

ASADAL/OBJ Hybrid: 고신뢰성 하이브리드 시스템 모델링을 위한 객체지향 프레임워크

신승엽^o 김문주 강교철
포항공과대학교 컴퓨터공학과
{horang11^o, moonzoo, kck}@postech.ac.kr

ASADAL/OBJ Hybrid: An Object-Oriented Framework for Modeling and Validating Hybrid Systems

Seungyeob Shin^o Moonzoo Kim Kyo C. Kang
Dept. of Computer Science and Engineering, POSTECH
요 약

하이브리드 시스템이란 연속적인 변화와 이산적인 변화가 함께 존재하는 시스템으로 많은 임베디드 시스템이 하이브리드 시스템으로 모델링 된다. 이러한 하이브리드 시스템을 모델링하고 타당성을 검사(validation) 할 수 있는 도구의 제안은 시스템 개발자들에게 하이브리드 시스템의 복잡한 변화로 인한 시스템의 복잡성을 간소화 시킬 수 있는 방법을 제공하고, 고품질의 제어 소프트웨어를 개발할 수 있도록 해준다. 본 논문에서 소개하는 ASADAL/OBJ Hybrid는 고신뢰성 하이브리드 시스템 모델링을 위한 도구로써 객체지향 모델링에 따라 여러 하이브리드 시스템을 객체로 모델링하며, 이들 간의 통신은 이벤트와 데이터 전송으로 이루어지도록 한다. 환경 객체의 경우 3D 형태 정보를 가지고 있어서 3D 시뮬레이션을 통해 시스템의 변화 양상을 개발자에게 직관적으로 제공하도록 하고 있다.

1. 서 론

하이브리드 시스템이란 연속적인 변화와 이산적인 변화가 함께 나타나는 시스템으로 많은 임베디드 시스템이 하이브리드 시스템으로 모델링 된다. 이러한 하이브리드 시스템의 모델링 및 타당성 검사 도구로써 수식기반으로 모델링 하는 MATLAB(참고: <http://www.mathworks.com>)이 주로 이용되어 왔으나, 시스템을 모듈화 시켜 모델링 할 수 없어, 모델 자체가 지나치게 복잡하고 이해하기가 힘들다는 단점이 있다. 이를 보완하기 위해 Shift[1]나 Charon[2]과 같은 도구들이 각각 계층적 명세와 모듈 개념을 도입하였으며, Charon은 캡슐화나 재사용 개념을 도입하여 하이브리드 시스템의 모델링을 보다 용이하게 하였다. 하지만, 실제로 이식 혹은 구현되어 이용될 제어 소프트웨어 부분과 물리 환경 부분의 모델이 혼재되어 있어 이러한 도구들을 이용하여 모델링 한 결과를 이용하기에는 어려움이 따른다. 또한, 이 도구들의 그래프를 이용한 시뮬레이션 결과의 가시화는 하이브리드 시스템의 복잡한 변화 양상을 직관적으로 보여주지 못한다.

본 논문은 ASADAL/OBJ[3]라는 실시간 임베디드 시스템의 모델링 및 타당성 검사 도구를 확장하여 하이브리드 시스템의 효과적인 모델링과 3D 시뮬레이션을 통해 시스템의 복잡한 변화 양상을 직관적으로 볼 수 있도록 한 ASADAL/OBJ Hybrid에 대한 소개를 하고 있다. 논문의 구성은 2장에서 ASADAL/OBJ에 대한 간략한 소개를 하고 있으며, 3장에서는 ASADAL/OBJ Hybrid에 대한 설명을 하고 있다. 마지막 4장에서는 결론과 함께 향후 연구 계획을 기술하고 있다.

2. ASADAL/OBJ 소개

ASADAL/OBJ(A System Analysis and Design Aid Tool/Object-oriented)는 실시간 임베디드 시스템의 모델링 및 타당성 검사 도구로써, 본 장에서는 ASADAL/OBJ에서의 환경 모델 명세 방법과 시뮬레이션에 관해 중점적으로 기술하고 있다. ASADAL/OBJ에 대한 자세한 내용은 [3]에서 다루고 있다.

2.1 환경 모델 명세

환경이란 제어 소프트웨어가 제어하는 물리적 시스템과 그 주변 환경을 의미한다. ASADAL/OBJ에서 환경 모델링은 형태를 가지는 환경 객체를 기반으로 모델링되며, 환경 객체의 명세는 행위명세와 기능명세, 형태명세를 통해 이루어진다. 이는 각각 Statecharts와 DFD, 형태 명세 언어를 이용하여 명세 한다.

환경 객체의 움직임은 선형 움직임과 회전 움직임으로 구분지어 명세하고 있다. 선형 움직임은 3D 좌표계에서 직선으로 움직이는 변화를 표현하고 있으며, 회전 움직임은 특정 좌표를 축으로 회전하는 움직임을 표현하고 있다. 각 변화는 환경 객체에 미리 정의되어 있는 `acc<x>`와 `vel<x>`, `transVal`을 이용하여 선형 움직임의 가속도와 속도, 이동거리를 나타내며, `aacc<x>`와 `avel<x>`, `rotVal`을 이용하여 회전 움직임의 각가속도와 각속도, 회전각을 나타낸다. (<x>는 x와 y, z축을 나타낸다.) 이러한 변수를 이용한 움직임의 명세는 DFD의 프로세스 명세에서 각 변수에 대응되는 값을 할당하는 형태로 명세 된다. 따라서 표현 가능한 움직임은 가속도와 상수인 움직임으로 제한된다.

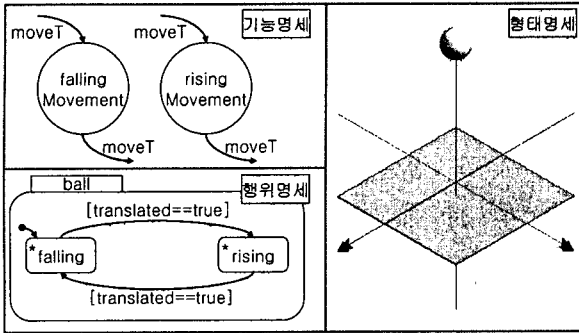


그림 1: <Bouncing Ball 예제> ASADAL/Obj 명세

그림1은 ASADAL/Obj를 이용하여 모델링한 Bouncing Ball 예제의 부분 명세로 프로세스 명세를 위해 가속도에 영향을 주는 요소로 중력 이외의 것을 고려하지 않았다. falling 상태에서는 fallingMovement 프로세스를 수행하도록 명세 되어 있으며, rising 상태에서는 risingMovement 프로세스를 수행하도록 명세 되어 있다.

2.2 시뮬레이션

ASADAL/Obj는 시뮬레이션을 위해 제어 소프트웨어 시뮬레이터와 환경 시뮬레이터가 있으며, 각 시뮬레이터는 시뮬레이션 하는 객체들의 행위명세들을 바탕으로 스텝 문맥에 따라 수행되어야 할 프로세스들을 스케줄링 한다. 즉, ASADAL/Obj에서 스텝이란 현재의 현황(configuration)에서 다음 현황을 구하는 과정을 말하며, 이에 따라 각 시뮬레이터는 현재 현황에서 스케줄링 되어야 할 프로세스들을 수행시킨다. 이때, 환경 시뮬레이터는 프로세스에 기술된 명세에 따라 환경 객체들의 형태정보를 매 스텝마다 갱신한다.

예를 들어, 그림1의 fallingMovement 프로세스에서는 공이 이동해야 할 거리와 초기 속도, 가속도가 명세 되어 있다. 시뮬레이션 중 falling 상태에 제어권이 넘겨질 때, 환경 시뮬레이터의 스케줄러에는 fallingMovement 프로세스가 등록되며, 환경 시뮬레이터는 스케줄링 되는 fallingMovement 프로세스에 명세 된 내용을 바탕으로 매 스텝에 할당된 시간 동안 움직여야 할 공의 이동거리를 계산하여 공의 위치정보를 갱신한다.

3. ASADAL/Obj Hybrid

하이브리드 시스템의 수학적 모델은 아래의 식(1)과 같이 연속적인 변화는 상미분방정식의 집합으로 모델링되며, 이산적인 변화의 원인은 각 상미분방정식의 우측에 표기된 가드 함수 g에 관한 부등식으로 모델링 된다.

$$\dot{x} = \frac{dx}{dt} = \begin{cases} f_a(t, x), & g(x) < 0 \\ f_b(t, x), & g(x) \geq 0 \end{cases} \quad (1)$$

즉, 시간 t가 0일 때 시스템의 변화가 $f_a(t, x)$ 라면, $g(x) < 0$ 가 참일 동안 시스템의 변화는 $f_a(t, x)$ 에 따

른다. 그리고 $f_a(t, x)$ 에 의해 시스템이 변화하는 중에 $g(x) < 0$ 이 거짓이 되고, $g(x) \geq 0$ 가 참이 되는 순간의 시스템 변화는 $f_a(t, x)$ 에서 $f_b(t, x)$ 로 변하게 된다.

3.1 연속적 변화 명세

연속적인 변화를 나타내는 상미분방정식은 데이터가 변화하는 양상을 표현하는 것이므로 ASADAL/Obj Hybrid에서는 이러한 상미분방정식을 DFD의 프로세스 명세에 표1의 diff()를 이용하여 명세 하고 있다.

표 1: 연속적 변화 명세 구문

diff(Var v, "Equation e", "Invariant i")	
Var	상미분방정식의 종속 변수
Equation	비선형 상미분방정식
Invariant	e에 의해 연속적인 변화를 할 때에 반드시 지켜져야만 하는 조건

diff()를 이용하여 명세 된 Var v의 연속적인 변화는 Invariant i가 참이고, diff()가 명세 되어 있는 프로세스 p가 활성화 되어 있을 동안에 Equation e에 따른 변화를 하게 되며, 프로세스 p가 비활성화 되거나 Invariant i가 거짓이 될 때에 Equation e에 따른 변화를 멈추게 된다. 그림2의 명세는 ASADAL/Obj Hybrid를 이용한 Bouncing Ball 예제의 프로세스 명세를 보이고 있다.

```

1: Process ballMovement{
2:   Input      double moveT.velZ, z;
3:   Output     double moveT.velZ, z;
4:   [true](0){
5:     diff(moveT.velZ, "-9.8+moveT.velZ*moveT.velZ", "true");
6:     diff(z, "moveT.velZ", "z>=0");
7:     put(moveT.velZ); put(z);
8:     translate(moveT);
9:   }
10:}
    
```

그림 2: <Bouncing Ball 예제> 프로세스 명세

ASADAL/Obj Hybrid는 비선형 상미분방정식으로 표현되는 변화의 명세가 가능하므로, 그림2의 5-6라인에서는 중력과 공기저항이 고려되어 공이 떨어지는 연속적인 변화가 명세 되어 있다. moveT.velZ가 공의 속도를 표현하고 있으며, z가 공의 위치를 표현하고 있다.

3.2 이산적 변화 명세

이산적인 변화는 Statecharts의 상태전이를 이용하여, 상태전이에 명세 되는 event[guard]/action의 action부분에 대수식을 기술하고 상태전이의 대상 상태에서 활성화 시키는 프로세스에 기술된 diff()에 의해 명세 된다.

그림3은 Bouncing Ball 예제의 statechart로 공의 z 위치가 0보다 작아졌을 경우에 falling 상태에서 rising 상태로의 전이를 하도록 명세 되어 있다. 이때, 상태 전이는 falling 상태에서 활성화 시킨 그림2의 프로세스에 기술되어 있는 불변식들(invariants)이 참이고 가드 [z<=0]가 참일 경우에 일어날 수 있으며, 공이 지면과

충돌함으로 인해 속도의 방향이 바뀌고 속력이 저하되는 이산적인 변화를 action 부분의 $moveT.velZ*=-0.9$ 로 명세하고 있다.

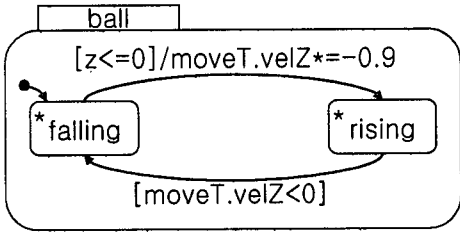


그림 3: <Bouncing Ball 예제> Statechart

3.3 동시성(Concurrency)

ASADAL/Obj Hybrid에서 객체들은 동시성을 지니는 요소로 인식되며, 하나의 객체에서는 Statecharts의 'And line'으로 시스템의 동시성을 명세 한다. 객체들 간의 통신은 이벤트와 데이터 전송을 통해 일어나며 이때, 수행되고 있는 프로세스들에 명세 되어 있는 상미분방정식이 서로 의존적인 경우가 존재하게 된다. 즉, 상미분방정식 $e1$ 의 종속변수 $v1$ 이 다른 상미분 방정식 $e2$ 의 종속변수 $v2$ 의 변화에 영향을 받는 경우가 발생한다. 이러한 경우 시뮬레이션 시에 수치 해를 구하는 알고리즘(예,[4])에 의해 먼저 구해지는 해가 다른 상미분방정식의 수치 해를 구하는 과정에 영향을 주게 되므로, 시뮬레이션 결과가 부정확해지는 문제가 발생한다. 따라서 이들 간의 동기화를 위해 여러 상미분방정식의 독립변수 t 는 글로벌 시간을 이용하여 증가하며, 이전 스텝에서 구해진 수치 해들을 현재 스텝에서 수치 해를 구할 때 이용한다.

3.4 3D 시뮬레이션

하이브리드 시스템의 복잡한 변화 양상을 가시화 하는 것은 하이브리드 시스템의 분석에 있어서 매우 큰 비중을 차지하는 요소이다. 기존 연구들의 그래프를 이용하여 시뮬레이션 결과를 가시화하는 것만으로는 하이브리드 시스템의 복잡한 변화 양상을 직관적으로 전달하지 못한다. 즉, 하이브리드 시스템에서 변화하는 요소들이 다양하게 존재할 경우, 그래프를 통한 분석만으로는 시스템의 전반적인 변화 양상을 파악하기가 힘들다. 따라서 ASADAL/Obj Hybrid는 하이브리드 시스템의 변화 양상을 보다 직관적으로 전달하기 위해 3D 시뮬레이션을 제공하고 있다. 이를 위해, 환경 객체들의 움직임을 표현하는 상미분방정식을 선형 변화와 회전 변화로 구분지어 실제계의 움직임을 묘사하도록 하고 있다. 각 변화는 2.1절에서 설명된 선형 움직임과 회전 움직임을 표현하는 변수들을 상미분방정식의 종속변수로 이용하여 환경 객체의 연속적인 변화를 명세 한다. 환경 시뮬레이터는 이러한 상미분방정식들의 해를 4차 Runge-Kutta 방법을 이용하여 구하며, 구해진 해를 바탕으로 환경 객체들의 위치정보를 매 스텝마다 갱신시킨다. 그림4는 이러한 ASADAL/Obj Hybrid를 이용하여 Bouncing Ball 예제를 시뮬레이션 한 결과를 보이고 있다.

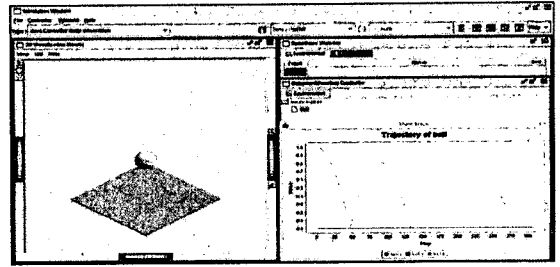


그림 4: <Bouncing Ball 예제> 시뮬레이션 결과

4. 결론 및 향후 연구 계획

본 논문에서 소개한 ASADAL/Obj Hybrid는 객체를 기반으로 하이브리드 시스템을 모델링하고 있으며, 여러 하이브리드 시스템들 간의 상호 작용을 모델링하기 위해 객체간의 통신을 이벤트와 데이터 전송을 통해 일어나도록 하고 있다. 그리고 시스템의 동시성으로 인한 데이터의 동기화를 위해 글로벌 시간을 이용하여 여러 상미분방정식의 해를 구하고 있으며, 하이브리드 시스템의 복잡한 변화 양상을 효율적으로 분석할 수 있도록 하기 위해 그래프와 더불어 3D 시뮬레이션을 함께 제공하고 있다.

앞으로 ASADAL/Obj Hybrid는 크게 다음의 2가지 이슈를 고려하여 하이브리드 시스템 개발 지원도구로서의 가치를 높일 계획이다. 우선 고정된 스텝을 이용하게 됨에 따라 상미분방정식의 치환 시점을 정확하게 판단할 수 없는 이벤트 검출 문제(예,[5])를 해결하여야 한다. 다음으로는 시뮬레이션 시에 발생하는 오차의 허용 범위를 시스템 개발자가 결정할 수 있도록 하여 보다 테크니컬하게 분석을 할 수 있도록 연구를 진행할 계획이다.

참고 문헌

- [1] A. Deshpande, A. Göllü, and L. Semenzato, "The Shift Programming Language and Run-time System for Dynamic Networks of Hybrid Automata," California PATH Technical Report, UCB-ITS-PRR-97-7, 1997.
- [2] R. Alur, R. Grosu, Y. Hur, V. Kumar, and I. Lee, "Modular Specification of Hybrid Systems in Charon," Hybrid Systems: Computation and Control, LNCS, pp. 6-19, March, 2000.
- [3] J. Lee, H. Kim, and K. Kang, "A Real World Object Modeling Method for Creating Simulation Environment of Real-time Systems," OOPSLA, pp. 93-103, October, 2000.
- [4] D. Kincaid and W. Cheney, "Numerical Analysis: Mathematics of Scientific Computing," 2nd Edition, Brooks/Cole Publishing Company, pp. 564-662, 1996.
- [5] J. Esposito and V. Kumar, "An Asynchronous Integration and Event Detection Algorithm for Simulating Multi-Agent Hybrid Systems," TOMACS ACM, Vol. 14, Issue 4, October, 2004.