

**2015**



제 17 권 제 1 호  
Vol. 17 No. 1



## 제 17회 한국 소프트웨어공학 학술대회 논문집

**Proceedings of the 17th Korea Conference on  
Software Engineering (KCSE 2015)**

- 일시: 2015년 1월 28일(수) ~ 1월 30일(금)
- 장소: 강원도 평창 한화리조트(휘닉스파크점)

**주최:** 한국정보과학회, 한국정보처리학회  
**주관:** 한국정보과학회 소프트웨어공학 소사이어티  
    한국정보처리학회 소프트웨어공학 연구회  
    한국전자통신연구원  
**후원:** (주)비트컴퓨터, (주)코스콤, (주)다한테크, (주)솔루션링크,  
    슈어소프트테크(주), 정보통신산업진흥원 SW 공학센터,  
    (주)ESG 솔루션, ITRC 고품질융합소프트웨어연구센터,  
    STA 테스팅컨설팅(주), TTA 소프트웨어 시험인증 연구소,  
    무인자율 및 자가적응형 소프트웨어 센터

### 자가 적응 SW 기술 1 (D3)

모델-검증을 사용한 자가-적응 소프트웨어 검증 기법 [단편논문]

이의종, 백두권(고려대)

안드로이드 플랫폼 기반의 IT 생태계 자가 적응 프레임워크의 무인삼림관리 시스템 적용사례 [단편논문]  
고동휘, 박수진(서강대)

적응 지연 시간을 고려한 선행적 자가적응 시스템 [단편논문]

정호현, 김미수, 이은석(성균관대)

자가적응형 소프트웨어의 문제인지를 위한 학습 기반 적응 시점 판단 기법 [단편논문]

설광수, 백두권(고려대)

### 아키텍처 설계 및 개선 3 (D4)

Flexible architecture for pervasive computing with wearable device

이동주, 민상윤(KAIST)

스마트폰을 이용하여 마이컴의 펌웨어를 업데이트하는 시스템 아키텍처 설계 [산업체논문]

김강현, 강성원(KAIST), 김진규(ADD)

휴먼요인의 xSHEL 모델 기반 항공교통관제 훈련을 위한 애니메이션 콘텐츠 개발방법 [산업체논문]

이경환(중앙대), 정현규(이투 OPS)

### 명세 및 검증 (E1)

1차 추론 기법을 사용한 대화형 웹서비스 API 명세 생성 및 활용

권정현, 고인영(KAIST)

반응형 시스템 개발을 위한 안전-뷰키 게임의 구현 및 비교 [우수논문]

권혁, 권기현(경기대)

차량 전장용 운영체제 대상 코드기반 모델검증 적용가능성 사례 연구 [우수단편논문]

박민규, 최윤자(경북대)

SCXML Modeling and Simulation (M&S) based on Stochastic based State Diagram [단편논문]

문소영(홍익대), 박용범(단국대), 김영철(홍익대)

# SCXML Modeling and Simulation (M&S) based on Stochastic based State Diagram

문소영\*, 박용범\*\*, 김영철\*

\* 홍익대학교 일반대학원 소프트웨어공학연구실  
세종캠퍼스  
{msy, bob}@selab.hongik.ac.kr

\*\* 단국대학교 컴퓨터과학과  
ybpark@dankook.ac.kr

**요약:** 기존의 상태 다이어그램들은 확률적 (Stochastic) 시스템 설계가 미비하다. 그래서 확률적 임베디드 소프트웨어 시스템을 개발할 때, 동적 및 확률적 상태 설계를 위해 확률 기반 상태 다이어그램[1]을 제안했다. 이를 통해 M&S(Modeling & Simulation) 기반 설계 및 검증을 수행한다. 또한 설계와 시뮬레이션을 위해 기존 아파치 커먼즈(Apache Commons) 프로젝트의 Harel 기반 SCXML 을 이용하여 검증을 시도한다. 결과적으로 개발한 JAVA 프로그램으로 상태 추출과 상태 변화 검증 시뮬레이션을 통해 모델 검증을 한다. 또한 이는 정확한 모델의 재사용 의미에서 임베디드 소프트웨어의 재사용성과 신뢰성을 높일 수 있다.

**핵심어:** Stochastic based State Diagram (SBSD), SCXML (State Chart eXtensible Markup Language), Modeling & Simulation (M&S)

## 1. 서론

확률적 임베디드 소프트웨어는 각종 전자제품이나 정보기기 등에 내장되며, 초창기 임베디드 소프트웨어는 간단한 제어 프로그램만으로 산업용 기기를 제어했다. 최근 멀티미디어 처리와 항공 및 군사용 시스템과 같은 복잡하고 거대한 기기를 통제해야 한다.

“Modeling Embedded Systems and SoCs[2]”에서 임베디드 시스템 모델의 종류는 정적 모델링과 동적 모델링으로 분류된다. 우리는 기존 논문에서 동적 모델링의 비결정, 결정, 확률론적 영역 중에서 확률론적(Stochastic) 영역 모델링을 위해 확률 기반 상태

다이어그램[1]을 제안했다.

본 논문에서는 M&S 기반 모델링 검증에 초점 둔다. W3C 와 Apache 의 SCXML 로 제안한 확률 기반 상태 다이어그램이 표현되는 것을 확인했다. 시뮬레이션은 어떤 시스템의 행동을 이해, 그 시스템 운영을 위한 여러 가지 전략들을 평가, 실제 시스템의 모델 고안과 실험을 수행하는 과정이다[3]. 제안한 확률 기반 상태 다이어그램을 설계 및 시뮬레이션이 가능하도록 Harel 상태 차트 기반 수행 환경을 제공하는 W3C 와 Apache 의 SCXML 로 변환을 했다. 그리고 개발한 JAVA 프로그램으로 상태 추출을 하여 상태 변화 검증 시뮬레이션을 통해 모델이 정확히 동작하는 것을 확인했다. 이를 통해 확률적 임베디드 소프트웨어를 개발할 때 모델링 한 후, 모델 검증 시뮬레이션으로 모델의 정확성과 정확한 모델을 재사용한다. 이런 의미에서 확률적 임베디드 소프트웨어의 재사용성과 신뢰성 또한 높일 수 있다.

## 2. State Chart XML

SCXML[5,6]은 XML 기반으로 상태 기계를 표현하는 마크업 언어이다. Harel 상태차트를 기반으로 한 상태 기계에 대한 수행 환경을 제공한다. W3C 와 Apache 에서 SCXML 명세서와 Commons SCXML 컴포넌트를 제공한다.

그림 1 과 같이 상태 다이어그램을 SCXML 형태로 표현하여 표준 형식에 맞게 파일을 생산할 수 있다. 이것은 표준 형식으로 작성할 수 있는 장점을 제공한다.

<sup>†</sup> 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업(No. 2012M3C4A7033348)과 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A2011601)

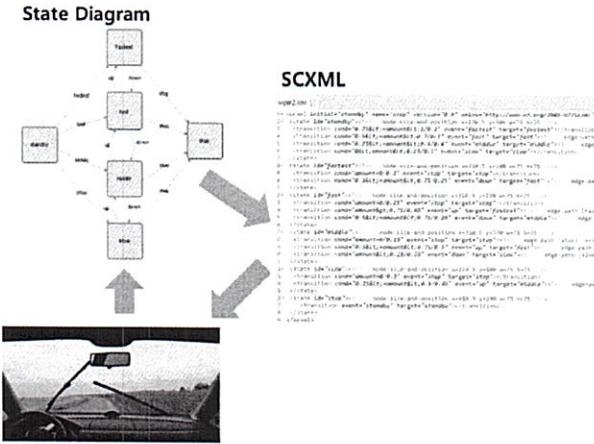


그림 1 SCXML 변환 과정

### 3. Transforming SBSD to SCXML form

일반적인 확률과정(stochastic process)은 랜덤 함수가 시간 공간 안에서 불확실성을 가지고 변하는 일련의 과정이다[4]. 기존의 Mealy, Moor, Harel 의 상태 차트는 시간 기반 확률론적 상태 설계가 불가능하여 확률 기반 상태 다이어그램을 제안했다. 확률 기반 상태 다이어그램은 ( $\Sigma, S, g, s_0, F$ )으로 구성된다[7].

$\Sigma$ : 입력 알파벳이라고 불리는 유한개의 심볼 집합

$S$ : 상태들의 유한 집합

$g$ : 상태 전이 함수,

$$S \times \Sigma \rightarrow ff(E) \text{ 인 전이 함수, } ff \text{ 는 랜덤 함수}$$

$$S = ff(\mathcal{E}, G_c) \text{ 와 } \mathcal{E} \subseteq E, E \text{ 는 이벤트 집합}$$

$s_0 : s_0 \in S$  인 시작 상태

$F : F \subseteq S$  인 최종 상태의 집합

<확률 기반 상태 다이어그램의 전이 정보>

event with parameter	guard	action	probabilistic weight value
enter(car)	[NumberOfCars > max]	rejectMessage0	0.23

그리고 시스템 개발의 구현 단계 전에 모델링과 시뮬레이션을 통해 모델 검증을 하고, 모델의 정확성을 항상 시킨다. 따라서 확률 기반 상태 다이어그램을 가지고 설계 및 시뮬레이션을 위해 모델의 실행 환경을 제공하는 SCXML로 변환하여 시뮬레이션을 수행하여 요구된 시스템으로 정확히 동작하는지에 대한 유무를 판단한다.

레인센서는 빗물의 양을 감지하여 시스템이 와이퍼를 알맞은 속도로 작동 시킨다. 레인센서는 빗물의 양을 감지하게 앞 유리 상단 중앙에 설치된다. 빗물이 유리창에 떨어지면 발산하고 있는 적외선이 빗물에 반사되어 되돌아오는 적외선을 수신 장치에서 수신하여 빗물의 양을 감지하도록 되어 있다. 빗물의

양에 따라서 와이퍼의 작동속도를 자동으로 제어하는 시스템에 장착되는 센서이다. 그림 2는 자동차 와이퍼의 동작을 상태 다이어그램으로 표현한다.

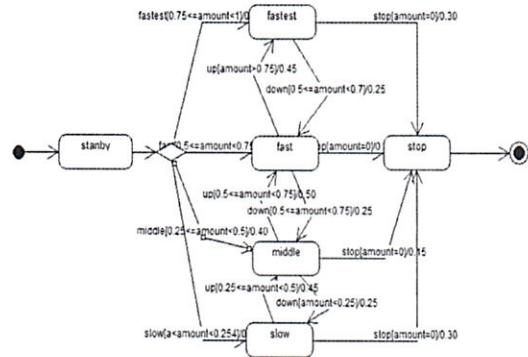


그림 2 자동차 와이퍼의 확률 기반 상태 다이어그램

```

<scxml initial="standby" name="stop" version="0.9"
  xmlns="http://www.w3.org/2005/07/scxml">
  <state id="standby">
    <transition cond="0.75<=amount<1/0.2"
      event="fastest" target="fastest">
    </transition>
    <transition cond="0.5<=amount<0.75/0.3"
      event="fast" target="fast">
    </transition>
    <transition cond="0.25<=amount<0.5/0.4"
      event="middle" target="middle">
    </transition>
    <transition cond="0&lt;amount<0.25/0.1"
      event="slow" target="slow">
    </transition>
  </state>
  ..... 종략 .....
  <state id="fast">
    <transition cond="amount=0/0.25"
      event="stop" target="stop"></transition>
    <transition cond="amount>0.75/0.45"
      event="up" target="Fastest"></transition>
    <transition cond="0.5<amount<0.75/0.25"
      event="down" target="middle"></transition>
  </state>
  ..... 종략 .....
</scxml>

```

그림 2에 대한 사례로써, 상태는 standby, fastest, fast, middle, slow, stop 으로 구성된다.

(1)  $\text{fastest(amount)[0.75}<\text{amount}<1]/\text{runFastest()}/0.2$   
standby 상태에서 fastest 상태로 트랜지션 이 일어나려면 빗물의 양은 1 을 100 으로 봤을 때, 빗물의 양은 0.75 보다 작거나 같고 1 보다 작아야 하며 확률 가중치 값은 0.2 이다.

(2)  $\text{fast(amount)[0.5}<\text{amount}<0.7]/\text{runFast()}/0.25$   
Fastest 상태에서 fast 상태로 가려면 down 트랜지션 이 실행되는데, 빗물의 양은 0.5 보다 크고 0.7 보다 작고, 확률 가중치 값은 0.25 이다.

## (3) stop(amount)[amount=0]/stop()/0.33

(1)의 상태에서 stop 상태로 트랜지션이 발생하려면 빗물의 양은 0이며 확률 가중치 값은 0.33이다.

위 텍스트 상자 안의 내용은 그림 2의 확률 기반 상태 다이어그램을 SCXML로 변환한 것이다.

<scxml>태그를 시작으로 각각의 상태를 나타내는 <state>태그가 있고, <state> 태그의 내부에는 <transition>태그가 있다. 각 <state> 태그의 id 속성은 상태의 명칭을 갖고 있다. 그리고 <transition>태그의 cond 속성에는 트랜지션의 일어날 수 있는 조건이 작성되며, event 속성에는 발생될 이벤트(메서드명)가 입력되며, target 속성에는 어느 상태로 갈 것인지 다음 실행될 상태를 입력한다.



그림 3 시뮬레이션 결과

위의 SCXML 이 작성된 파일을 JAVA 언어로 프로그래밍한 시뮬레이션 프로그램의 넣어 입력으로 하나의 시나리오에 대한 시뮬레이션을 수행하면 그림 3과 같이 상태 변화가 되는 것을 확인할 수 있다. 이 의미는 확률 기반 상태 다이어그램 설계의 상태 변화 및 검증이 가능하다.

## 4. 결 론

본 논문에서는 확률 기반 상태 다이어그램을 통해 자동차 와이퍼의 확률적인 상태를 모델링과 SCXML로 변환하여 Commons SCXML API를 활용해 개발한 시뮬레이션 프로그램으로 M&S를 수행한 결과 정확히 실행되는 것을 확인했다. 그러나 우리가 제안한 확률 기반 상태 다이어그램을 작성하기 위해서는 SCXML에 확률론적인 상태에 대한 기능 추가가 필요하고, Commons SCXML에서는 SCXML의 cond의 데이터를 읽어와서 다른 추가 작업 없이 프로그래밍

하도록 보완해야 한다.

## 참고문헌

- [1] So Young Moon, B. K. Park, Y. B. Park, R. Y. C. Kim, "Modeling and Simulation for Embedded Software System", International Conference on information Science and Applications, pp.585~586, 2014
- [2] Jantsch A., Modeling Embedded System and SOCs, Mogan Kaufmann Publishers an imprint of Elsevier Science, 2004
- [3] J. Y. Kim, Computer Simulation, ParkYoungSa, 1994
- [4] H.W. Lee, Queuing Theory, Sigma Press, 2004
- [5] <http://www.w3.org/TR/scxml/>
- [6] <http://commons.apache.org/proper/commons-scxml/>
- [7] So Young Moon, A study on modeling and simulation for embedded software system: stochastic based state diagram, M.S. Thesis, Hongik University, Seoul, Korea., 2007