1999년도 대한전기학회 하계학술대회 논문집 1999. 7.19-21

시간논리 구조와 Petri Net의 합성방법을 사용한 이산사건 시스템의 모델링

김진권*, 모영승, 류영국, 황형수
원광대학교 공과대학 제어계측공학과

A Modeling of Discrete Event System Using Temporal Logic Framework and Petri Net

Jin Kwon Kim*, Young Seung Mo, Young Guk Ryu, Hyung Soo Hwang
Dept. of Control & Instrumentation Engineering, Wonkwang Univ.

Abstract - In this paper, modeling and analysis of discrete event systems by temporal logic frame works and petri net is considered. The reachabil ity tree of the petri net can be used to solve the safeness, boundedness, conservation and coverabi lity problems of discrete event systems. But the reachability tree of the petri net do not solve rea chability and liveness problems in general. We proposed a method that synthesised the petri net and the temporal logic frameworks. This method slove some problems of petri net by logical representation of temporal logic frameworks.

1. 서 론

연속변수 시스템에는 상미분, 편미분 그리고 차분 방정식 등에 의한 안정도 이론, 최적제어 이론 등과 같은 많은 제어이론이 연구되어져 있다. 그러나 공정제어, 교통시스템, Flexible Manufacturing System, 통신등의 여러 응용분야에서 발견되는 이산사건 시스템의 모델링 및 제어에 관한 연구가 활발하게 진행되고 있으나, 아직도 포괄적이고 융통성 있는 제어이론이 연구되지 않았다. 현재 이산사건 시스템의 모델링 및 제어에 대한 연구는 Automata and Formal Language[5], Petri Net[1,2], Temporal Logic Framework[4,6,7]등을 이용하여 많은 연구가 이루어지고 있다. 이산 사건시스템을 모델링 및 해석하기 위해 1962년 Petri net 이론이 제시된 이래 Petri net는 여러 분야에서 응용되고 있다. 이산사건 시스템은 시스템 구성요소에서 발생되는 불연속적인 변화에 의해 상태가 변화하며, 이산사건의 특징으로서는 사건의 발생이 비동기, 병렬적이며, 사건의 발생이 다른 사건의 발생을 유도하는데 있다. 이러한 이산사건시스템을 분석하고 제어하기 위해 Petri Net는 시스템을 도식적 계층적으로 모델링 할 수 있고 사건 발생의 동시성, 분산성, 병렬성 특징을 간단하게 표현할 수 있다. Petri Net로 시스템을 모델링하고 해석 할 때 도달성그래프의 ω는 도달성 그래프를 간결하고 시스템의 해석을 편리하게 하나, 각각의 Place 토큰 표현에 문제가 발생한다. 이 문제를 해결하기 위하여 본 논문에서는 이산사건 시스템의 다른 해석 방법인 시간논리구조의 논리적 표현 방법을 Petri Net에 합성하는 방법을 제시 하였다.

2. 본 론

2.1 이산사건시스템(DES)

많은 대규모의 시스템들은 이산사건 구조(DEF : Discrete Event Framework)로 이루어져 있다. 이러한 시스템들은 이산적인 상태와 이 상태들 사이에서 사건의 천이관계를 나타냄으로써 표현할 수 있다. 이러한 시스템으로 생산 시스템, 교통 시스템, 일괄처리, 통신 시스템 등을 예로 들수 있다. 이런 시스템을 이산사건 시스템(DES : Discrete Event Systems)이라고도 부른다. 또한 이러한 시스템들은 시스템 궤적이 연속적인 연속변수 시스템(CVS : Continuous Variable Systems)처럼 상미분 방정식이나 편미분 방정식에 의해 처리되는 기존의 제어 및 시스템이론으로 취급할 수 없다.

이산사건시스템은 이산시간에 사건이 발생하고 그 사건에 의해 동작하며, 연속변수시스템은 일반적으로 연속적인 입력이나 클럭(clock)에 의해 동작한다. 이산사건시스템에서 상태는 송신을 기다리는 통신망에서 각 마디(node)에 대한 메시지의 수와 같이 물리적인 상태를 말하는 반면, 연속변수시스템에서의 상태는 추상적 의미에서 수학적 상태를 말한다.

2.2 시간논리구조(TLF)

시간논리(Temporal Logic)는 물리적 또는 프로그램에서의 상태가 시간에 따라 변하는 궤적을 수식적으로 표현하려는 방법으로 제안되었다. 시간논리구조는 시간개념을 포함한 술어논리의 확장된 형태이며, 시간 순차열에 따른 추론 지향적인 논리이다. 이 시간논리는 주로 Software 증명에 이용되었으며 최근에는 이산사건시스템의 제어문제에 적용되었다.

2.2.1 시간논리구조의 기호와 표현

시간논리구조는 종래의 Boolean 연결자인 ¬(not), ∧(and), ∨(or), →(implies), ↔(if and only if)를 포함하며 시간의 변화와 시간의 양을 표현하기 위한 시제연산자 □(henceforth), ◇(eventually), ○(next), U(until), P(proceed) 와 같은 연결자와 연산자를 이용하여 시간과 시제관계에 관한 추론이 가능한 구조이다. 이들 연산자들의 정의는 다음과 같다[8].

- 부울 논리 연산자
  - (1) $(\neg\varphi)$: φ는 참이 아니다.
  - (2) $(\varphi\wedge\phi)$: φ가 참이고 ϕ도 참이다.
  - (3) $(\varphi\vee\phi)$: φ가 참이거나 ϕ가 참이다.
  - (4) $(\varphi\rightarrow\phi)$: φ가 참이면 ϕ가 참이다.
  - (5) $(\varphi\leftrightarrow\phi)$: φ가 참이면 ϕ가 참이고 ϕ가 참이면 φ가 참이다.
- Temporal 연산자
  - (1) $(\Box\varphi)$: 지금부터 φ는 참이다.
  - (2) $(\Diamond\varphi)$: 미래에 φ가 참이 되는 점이있다.
  - (3) $(\bigcirc\varphi)$: 다음 번에 φ는 참이 된다.
  - (4) $(\varphi U\phi)$: 현재 φ가 참이고, 미래에 ϕ가 참이 되는 점이 존재한다면, 그때까지 φ는 참일 것이다.

(5) $(\varphi P \psi)$: ψ가 참이기 전에 φ는 참이어야 한다.

2.3 Petri Net

Petri Net은 상호 작용하는 동시발생 구성요소를 갖는 이산사건 시스템을 모델링하고 설계할 수 있는 그래프 이론적인 그리고 시각적인 도구이다. Petri Net는 동시적, 비동기적, 분산적, 병렬적, 비확정적, 그리고 확률적인 현상을 정보처리, 작업공정 시스템등을 묘사하고 분석, 연구하는 데에 유익한 도구이다. 그래픽 도구로서, Petri Net은 flow chart 나 block diagram 또는 network 과 유사한 시각적인 정보매체로 사용될 수 있다.

Petri Net은 시스템의 관계를 시각적으로 보여주어 시스템의 모델링을 쉽게 만든다. 시스템의 각 부분에 관한 정보를 가지고 전체와 부분에 대하여 상태들과 사건의 천이를 이용하여 하향식(top-down)과 상향식(bottom-up)설계를 할 수 있다[1,2].

2.3.1 Petri Net의 기호와 표현

일반적으로 Petri Net은 5개의 원소로 구성되어 있다[3]. 여기서,

$$N = \langle P, T, I, O, M_0 \rangle$$

$P = \{P_1, P_2, \cdots\cdots, P_m\}$ 는 유한한 플레이스의 집합이며,

$T = \{t_1, t_2, \cdots\cdots, t_n\}$ 는 유한한 트랜지션의 집합이다.

(1) P (Place) : 조건, 자원의 사용 가능성 또는 공정상태, 원(Circle)으로 표기한다.

(2) T (Transition) : 사건(event) 또는 공정의 시작과 끝, 막대(Bar)로 표기한다.

(3) I (Input Function) : Place로부터 Transition로의 방향성 화살표(Arc/Arrow).

(4) O (Output Function) : Transition으로부터 Place로의 방향성 화살표.

(5) M (Marking) : 각 플레이스에 있는 토큰의 갯수 (M_0 : 초기마킹)

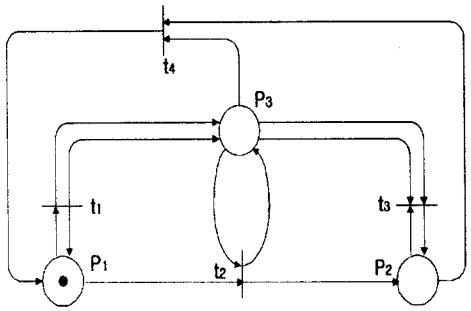
그리고, Place내부에 있는 Marker들의 분포로써 Petri Net의 상태를 표현하며, Token이라고 한다. Token들은 집합이론의 일종인 Bag이론에 의한다. 임의의 Place p_i 의 Token의 수는 $M(p_i)$ 로 표기한다.

2.4 TLF 와 Petri Net의 합성

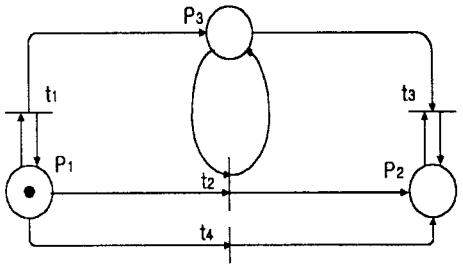
먼저, 시간논리 구조에서는 이산사건 시스템을 기존의 논리식과 시제논리를 도입하여 효과적으로 표현이 가능하지만, 수학적인 표현이므로 시각적 표현효과를 얻기에는 어렵다. 그러나 Petri Net은 시각적인 표현이 매우 뛰어나 이산사건 시스템의 특성을 잘 나타내지만, 해석방법에 있어서 Reachability Tree와 Matrix Equation을 사용하는데 서로 다른 이산사건 시스템을 모델링한 Petri Net의 Reachability Tree가 동일한 Reachability Tree로 표현되어지는 단점과 Matrix형식으로 표현됨으로 해서 Firing vector의 순서적인 정보의 부제, Self loop를 반영할 수 없으며, 다시 원상태의 Petri Net으로 구현하기 힘들다는 단점을 가지고 있다.

이런 단점들을 상호 보완하기 위한 모델링방법을 본 논문에서는 아래와 같은 동일한 Reachability Tree를 가지는 두 개의 공정을 예로 들어 설명한다[1].

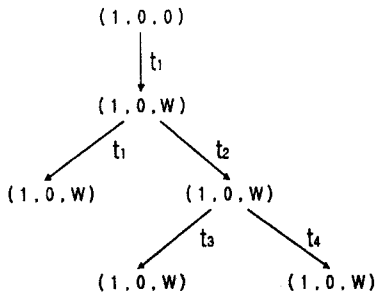
〈그림 1,2〉의 Reachability Tree는 〈그림 3〉과 같은 동일한 형태를 가진다. 이때, 〈그림 3〉에서 사용된 기호 ω는 상태변화에 따른 각 Place의 내부 Token수의 변화를 나타내는 것이다. 이 기호를 사용함으로 해서 Reachability Tree가 간결하게 표현되는 장점이 있지만, 개별적인 수치들이 제거되기 때문에 전체적으로 다른 두 가지의 공정이 같은 Reachability Tree로 표현되어지는 단점이 발생한다.



〈 그림 1 〉



〈 그림 2 〉



〈 그림 3 〉

이제, TLF로 〈그림 1,2〉의 두 가지공정을 모델링을 하여 보자.

2.4.1 Temporal Logic을 이용한 모델링

TLF를 이용하여 모델링을 하기 전에 필요한 입력과 출력함수들을 다음과 같이 정의하였다.

▶〈 정 의 〉

- STA(p_i, x) : place p_i의 token 수가 x개이다
- IN(p_i, t_j) : transition t_j에서 place p_i로의 입력함수이다.
- OUT(p_i, t_j) : place p_i에서 transition t_j로의 출력함수이다.
- FIX(p_i, x) : place p_i의 token 수가 변화가 없다.
- FIRE(t_j) : transition t_j가 발화(Fire)한다.
- ADDk(x_i) : place p_i의 token 수가 x개에서 k만큼 증가한다.

• MINk(x_i) : place p_i의 token 수가 x개에서 k만큼 감소한다.

이때, $p_i \in P$ ($i = 1, \cdots, m$), $t_j \in T$ ($j = 1, \cdots, n$)이고 . $m, n, k \in N(>0)$ 이다.

▶〈그림1〉의 TLF

(1) 발화규칙

$\Box[STA(p_1,1) \rightarrow \bigcirc FIRE(t_1)]$

$\Box[\{STA(p_1,1) \wedge STA(p_3,1)\} \rightarrow \bigcirc FIRE(t_2)]$

$\Box[\{STA(p_2,1) \wedge STA(p_3,2)\} \rightarrow \bigcirc FIRE(t_3)]$

$\Box[\{STA(p_2,1) \wedge STA(p_3,1)\} \rightarrow \bigcirc FIRE(t_4)]$

(2) 상태변화

$\Box[FIRE(t_1) \rightarrow \bigcirc\{FIX(p_1,x_1) \wedge (IN(p_3,t_1) \wedge ADD2(x_3))\}]$

$\Box[FIRE(t_2) \rightarrow \bigcirc\{(OUT(p_1,t_2) \wedge MIN1(x_1)) \wedge FIX(p_3,x_3) \wedge (IN(p_2,t_2) \wedge ADD1(x_2))\}]$

$\Box[FIRE(t_3) \rightarrow \bigcirc\{FIX(p_2,x_2) \wedge (OUT(p_3,t_3) \wedge MIN2(x_3))\}]$

$\Box[FIRE(t_4) \rightarrow \bigcirc\{(IN(p_1,t_4) \wedge ADD1(x_1)) \wedge (OUT(p_2,x_4) \wedge MIN1(x_2)) \wedge (OUT(p_3,t_4) \wedge MIN1(x_3))\}]$

▶〈그림2〉의 TLF

(1) 발화규칙

$\Box[STA(p_1,1) \rightarrow \bigcirc FIRE(t_1)]$

$\Box[\{STA(p_1,1) \wedge STA(p_3,1)\} \rightarrow \bigcirc FIRE(t_2)]$

$\Box[\{STA(p_1,1) \wedge STA(p_3,1)\} \rightarrow \bigcirc FIRE(t_2)]$

$\Box[STA(p_2,1) \rightarrow \bigcirc FIRE(t_4)]$

(2) 상태변화

$\Box[FIRE(t_1) \rightarrow \bigcirc\{FIX(p_1,x_1) \wedge (IN(p_3,t_1) \wedge ADD1(x_3))\}]$

$\Box[FIRE(t_2) \rightarrow \bigcirc\{(OUT(p_1,t_2) \wedge MIN1(x_1)) \wedge (IN(p_2,t_2) \wedge ADD1(x_2)) \wedge FIX(p_3,x_3)\}]$

$\Box[FIRE(t_3) \rightarrow \bigcirc\{FIX(p_2,x_2) \wedge (OUT(p_3,t_3) \wedge MIN1(x_3))\}]$

$\Box[FIRE(t_4) \rightarrow \bigcirc\{(IN(p_1,t_4) \wedge ADD1(x_1)) \wedge (OUT(p_2,x_4) \wedge MIN1(x_2))\}]$

위와 같이 TLF로 모델링된 것들로부터 place와 transition 사이의 token들의 입력과 출력상태 변화가 다른것을 발견할 수 있다. 이때, 변화가 다른 부분들을 그대로 Reachability Tree에 적용을 시키는 새로운 표현방법을 제시할 수 있다.

다음과 같이 위에서 제시한 Temporal Logic과 Petri Net의 합성방법을 이용하여 〈그림 1,2〉 공정의 Reachability Tree를 표현하였다.

〈그림 4,5〉에 두 개의 공정이 합성방법에 의한 다른형태의 Reachability Tree로 표현이 가능함을 알 수 있다

3. 결 론

본 논문에서는 이산사건시스템제어를 위해 사용되어지는 Petri Net의 단점을 Temporal Logic Framework을 이용하여 상호 보완하였다. TLF의 수학적 논리형식과 Petri Net의 해석방법인 Reachability Tree를 합성하여 제시한 모델링 방법은 Reachability Tree로서 이산사건시스템을 해석할 때 시스템실행의 상태를 표시하는 Token의 간결한 표현을 위해 이용되는 ω로 인한 문제점을 효과적으로 해결하였다. 앞으로, 본 논문에서 시도된 합성방법을 이용하여 Petri Net의 Matrix Equation 해석법에서 발생되어지는 문제점과 좀더 복잡한 공정에서 발생되어지는 Mutual Exclusion에서의 효과적인 모델링 방법을 연구하려고 한다.

[참 고 문 헌]

[1]J. L. Peterson, "Petri Net Theory and the Modeling of Systems", Prentice Hall, 1981

[2]T. Murata, "Petri Nets : Properties, analysis and application", Proc. IEEE, vol.77, no.4, pp.541-579, 1989

[3]V. Varadharajan, "A Petri Net Model for System Design and Refinement", J. Systems Software, 1991

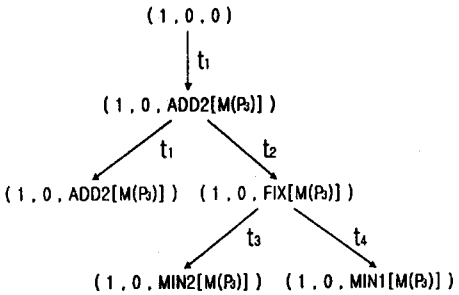
[4]J. Lin, D. Ionescu, "A generalized temporal logic approach for control problems of a class of nondeterministic discrete event systems", Proc. 29th IEEE Conf. D&C, pp.3440-3445, 1990

[5]P.J.Ramadage, W.M.Wonham, "Supervisory control of a class of discrete event process",SIAM J. Contr. Optimiz., vol.25, no.1, pp.206-230,1987.

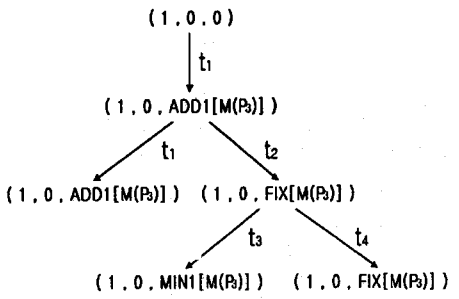
[6]Z.Manna, A.Pnueli, "Verification of concurrent programes : A Temporal proof system", Foundations of Computer Science IV, Mathematical Centre Tracts 159, Mathematish Centrum, Amsterdam, pp. 163-255,1983.

[7]J.G.Thistle, W.M.Woham, "Control problems in a temporal logic frame work", Int.J. Control, vol.44, pp.943-976,1986.

[8]Ostroff,J.S., "Temporal logic for Real Time Systems", Press Limited and Wiley, New York, 1989.



〈 그림 4 〉



〈 그림 5 〉