'88 韓國自動制御學術會議論文集 1988. 10. 21~22

# 비연속 시스템 제어를 위한 확장된 Safe Petri Net 모델과 실시간제어를 위한 Scanning Algorithm의 개발

황 창 선 , °서 정 일 , 이 재 만
부산대학교 공과대학 전기공학과

# Development of Extended Safe Petri Net Model for Discrete System Control and Scanning Algorithm for Real Time Control

Chang-Sun Hwang , Jeong-Il Seo , Jae-Man Lee
Dept. of Electrical Eng., Pusan National University

Abstract

Recently, in sequence control systems, high flexibility and maintenance of control software are required. This is because product life cycles become shorter and control specification must be changed frequently. The authors extend the concept of Safe Petri Net to develop the design and analysis tool for sequence control systems taking the safeness and notation of input/output functions into consideration. Extended Safe Petri Net (S-Net) is proposed as such a new graph model and real time scanning algorithm based on S-Net is developed.

## I. 서 론

최근에 이르러 공장자동화 (FA; Factory Automation) 분야에서 고도의 유연성과 유지성을 가지는 제어소프트웨어의 개발이 점차로 요구되어지고 있다. 이는 생산제품의 Life Cycle 이 짧고 제어사양이 자주 변동하기 때문이다. 특히, 상호 호완성이 없는 PLC, Robot Controller, NC machine 등의 제어언어는 서로 상이할 뿐만 아니라, 현대의 공장자동화 경향은 이러한 서로 다른 수준의 도구를 하나로 결합하여 이를 중앙컴퓨터를 이용해 총괄적으로 제어하고, 감시하려는 것이기 때문에 이를 만족시키는 새로운 소프트웨어 의 개발방식이 요구된다. 또한 시스템 모델링에 있어서, 공장자동화는 비동기성 병렬성을 가지므로 이를 그래프 이론적인 관점에서 접근하려는 시도가 있어왔다. 이를 위해 Petri Net 이론이 도입되어 그래프적인 해석이 가능해졌으며 다시 제어이론의 관점에서 개발되어 왔다. 이중 대표적인 것은 GRAFCET[3], C-NET[2], MFG[1] 등이 있다.

본 연구에서는 시퀀스제어 시스템이 Event Condition 시스템인 것에 착안하여, 시퀀스제어에 관한 설계, 해석을 위한 확장된 Safe Petri Net 모델을 제안하고, 이 모델에 의한 계층적 접근법에 의해 시퀀스 제어 시스템 설계법의 이론화를 기하고, 실시간 제어를 하기위한 Scanning Algorithm을 개발하고자 한다.

## II. S-Net의 구성및 성질

### A. Safe Petri Net 모델 [6,7]

Petri Net은 비동기, 병렬시스템의 거동을 묘사하고 해석하기위한 간편하고도 강력한 도구이다. 이 그래프모델은 Place 와 Transition 의 두가지 node를 가지며, Petri Net의 실행은 token에 의한다. 한개의 Place 에 많아야 한개의 token을 가지는 것을 Safe하다고 한다. Safe Petri Net 모델은

PN = ( P , T , I , O , M ) (1)

의 5개의 tuple을 가지는 bipartite 그래프의 일종이며, P는 Place의 집합, T는 Transition 의 집합이고, I : T → P , O : T → P, M : P → N (N=0,1) 의 함수이다. Transition T는 marking M에 대해 M(Pi)=1 이고, M(Pj)=0 일 때, enable이 된다. 단, Pi∈I(t) 이고 Pj∈O(t)이다. 그러나, 시퀀스제어에 적용하기 위해서는 Safe Petri Net 모델이 단지 2진상태 밖에 나타내지 못하고, 실제의 기계들은 여러 상태를 가지는 것이 보통이기 때문에, Safe Petri Net 로 모델화하면 많은 양의 Place와 Transition이 필요하다.

### B. C-Net 모델 [2]

C-Net는 Safe Petri Net의 문제점을 개선하기위해

도입된 그래프 모델이다.

CN = ( P,T,I,O,$\delta$ ,$\phi$ ,$\eta$,U,V,M ) (2)

이 모델은 C-Net 그래프의 구성요소, 즉, Receive Box, Source Box, Counter Gate 등의 많은 확장이 있었으며, Selective Scanning Algorithm을 이용한 실시간 제어의 한 방법을 제시하고 있다. 그러나, Petri Net의 무리한 확장으로 인해 원래의 Safe Petri Net의 성질을 벗어나는 수가 있어서 개선이 필요하다.

C. Mark Flow Graph 모델 [5]

Mark Flow Graph는 시퀀스 제어에 적합하게 Safe Petri Net 모델을 개량한 그래프이다.

MFG = ( B, T, A, Gi, Ge , S ) (3)

이 모델은 6개의 tuple을 가지며, 시퀀스 제어계가 Event-Condition 시스템인 것에 착안하여, Activity Box와 Distributing Box로 Place를 세분화시키고, Permissive Arc 와 Inhibit Arc를 도입하였다. 또한, Capacity Box를 도입하여 단순화 하였으나, 복잡한 사양을 가지는 제어시스템에서는 기능의 확장이 요구된다 [2].

D. 확장된 Safe Petri Net 모델

앞 절에서 제시한 여러 모델들의 문제점을 개선하기 위한 새로운 모델로서 S-Net를 제시한다.

S-Net = ( B, T, A, Gp, Gi, $\mu$ ) (4)

B : Box 의 집합

T : Transition 의 집합

A : Arc 의 집합

Gp: 허가 Arc 의 집합

Gi: 금지 Arc 의 집합

$\mu$ : Marking의 집합

이 모델은 6 개의 tuple 을 가지며, 원래의 Safe 한 Petri Net 과의 해석능력을 그대로 유지하면서 각 tuple 들이 세분화되고 확장된다. S-Net의 각 요소는 그림 1 과 같다.

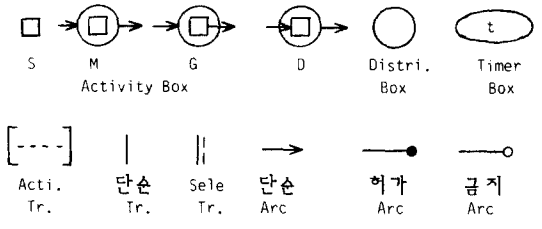



그림 1. S-Net의 요소.

1. Box 의 세분화

(1) Activity Box

이 Box에 token이 유입되면, 실제의 외부기기와 연결된다. 즉, 선택된 외부기기에 정해진 코드를 전송하여 Return 코드를 다시 받을 때 까지 wait동작을 행한다. 또한, 여러 Box를 조합하여 용량성 Box를 정의하여, Safe한 성질은 그대로 유지하면서 용량이 있는 기계, 버퍼등을 모델화 할 수 있게 하였다.

(2) Distributing Box

Activity Box 사이에 존재하는 Buffer 를 나타내며, 용량성을 가질 수 있다. 또한, 입력 수와 출력 수는 서로 동일하며, 연이어 Activity 가 발생하는 경우에는 생략할 수 있다.

(3) Timer Box

Token 이 유입된 후, 주어진 t 시간이 경과해야 출력측의 Transition이 enable 되는 시간지연 Box이다.

2. Transition 의 세분화

Transition은 Event-Condition 시스템의 Event에 해당하며, 원래의 Petri Net의 정의와 같이 두개 이상의 Transition이 enable되어도 오직 하나만이 먼저 Firing 해야하고, 동시에 Firing 할 수 없는 연속성(Simultaneous Property)를 가진다. 또,Transition은 Primitive Event를 나타내므로 Firing하는데 무한히 짧은 시간이 소모되는 것으로 가정한다.

(1) Activity Transition

두개의 transition (|,|)로서 표시하며, 두 Transition 사이에는 여러 condition과 event를 포함할 수있다.

(2) 단순 Transition

두개의 Activity Transition 사이에 존재하여, Primitive Event 를 나타낸다.

(3) Selective Transition

입력측에 있는 Activity Box의 Return 코드에 의해 조건분기를 행한다. 각 sub-transition은 조건분기를 위한 분기코드를 데이타로 가지고 있다.

3. Arc 의 세분화

Arc 는 P → T , T → P로 mapping 하는 함수로써, 방향성을 지닌다.

(1) 단순 Arc

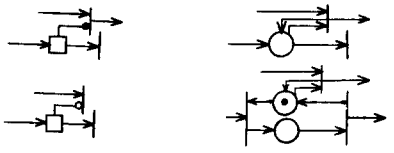
출력측의 Transition이 Firing하게 되면, 이 transition의 입출력 Box에 token이 이동된다.

(2) 허가 Arc

단순 Arc와 같이, Box 에 token 이 있어야 Transition이 enable 하게되나, 이 transition이 firing 한 후에도 이 Box에 token의 변화는 없다. P→T로 mapping 한다.

(3) 금지 Arc

허가 Arc 의 firing 조건이 반대되는 논리를 가지는 Arc로써, 역시 firing 후에도 token 의 변화는 없다.



(a) S-Net (b) Safe PN

그림2. 허가,금지Arc의 Safe Petri Net에의 구현

4. Marking 함수

Marking 함수 $\mu$는

$$\mu(Bi) \rightarrow \{ 0, 1 \} \quad (5)$$

인 safe한 경우이다. 단, $Bi \in B$

5. Process 입출력 함수

S-Net에 가제어성과 가관측성을 가지게 하기위해 Process 출력함수 $\delta$ 와 Process 입력함수 $\phi$를 정의한다. A 를 제어신호의 집합, E 를 관측신호의 집합이라고 할 때,

$$\delta(Bi) = Xi$$

$$\phi(Bi) = \{ Yi1, Yi2, Yi3, \ldots Yin \} \quad (6)$$

( 단, $Xi \in A$, $Yij \in E$, $Bi \in$ Activity Box)

Process 출력함수는 Activity Box당 한개의 신호만을 출력할 수 있으며, 제어기에서 반환되는 관측신호는 E 집합중의 한 원소가 된다. 관측신호에 의해 Selective Transition 의 조건분기가 이루어진다.

6. Process 상태함수

$$U(Bi) \rightarrow \{ 0, 1, 2 \} \quad (7)$$

( 단, $Bi \in$ Activity Box )

0 : Token 이 유입되기 전의 상태

1 : Token 이 유입되었고, 외부기기에 출력코드를 전송한 상태

2 : Token 이 유입되었고, 외부기기에서 Return 코드를 전송받은 상태

이 Process 상태함수의 도입으로 Box의 상태를 감시하여 제어할 수 있게하였고, MFG의 External Arc 와 같은 기능을 할 수 있다.

7. Transition Firing 규칙

어떠한 Transition $ti \in T$가 marking $\mu$에서 enable

$\Leftrightarrow \mu(I(ti)) = 1$ and $\mu(O(ti)) = 0$

$U(I(ti)) = 2$ (단, I(ti) 가 Activity Box)

$U(O(ti)) = 0$ (단, O(ti) 가 Activity Box)

$PA(ti) = 1$ (단, PA는 ti의 허가 Arc논리)

$IA(ti) = 0$ (단, IA는 ti의 금지 Arc논리) (8)

어떠한 Transition $ti \in T$ 가 marking $\mu$에서 firing

$\Leftrightarrow \mu(I(ti)) = 0$

$\mu(O(ti)) = 1$

$U(I(ti)) = 0$

$U(O(ti)) = 1$ (9)

용량성을 가지는 Box인 경우 표 1 과 같고, 허가, 금지 Arc의 논리는 표 2 와 같다.

| TYPE | Firing 필요조건 ti | Firing 필요조건 to | Firing 후 Marking ti | Firing 후 Marking to |
|---|---|---|---|---|
| S | n = 0 | n = 1 | n ← 1 | n ← 0 |
| M | n < N | n > 0 | n ← n+1 | n ← n-1 |
| G | n < N | n = N | n ← n+1 | n ← 0 |
| D | n = 0 | n > 0 | n ← N | n ← n-1 |

단, n : token 의 수

N : 용량

ti : $I(Bi) \in T$, $Bi \in$ Activity Box

to : $O(Bi) \in T$, $Bi \in$ Activity Box

표 1. 용량성 Box에서의 Firing 규칙.

| Gate Arc Type | Gate Arc의 논리값 0 | 1 |
|---|---|---|
|  | n < N | n = N |
|  | n = 0 | n > 0 |

단, n : token 의 수

N : 용량

표 2. Gate Arc의 논리값.

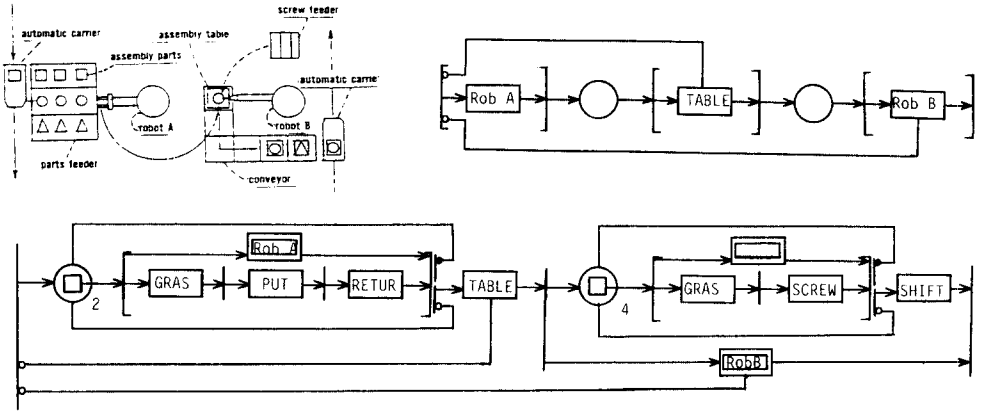



그림 3. S-Net에 의한 체계적 설계의 예

8. S-Net에 의한 제어사양 표현의 예

그림3 과 같이 Robot A 가 part feeder에서 두개의 Assemble part 를 집어서 Assemble table 에서 조립한 후, Robot B가 4개의 screw 를 feeder에서 집어서 assemble part에 삽입하는 경우이다 [2].

이를 C-Net의 표현과 같은 무리한 확장을 행하지안고 S-Net의 구조만으로 C-Net의 Similiar Job Repeation Block, Receive Box, Count Gate 등의 표현이 가능함을 알 수 있다.

III. Case Graph 를 이용한 Scanning Algorithm

어떤 특정한 상태에서 도달가능한(Reachable) 모든 상태를 Directed Arc로 연결한 그래프를 Case Graph [7] 이라고 한다.

A. S-Net에서의 Case Graph의 성질

S-Net가 Safe한 그래프모델이기 때문에 Case Graph 를 적용하면 반드시 유한하게되므로 이를 해석과 Scanning 을 위한 도구로서 사용할수 있게된다. Case Graph를 작성하면 다음의 성질을 알 수 있다.

1. S-Net가 Cyclic 시스템인 경우

어떤 초기 상태에서 시작하여 다시 초기 상태로 돌아올 수 있는 시스템을 Cyclic 하다고 한다. 이러한 시스템에서는 Live 한 Firing Sequence σ가 존재하여 Case Graph상에서 반드시 초기 상태로 돌아오는 Path가 존재한다. 이 시스템은 Live 하다.

2. S-Net가 부분적 Cyclic 시스템인 경우

이러한 시스템은 Case Graph 상에서 Sub-Loop가 존재하여, Sub-Firing Sequence σ1이 계속 반복된다면 부분적인 Dead Lock 상태를 이르킬 수 있는 가능성이 있다.

3. S-Net가 Dead Lock 시스템인 경우

이 시스템은 Dead 상태로 가는 σ가 존재하므로, Case Graph 상에는 Loop가 되지않고 말단 node로 가는 Path가 존재하여, 이 Path로 Firing 한다면 DeadLock 상태를 일으킨다.

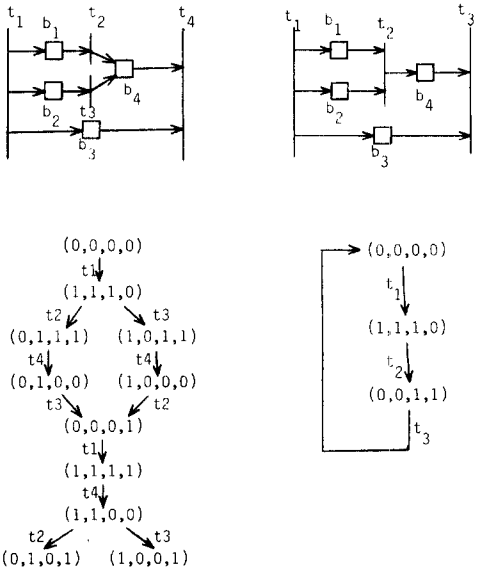



그림 4. Case Graph 를 S-Net에 적용한 예

B. 실시간 제어를 위한 Scanning Algorithm

Safe한 Petri Net은구조적으로 도달가능한 상태가

유한개일뿐만 아니라, 임의의 Case에서 바로 도달가능한 (Direct Reachable) 상태는 전체 System의 marking을 이 case로 하였을 때와 동일함으로, 검사하여야 하는 transition의 갯수는 극소수임에 착안하여, 실시간제어에 적합한 다음의 Scanning Algorithm을 개발하였다.

Step 1 : 일단, 주어진 S-Net의 Case Graph를 소프트웨어에 의해 Linked List구조로 자동생성 시킨다.

Step 2 : 새로운 pointer A를 초기 marking인 state를 가르키도록 한다.

Step 3 : Pointer A 가 가르키는 State에 기억된 enable transition을 조사하여 enable 검사를 행한 후, enable 된 transition이 발견되면 step 4 로 가고 발견되지 않으면 input box가 동작중이므로 계속해서 enable 검사를 한다.
기억된 transition이 없으면 Dead Lock 이므로 실행을 중지한다.

Step 4 : Step 3 에서 발견된 enable transition을 firing 하고, 이 transition이 가르키는 다음의 state로 pointer A 를 옮긴다.

Step 5 : 다시 Step 3 을 반복한다.

## V. 결 론

본 논문에서는 지금까지 제시되었던 여러 Petri Net의 문제점을 파악하여 개선된 S-Net 모델을 제시하였고, 이를 실시간처리 하기위한 Scanning Algorithm을 제시하였다. 이로써 Computer Integrated Manufacturing (CIM) System 구현을 위한 이론적이고 체계적인 방법을 시도하였다.

## VI. 참고문헌


[1]. Kensuke Hasegawa and Paulo Eigi Miyagi : Application of the Mark Flow Graph to Represent Discrete Event Production System and System Control, Tran. of SICE, 24-1, 69/75, Jan. 1988

[2]. Tomohiro Murata and Norihisa Komoda : A Petri Net-Based Controller for Flexible and Maintainable Sequence Control and its Application in Factory Automation, IEEE Transaction on Industrial Electronics, vol IE-33, no, 1, pp 1/8, Feb., 1986.

[3]. Kichie Matsuzaki, et. al :Petri Net structured sequence - control language with GRAFCET like graphical expression for programmable controllers, Proc. IECON '85, pp 423-438, 1985.

[4]. H. Atabachche, et. al : From Petri Net based PLCs to knowledge base control, IECON'86, pp 817-822, 1986

[5]. Masaki Hasegawa and Yoshinori Sakaue : Programming System for Work Cell, Proc. IECON '86,pp 443-448, 1986.

[6]. J.L. Peterson : Petri Net Theory and the Modelling of Systems, Prentice-Hall, 1981.

[7]. Wolfgang Reisig : Petri Nets, Springer Verlag, 1985.