguna Beach, California, Feb. 1996