한국시뮬레이션학회 '00추계학술대회 논문집 2000.11.11 아주대학교

시스템 Morphism을 이용한 이산사건 시스템의 검증

홍 기 정, 김 탁 곤

한국과학기술원 전자전산학과
시스템 모델링 시뮬레이션 연구실

E-mail : kjhong@smslab.kaist.ac.kr, tkim@ee.kaist.ac.kr

Homomorphic Verification of Hierarchical DEVS Model at Structure Level

Ki Jung Hong, and Tag Gon Kim

Systems Modeling Simulation Laboratory
Department of Electrical Engineering and Computer Science, KAIST

요 약

이산사건 시스템의 검증은 설계된 모델의 동작을 검증하는 것이 목적이다. 지금까지 이산사건 시스템 검증은 설계된 모델의 safety, liveness등을 검사하는 model check방법론과 모델의 spec과 설계된 모델을 검증하는 bisimulation의 2가지 방향으로 발전해왔다. model check방법론은 temporal logic, process algebra등의 분야에서 연구되어 왔으며 모델이 커질 경우 state explosion문제가 발생하여 검사할 수 있는 state의 개수가 제한되어 있다. bisimulation 은 CCS등에서 연구되어온 방법론으로써 모델의 spec이 정확하다는 가정아래 설계된 모델과 bisimulation한 관계를 찾음으로써 모델의 검증을 하게 된다. 시스템 morphism은 bisimulation의 다른 표현으로써 목적 시스템의 top-down 설계 시 시스템의 추상 계층에 따른 모델들 사이의 homomorphic한 관계가 있음을 뜻한다.

본 연구에서는 이산사건 시스템 형식론인 DEVS를 이용하여 top-down설계시 structure level에서 높은 추상 계층의 모델과 보다 낮은 추상 계층의 모델들 사이의 시스템 morphism의 관계가 있다는 것을 보임으로 써 검증할 수 있는 방법론을 제시한다.

1. 서론

이산사건 시스템의 검증은 설계된 모델의 동작을 검증하는 것이 목적이다. 지금까지 이산사건 시스템 검증은 설계된 모델의 safety, liveness등을 검사하는 model check방법론과 모델의 spec과 설계된 모델을 검증하는 bisimulation의 2가지 방향으로 발전해왔다. model check방법론은 temporal logic, process algebra등의 분야에서 연구되어 왔으며 모델이 커질 경우 state explosion문제가 발생하여 검사할 수 있는 state의 개수가 제한되어 있는 데 이것 또한, BDD등의 technique을 썼을 때에 best case인 경우이다[2,6]. 그리고, bisimulation 은 CCS등에서 연구되어온 방법론으로써 모델의 spec이 정확하다는 가정아래 설계된 모델과 bisimulation한 관계를 찾음으로써 모델의 검증을 하게 된다. 시스템 morphism은 bisimulation의 다른 표현으로써 목적 시스템의 top-down 설계 시 시스템의 추상 계층에 따른 모델들 사이의 homomorphic한

관계가 있음을 뜻한다. bisimulation에는 strong bisimulation과 weak bisimulation이 있는데 strong bisimulation은 state까지 같이 고려하여 모델을 검증하는 것이고 weak의 경우는 단순히 I/O level equivalence만을 검사하는 것이다[1].

본 연구에서는 이산사건 시스템 형식론인 DEVS를 이용하여 top-down설계 시 structure level에서 높은 추상 계층의 모델과 보다 낮은 추상 계층의 모델들 사이의 시스템 morphism의 관계, 즉, I/O level equivalence가 있다는 것을 보임으로 써 검증할 수 있는 방법론을 제시하고 간단한 예제로써 PLC로 컨트롤하는 조립공정라인과 시스템 morphism적용 예를 보인다.

2. 이산사건 검증 시스템의 개요

2.1 검증 시스템 소개 및 필요성

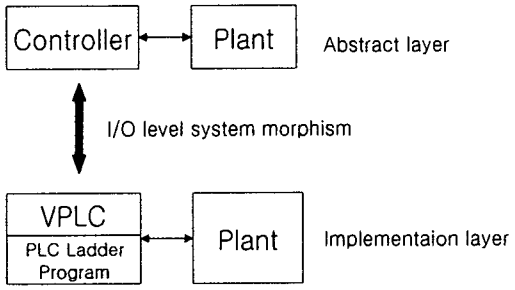

그림 1 시스템 Morphism을 이용한 예

그림 1과 같이 추상수준에서 위와 같은 플랜트의 컨트롤러를 설계하였을 때 실제 조립 공정에서 사용하기 위해서는 추상수준의 모델을 실제 사용 가능한 구현수준으로 낮추어야한다. 이렇게 같은 모델의 추상수준의 변화를 줄 때 모델간의 검증을 할 필요성이 있다. 즉, 실제 PLC (Programmable Logic Controller)에 사용되는 프로그램이 설계된 모델과 일치한다는 것을 보이기 위한 프래임웍이다. 본 논문에서는 추상수준의 컨트롤러와 플랜트는 DEVS형식론을 기반으로 한 모델들을 사용한다.

2.2 시스템 Morphism

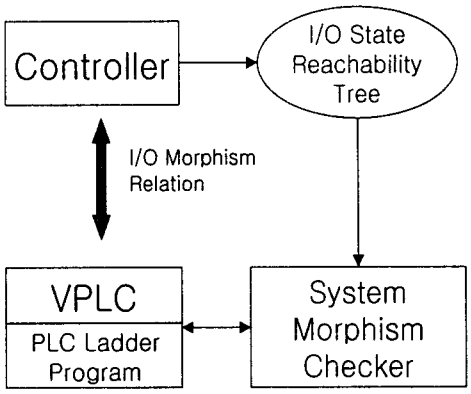

그림 2 System morphism checker

시스템 Morphism은 적용대상에 따라 다음과 같이 나뉜다. I/O만을 대상으로 하는 것, 모델들 간의 구조들도 포함하는 것, 모델 내부의 state도 포함하는 것 등으로 구분된다. 제안하는 프레임웍은 I/O수준의 시스템 Morphism을 대상으로 하며 설계된 컨트롤러 모델은 I/O State reachability tree로 변환되어 검증해야할 ladder logic을 검증하게된다.

3. DEVS 기반 검증 시스템

3.1 DEVS 형식론

DEVS형식론은 계층적 모듈라한 형태로 모델링하는 방법을 제시한다. 복잡한 이산사건 시스템은 계층적으로 분리해서 표시할 수 있는 데 더 이상 나눌 수 없는 가장 작은 개체를 DEVS형식론에서는 atomic 모델이라고 한다. atomic 모델의 수학적 정의 다음과 같다.

$AM = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$
X : 입력사건집합
S : 상태집합
Y : 출력사건집합
δ_{int} : $S \rightarrow S$: 내부상태전이함수
δ_{ext} : $Q \times X \rightarrow S$: 외부상태전이함수
λ : $S \rightarrow Y$: 출력함수
ta : $S \rightarrow Real$: 시간진진함수
$Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$
: total state of AM(e: elapsed time)

DEVS의 coupled모델은 계층적인 모델 구조를 기술한다. 이것은 새로운 모델을 구성하기 위해서 컴포넌트 모델을 coupling하여 더 작은 모델에서 더 크고 복잡한 모델을 만들 수 있도록 한다. Coupled모델의 수학적인 정의는 다음과 같다.

$$
\begin{aligned}
CM &= \langle X, Y, \{M_i\}, EIC, EOC, IC, SELECT\rangle \\
X &: \text{ 입력사건집합} \\
Y &: \text{ 출력사건집합} \\
\{M_i\} &: \text{ DEVS컴포넌트집합} \\
EIC &\subseteq X \times \bigcup_i X_i : \text{ 외부입력관계} \\
EOC &\subseteq \bigcup_i Y_i \times Y : \text{ 외부출력관계} \\
IC &\subseteq \bigcup_i Y_i \times \bigcup_i X_i : \text{ 내부입출력관계} \\
SELECT &: 2^{\{M_i\}} - \phi \rightarrow \{M_i\}
\end{aligned}
$$

DEVS 형식론에 대한 자세한 논의와 모델링은 [4,5]에서 찾을수 있다.

3.2 I/O Level 시스템 Morphism방법론

$$
\begin{aligned}
IOM &= \langle X, Y, f\rangle \\
X &= \text{입력사건집합} \\
Y &= \text{출력사건집합} \\
f &: \times_N X \rightarrow \times_N Y \\
&: \text{입출력관계함수}
\end{aligned}
$$

PLC 컨트롤러를 입출력수준의 모델로 표현된다. 위와 같은 모델과 DEVS모델 사이의 morphism관계는 다음과 같다.

$$
\begin{aligned}
IOM &= h \cdot AM \\
IOM.X &= AM.X \\
IOM.Y &= AM.Y \\
IOM.f &= AM.\lambda \cdot AM.\delta \\
AM.\delta &: \times_N X \rightarrow (\delta(s_0, x_1), \ldots, \delta(s_{n-1}, x_n)) \\
AM.\lambda &: \times_N S \rightarrow (\lambda(s_0), \ldots, \lambda(s_{n-1}))
\end{aligned}
$$

위의 수식은 DEVS 모델의 내부 상태 전이 함수와 외부 상태 전이 함수를 하나로 묶어둔 것이고 나타내어진 결과는 DEVS 모델과 I/O 수준의 모델과의 morphism 함수는 입출력 *sequence*에 의해 표시가 가능함을 알수가 있다. 이러한 morphism 함수가 존재하는 지를 알기 위해서 다음과 같은 검사 방법을 도입한다. 위의 morphism함수 h는 atomic model에 대해서만 나와 있지만 사실 coupled model은 atomic model과 같이 볼 수 있다. 그것은 아래 수식에서 증명되는데 morphism함수 g는 coupled model이 atomic model가 같다는 것을 보여준다.

$$
\begin{aligned}
AM &= g \cdot CM \\
AM.X &= CM.X \\
AM.Y &= CM.Y \\
AM.S &= \times_i CM.\{M_i\}.S \\
AM.\delta_{\text{int}} &= M_j.\delta_{ext}, (M_i.Y, M_j.X) \in IC \\
&= M_i.\delta_{\text{int}} \\
AM.\delta_{ext} &= M_i.\delta_{ext}, (X, M_i.X) \in EIC \\
AM.\lambda &= M_i.\lambda, (M_i.Y, Y) \in EOC \\
AM.ta &= MIN(M_i.ta)
\end{aligned}
$$

DEVS 모델의 State reachability tree를 구성한다. State reachability tree를 이용하여 입출력 사건의 sequence를 생성하여 IO수준의 모델이 생성된 입출력 사건의 sequence를 만족하는 지를 검사한다.

4. 예제: 조립공정 제어 프로세스

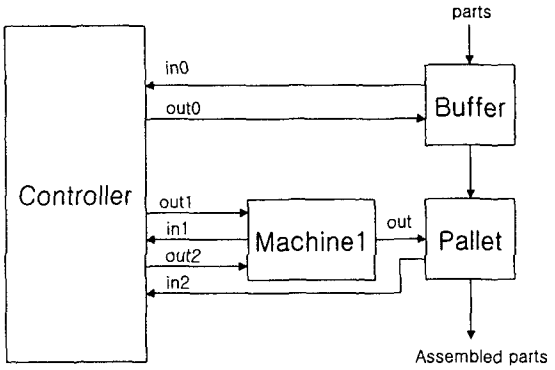

그림 3 조립공정라인 모델

그림 3과 같은 조립공정 라인이 있을 때 부품이 buffer에 들어 왔다가 machine 1이 비면 pallet에 부품을 싣고서 machine 1으로 부품을 공급하고 machine1은 부품이 자신에게 공급되면 부품을 미리 프로그램 된 것에 의해 가공하여 부품을 pallet으로 옮기고, 다시 조립된 부품을 pallet을 통해 다음 공정으로 가는 프로세스다. 사실 기계는 이보다 훨씬 많을 수 있지만 설명을 간략하게 하기 위해서 기계의 숫자를 하나로 정

했다.

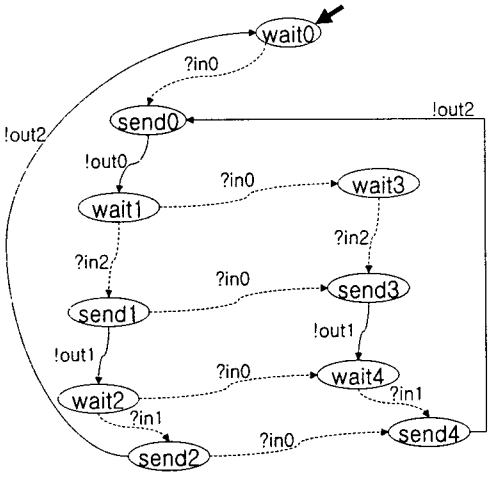

그림 4 컨트롤러 DEVS 모델

그림 3의 컨트롤러는 그림 4와 같이 설계를 하였다고 할 때 설계된 모델을 실제 사용하기 위한 PLC ladder program으로 다음과 같이 표현할 수 있다.

```
LD X0020
PLS M0000
LD M0000
SET M0010 /* set Buffer State */
LD X0021 /* from Machine 1 Done Signal */
PLS M0001
LD M0001
SET M0021 /* set Machine 1 State */
LD X0022
PLS M0002
LD M0002
SET M0011 /* set Pallet 1 State */
SET M0031 /* set Pallet 1 mreq */
LD M0010 /* If Buffer is full */
ANI M0011 /* If Pallet is Empty */
OUT Y0030
RST M0010
LD M0031 /* If Pallet 1 is mreq operation */
OUT Y0031
RST M0031
LD M0021 /* Machine 1 is done operation */
OUT Y0032
RST M0011 /* clear Pallet 1 state */
RST M0021 /* clear Machine 1 state */
END
```

위의 ladder program은 GoldSec M2N모델을 기반으로 작성된 것이다[3]. 이렇게 만들어진 ladder program이 DEVS 모델로 설계된 것과 동일한 것인지 검증하는 것이 I/O 수준의 morphism함수가 존재하는 지를 보는 것과 같은 의미를 가지게 된다. 그래서 다음과 같이 I/O수준의 morphism의 존재 유무를 검사하게 된다.

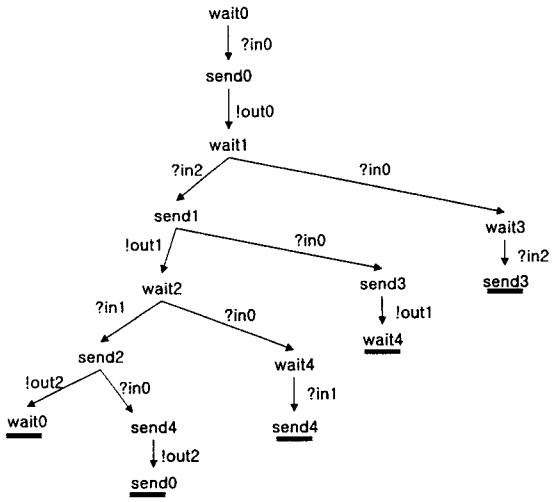

그림 5 컨트롤러 state reachability tree

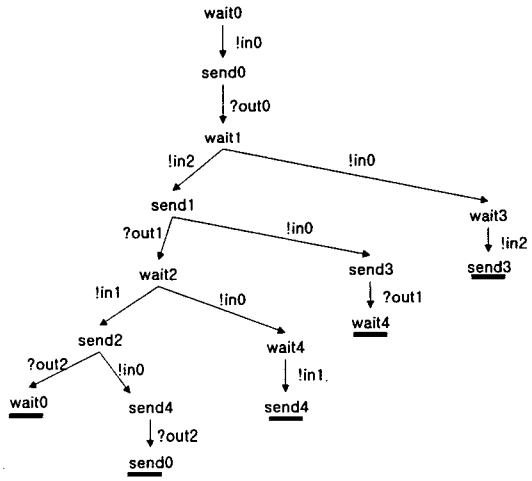

그림 6 inverse reachability tree

DEVS 컨트롤러 모델에서 state reachability tree를 추출하면 그림 5와 같이 구성된다. 구성된 모델을 실제 morphism 검사에 사용하려면 그림 6과 같이 변환된다 그림 6은 입출력을 inverse로 바꾸도록 되어 있는 데 PLC에 직접 입출력 sequence를 생성하려면 입출력이 inverse여야 한다. inverse tree로부터

입출력 sequence들이 추출되고 추출된 sequence를 따라 검사한다. 지금 예를 든 모델의 경우는 모든 state에서 다른 state로의 reachability가 있기 때문에 wait0에서부터 순서대로 모든 state로 전이하면서 입출력 사건을 PLC에 입력하고 출력을 비교하면 된다.

5. 결론

DEVS 모델과 I/O 수준의 모델사이의 morphism 관계를 조사하기 위한 수학적 기반의 방법론을 제안하였으며 제안된 방법론을 이용하여 응용 예로써 PLC controller logic을 검사하는 방법론을 제안하였다.

앞으로 할 일은 시스템 morphism을 좀더 복잡한 모델에 적용하는 것인데 이것은 단순한 I/O 수준의 모델에 대해서 만이 아니라 동일한 DEVS 모델에 관해서도 morphism관계를 이용하여 볼 수 있다. 응용분야는 interface synthesis, controller design, protocol conversion등이 있다. 그리고 추가로 고려해야할 것은 시간을 포함하는 모델인데 시간이 포함 될 경우 morphism을 이용한 검증에서 complexity가 증가한다. 시스템 morphism을 자동으로 검사할 수 있는 tool을 개발하여야 한다.

참고 문헌

[1] ROBIN MILNER, *Communication and Concurrency* , PRENTICE HALL ,pp84-128, 1989

[2] O. Kupferman , Moshe Y. Vardi, P. Wolper , *An Automata-Theoretic Approach to Branching-Time Model Checking,* CAV , 1994

[3] *MnNCPU USER'S Manual* , LG산전

[4] B.P Zeigler, *Multifacetted Modeling and Discrete Event Simulation,* Orlando, FL, Academic PRess, 1984

[5] Tag Gon Kim, *DEVSim++ User's Manual,* SMS Lab KAIST, 1994 http://sim.kaist.ac.kr/

[6] B. Yang, R. Bryant, and etc, A Performance Study of BDD Based Model Checking, *Proc of Formal Methods in Computer-Aided Design* , Palo Alto, 1998