실시간 시간논리 구조를 이용한 비결정적 이산사건 동적시스템의 모델링

Modeling of Nondeterministic Discrete Events Dynamic System Using Real-Time Temporal Logic Framework

김진권*, 이원혁, 최정내, 황형수
원광대학교 제어계측공학과, 전라북도 익산시 신룡동 344-2 [5][7][0]-[7][4][9]
Tel. : 0653-850-6345, Fax. : 0653-853-2196
E-mail : jinpang@gaebyok.wonkwang.ac.kr

Jin Kwon Kim*, Won Hyok Lee, Jeoung Nae Choi, Hyung Soo Hwang
Dept. of Control & Instrumentation Engineering, Wonkwang Univ., Iksan, Korea
Tel. : 0653-850-6345, Fax. : 0653-853-2196
E-mail : jinpang@gaebyok.wonkwang.ac.kr

이산사건 시스템은 시간의 이산순간에 상태변화가 발생하는 시스템으로서 공정제어, Robotics, 교통시스템, Flexible Manufacturing System, 통신등 많은 분야의 시스템이 이산사건 시스템들이지만 아직도 포괄적이고 융통성 있는 제어이론이 연구되지 않았다. 본 연구는 특히 Real-Time Temporal Logic Framework(RTTL)에서 비결정적으로 발생되어지는 확정적인 사건들로써 유발되어지는 비결정적 이산사건 시스템의 모델링 방법을 제시하였다. 이 방법을 두 개의 machine들로 구성된 Flexible Manufacturing System(FMS) 에 적용하여 설명하였다. 이 방법은 복잡한 이산사건 시스템의 모델링을 모듈화하여 간편하게 표현 할 수 있는 우수한 특성을 가지고 있다.

1. 서론

연속변수 동적시스템에는 상미분, 편미분 그리고 차분 방정식 등에 의한 안정도 이론, 최적제어 이론 등과 같은 많은 제어이론이 연구되어져 있으나, 여러 응용분야에서 발견되는 이산사건 시스템은 시간의 이산순간에 상태변화가 발생하는 시스템으로서 아직도 포괄적이고 융통성 있는 제어이론이 연구되지 않았다. 현재 이산사건 시스템의 모델링 및 제어에 대한 연구가 Automata and Formal Language[7], Petri Net[5], Temporal Logic Framework[4,6] 등을 이용하여 많은 연구가 이루어지고 있으며, 비결정적 이산사건 시스템의 모델링 및 제어에 대하여도 많은 연구가 이루어지고 있다.

본 논문은 $\neg$(not), $\wedge$(and), $\vee$(or), $\rightarrow$(implies), $\leftrightarrow$(if and only if) 등의 종래의 부울논리 연결자에 $\Box$(henceforth), $\Diamond$(eventually), $\bigcirc$(next), $\sqcup$(until), P(proce ed) 등과 같은 시간의 변화와 량을 표현하기 위한 시간논리 연산자[4]와 ∇ (certainly) 와 $\triangle$ (possible) 같은 형식 연산자[2]를 사용한 실시간 시간논리 구조를 정의하고, 이를 비결정적 이산사건 시스템에 적용하여 모델링하는 방법을 제시한다. 이를 두 개의 machine들로 구성된 FMS [9]에 적용하여 설명하였다.

2. 비결정적 이산사건 시스템

발생될 사건에 대한 확률적 지식을 알고있는 비결정적 이산사건 시스템을 확률적 이산사건이라고 하며, 전체 시스템은 Σ으로 명시한다. 이것은 또한 확실한 확률 분포를 가지는 반순서에 의해 발생되어지는 사건들의 집합이며, 이때 사건을 E라고 명시한다. 시스템은 이들 사건에 의한 특정조건으로 수립되어지며 이 규정은 논리식의 집합으로써 표현된다. 이 논리식을 F라고 명시한다. 시스템의 상태는 F의 부분집합으로 분류되고, 한 사건은 두 개의 상태사이 또는 순간적인 상황에 의해 유발되어지는 논리식의 부분집합들 간의 천이관계로 볼 수 있다. 사건은 자연적, 비

동기적 그리고 순간적으로 발생되어진다. 각 가능한 사건들의 실행은 BSM(bounded stochastic model)으로 정의한다.

3. BSM 과 RTTL

3.1 Bounded Stochastic Model

<정의 3.1> 집합 F가 주어졌을 때, bounded stochastic model M은 (S, E, s_0, A, p)인 5개의 요소 로 구성된다.

- S − 공집합이 아니며 셀 수 있는 상태들의 집합이다.
- E − 발생 가능한 사건들의 집합이다. 각 사건 e_{ij}는 $s_i \rightarrow s_j$로의 천이를 나타낸다. ($e_{ij} \in E$, $s_i, s_j \in S$)
- s_0 − 초기상태이다.($s_0 \in S$)
- A − $A: S \rightarrow F^*$, (F^*는 F의 부분집합들의 전체집합)이다. S안에 있는 모든 상태가 F^*중 하나의 집합인 식으로 분류되어지는 함수이다.
- P − $P: E \rightarrow [0,1]$, 모든 발생 가능한 사건 e_{ij}에 관련되어지며 확률 $p_{ij} \triangleq P(e_{ij})$을 가진다. $s_i \in S$, $e_{ij}: s_i \rightarrow s_j$, $j \in J$ 일 때 $\sum_{j \in J} P_{ij} = 1$ 이다.여기서 J는 무한이다.

J가 singleton $\{j\}$이면, $P_{ij} = 1$이고 사건이 결정적(deterministic)으로 발생하는 경우를 가진다. 이제, 초기상태 s_0에서 시작되는 궤적으로써 M을 해석할 수 있으며, 상태 천이를 실행시키는 사건의 열을 발생 시킬 수 있다. 사건의 천이 $e_{ij}: s_i \rightarrow s_j$는 아래첨자를 재 기입함으로써 $e_{k+1}: s_k \rightarrow s_{k+1}$ ($k = 0, 1, \cdots$)로 다시 쓸 수 있다. s_0 부터 s_n까지의 궤적 또는 경로는 $\sigma_n \triangleq s_0\, s_1 \cdots s_n$ 로써 주어진다. S상의 모든 유한(무한) 열(sequence)들의 집합을 S^*로 표시하고, W_s는 s($s_0 = s$)에서 시작되는 S^*의 모든 경로의 집합으로 표시한다. BSM에서 임의의 $s \in S$에 대하여 Markov chains 이론에 의하여 함수 p는 집합 W_s에서 하나의 확률분포로써 얻어진다. 이 확률분포는 $\tilde{p}_s$로 표시한다. 이것을 통해 S^*의 모든 열 σ의 집합 B를 얻기에 충분하며 측정 할 수가 있다. 유한 열 σ_n에 대하여 $\tilde{p}_{s_0}(B) = p(e1) \times p(e2) \times \cdots \times p(en)$.

모든 $\sigma \in S^*$, $\sigma = s_0\, s_1\, s_2\, s_3 \cdots$ 이므로, 다음 열 $\sigma^{(1)}$ 은 열 $s_1\, s_2\, s_3\, s_4 \cdots$ 로 표시한다. 그리고 열 $s_2\, s_3\, s_4\, s_5 \cdots$ 는 $\sigma^{(2)}$ 로 표시된다. 따라서, 한 상태는 그 사건의 현재상태의 속성을 표현하며 그리고 상태들의 한 열은 시간에 따라 시스템의 가능한 행태나 궤적을 표현한다.

3.2 실시간 시간논리 구조

본 논문에서 다루어지는 실시간 시간논리 모델은 (S, E, s_0, A, p, L, U)인 7개의 요소로 구성된다. S, E, s_0, A, p는 정의 3.1 에서 이미 설명하였다.

<정의 3.2> 실시간 요소인 L, U는 모델의 최대와 최소의 유지시간(holding time)이다.

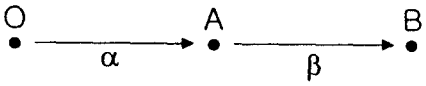
- $L: S \rightarrow R^+$, $U: S \rightarrow R^+$ − 상태 $s \in S$ 가 주어졌을 때, L과 U는 임의의 실시간 영역이다. 만약, $L = 0$ 이고 $U = \infty$ 이면, 시스템은 실시간 영역을 가지지 않는다.[8]

3.2.1 증명 시스템(The Forth Level)

실시간 추론을 위한 증명시스템의 Forth Level을 다시 고찰하면,[2] 이 level은 증명 시스템에 실시간 추론을 도입한다. 이 level에서 주어진 한 상태에서 유지시간을 표현할 수 있는 방식을 소개한다. 전체 상수기호들은 실수들이고 개별 변수기호는 기호 t이다. 식 $t = T$는 시간이 이 식의 경우 T시간단위의 값을 가진다는 의미이다. 하위(lower)와 상위(upper)영역의 표현은 식(1)과 같다.

$$\begin{aligned} L_A = a \;&\triangleq\; \Box[\delta = \alpha \wedge t = T] \\ &\Rightarrow \Diamond[\delta = \beta \wedge t \geq T + a] \\ U_A = b \;&\triangleq\; \Box[\delta = \alpha \wedge t = T] \\ &\Rightarrow \Diamond[\delta = \beta \wedge t \leq T + b] \end{aligned} \quad (1)$$

$\alpha \in E$는 <그림 1>에서처럼 상태 O ($O \in S$) 에서 A($A \in S$) 에 이르는 사건이며 $\beta \in E$ 는 상태 A($A \in S$) 에서 B($B \in S$) 에 이르는 사건이다. 영역 L_A와 U_A는 각각 최소와 최대 유지시간을 표현한다.



< 그림 1 >

세 가지 증명단계에서 표현된 규칙과 정리들 이외에, 우리는 실시간에서 파생된 두 가지 상태개념을 표현한다.[1,8]

$$\begin{aligned} &\Box[(\forall T : (w_1 \wedge t = T) \\ &\qquad \Rightarrow \Diamond(w_2 \wedge t \leq T + d_1))] \\ &\Box[(\forall T : (w_2 \wedge t = T) \\ &\qquad \Rightarrow \Diamond(w_3 \wedge t \leq T + d_2))] \end{aligned} \quad (2)$$

$$\Box[(\forall T : (w_1 \wedge t = T) \Rightarrow \Diamond(w_3 \wedge t \leq T + d_1 + d_2))]$$

w_1, w_2 와 w_3 가 임의의 시간논리식이라 하고, d_1, d_2 가 시간 t와 같은 종류의 임의의 상수들이라고 하고, T는 시간 t의 임의 정도 값을 나타낸다. 그때 식(2)와 같다. 이 규칙은 시간 논리식 w_1 을 만족하는 상태로 부터 시간 논리식 w_2 를 만족하는 상태로의 최대 시간간격이 d_1 이고, w_2 를 만족하는 상태로 부터 w_3 를 만족하는 상태로의 최대 간격이 d_2 일 때, 식 w_1 로부터 식 w_3 로의 최대 시간간격은 $d = d_1 + d_2$ 가 됨을 나타낸다. 식(3)은 Temporalized Chain Rule 또는 TCHAIN 이다.[4] 이때, v_i, w_i는 모두 상태식의 열이고, $d = d_1 + d_2 + \cdots + d_n$ 이다.

$$\Box[(v_i \wedge w_i) \Rightarrow \bigcirc \bigvee_{j=0}^{i-1} v_j]$$
$$\Box[v_i \Rightarrow \Diamond w_i]$$
$$\Box[v_i \Rightarrow (\bigcirc(\bigvee_{j=0}^{i-1} v_j) \vee \bigcirc v_i)]$$
$$\forall \ i = 1, \cdots, k;$$
$$\Box[v_i \wedge w_i \wedge t = T \Rightarrow \Diamond(\bigvee_{j=0}^{i-1}(v_j \wedge w_j \wedge t \leq T + d_j))] \quad (3)$$
$$\Box[(\bigvee_{i=0}^{n} v_i) \Rightarrow ((\bigvee_{i=1}^{n} v_i) \sqcup (v_0 \wedge t \leq T + d))]$$

4. RTTL에 의한 제어규칙 설계

RTTL을 이용하여 확률적 지식을 알고있는 비결정적 DES에 대한 제어문제에 사용할 수 있다. 이제, RTTL을 이용하여 플랜트, 페루프 시스템, 그리고 제어기 설계를 포함하는 비결정적 DES에 대한 동작설명의 일반적 모델을 고찰한다.

4.1 플랜트 specification

시간에 대해 독립적이며 플랜트에서 발생 가능한 상태값을 나타내는데 전체상수기호 $A, B, C, \cdots$ 를 사용하고 시간에 대해 유동적이며 플랜트의 상태들을 나타내는데 $x_1, x_2, \cdots, x_n$ 를 지역변수기호로 사용한다. 플랜트의 기본적인 동작설명은 사건의 발생에 따른 상태변화를 묘사한다. 일반적인 모델은 식(4)와 같다.

$$\Box[\wedge_i(\delta = \alpha_i \Rightarrow f(x_i, \bigcirc x_i) \wedge (\wedge_{j \neq i}(\bigcirc x_j) = x_j))] \quad (4)$$

이때, 사건 α_i 의 발생은 상태 x_i 의 변화를 유발한다. f는 연산 함수이며 x_i가 변하는 과정을 나타낸다. 결정적 DES일때에는, 현재상태 $x_i = P$ 이고 다음상태 $(\bigcirc x_i) = B$ 이면, 그때 함수 $f(x_i, \bigcirc x_i)$ 는 식(5)와 같은 의미이다.

$$x_i = P \wedge (\bigcirc x_i) = B \quad (5)$$

비결정적 DES에서는 다음상태가 한가지 이상이므로 $(\bigcirc x_i) = B$ 또는 $(\bigcirc x_i) = C$이다. 그때 함수 $f(x_i, \bigcirc x_i)$ 는 식(6)과 같이 사용한다.

$$x_i = P \wedge [(\bigcirc x_i = B \vee (\bigcirc x_i) = C] \wedge [\triangle(\bigcirc x_i) = B \wedge \triangle(\bigcirc x_i) = C] \quad (6)$$

이 식은 현재값으로부터 다음상태값이 결정되어진 경우와 확률적 값을 가지는 비결정적인 경우까지 사건의 발생에 따른 상태변화를 나타낸다. 이밖에, 시스템의 모든 발생 가능한 상태값들이 구분되어진다는 것을 명시해야하며 또한, 시스템의 초기상태를 명시해야한다.

4.2 페루프 시스템 specification

플랜트가 원하는 동작을 하도록 시스템의 움직임을 설계하는 것으로써, 다음과 같은 성질을 포함한다.

- Mutual Exclusion(상호배타): 두 가지 사건이 동시에 발생할 수 없다.
- Priority(우선순위): 특별한 사건은 어떤 특별한 조건하에서 항상 금지되거나 혹은 항상 발생해야만 한다.
- Precedency(선행조건): 사건들은 어떤 특별한 순서에 따라 발생한다.

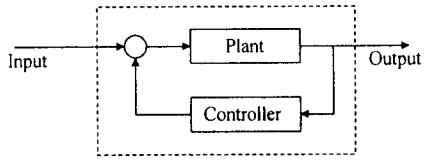
4.3 제어기 설계 specification

비결정적 DES에 대한 제어기의 설계에 있어서, 설계의 목적은 시스템이 바라는 대로 동작함으로써 제안된 사용자의 요구가 보장되어 지는 것이다. 제어기의 상태들은 지역변수기호로써 표현하며 제어기에 대한 Data를 저장하는데 사용한다. 시스템의 감시와 제어를 위하여 제어기의 상태들은 기본적으로 queue q와 count c로 구성한다. Queue는 1부터 N사이의 임의의 한정된 정수 수열(sequence) 또는 기호 ε로 정의되는 공 수열로 할당된 값이며, Count는 음이 아닌 정수 값들로 할당한다. 즉, 제어기의 기본적인 동작설명은 식(7)과 같은 일반적인 형태를 가진다.

$$\Box[\wedge_i(\delta = \alpha_i \Rightarrow g(q, \bigcirc q) \wedge h(c, \bigcirc c))] \quad (7)$$

모든 발생 가능한 사건은 queue와 count의 값을 변화시킨다. 식(7)에서 g와 h는 연산 함수이며 q와 c의 변화를 나타낸다. 그리고, 함수 g는 다음과 같은 형태로 사용한다. $(\bigcirc q) = q * i$: queue q의 맨 끝에 i를 저장한다. $i < q$: i는 queue q의 초기값이다. $(\bigcirc q) = q\,|^1$: q의 첫번째 원소를 제거하고 얻어진 열이다. $(\bigcirc q) = q$: 상태 유지한다. $q = \varepsilon$: queue q는 공수열이다. 함수 h는 다음과 같은 형태로 사용한다. $(\bigcirc c) = c$: count 값은 변하지 않는다. $(\bigcirc c) = c + 1$: count값이 1만큼 증가한다. $(\bigcirc c) = c - 1$: count값이 1만큼 감소한다. $c = 0$: count값이 0이다. 제어기의 기본적인 동작설명 이외에도 제어기의 초기상태를 명시해야한다. 비결정적 DES에 대한 제어기의 연산동작은 queue의 내용목록과 count의 값을 조절하는 것이다. 즉, 제어기는 queue와 count로써 상호배타, 우선

순위, 선행조건과 같은 페루프 시스템이 원하는 특성들을 보장할 수 있다. 페루프 시스템은 <그림 2>와 같다.



< 그림 2 >

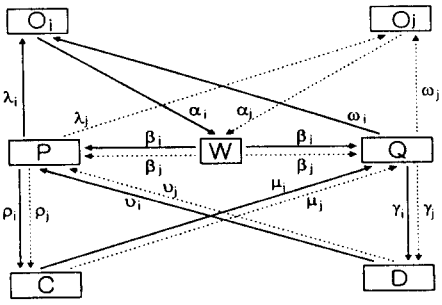
이와 같은 결과는 유한상태뿐만 아니라 무한상태에서 플랜트와 제어기에 폭넓게 활용할 수 있다.

5. RTTL에서 시스템 모델링의 예

이 예는 실시간 특성들을 강조하기 위해서 flexible manufacturing system을 들었다.[9] flexible manufacturing system은 두 개의 다른 종류 C_i 와 C_j 의 공정부품을 처리할 수 있는 두 개의 장치로 구성된다. 이 부품들은 다음 규칙에 따라서 처리된다

- $R1$ – C_i와 C_j의 모든 부품이 두 개의 장치 m_1 과 m_2를 통하여 처리된다. 먼저 처리되는 것이 어느 장치인가는 상관없다.
- $R2$ – 부품들은 상호 배타적으로 수행된다. 두 개 부품은 같은 시간에 같은 장치에서 수행될 수 없다. 즉, 부품들은 각 장치에서 한번에 하나가 수행된다.
- $R3$ – 부품들은 장치에 접근하기 위해 다음과 같은 확률분포를 가지고 있다. C_i (C_j)의 부품은 확률 p_{i1} (p_{j1})을 가지고 장치 m_1 에서 처리할 수 있고, 확률 p_{i2} (p_{j2})를 가지고 장치 m_2 에서 처리할 수 있다.
- $R4$ – 각 장치에 동일한 처리시간 Z가 고려되어 진다. 두 개의 장치에서 두 부품 수행시간의 간격은 $2Z$보다 절대로 클 수 없다.

<그림 3>에서는 두 공정부품(process parts) C_i 와 C_j 에 대한 상태천이를 가진다. $0<i,j\le N, i\neq j$



< 그림 3 >

- 부품들의 발생 가능한 상태(state).

O_i – workstation i의 바깥부분.
O_j – workstation j의 바깥부분.
W – 첫 번째 수행(process)을 위한 지연.
P – 장치 m_1에서 수행되어지는 것.
Q – 장치 m_2에서 수행되어지는 것.
$C(D)$ – m_1 (m_2)에서 수행된 후 m_2 (m_1)로 보내기 위한 지연.

- 발생 가능한 사건(event).

a_i , a_j – 부품(part)의 도착.
β_i , β_j – 처음장치에서 수행 시작(장치들의 확률적 선택을 포함) .
ρ_i , ρ_j – m_1 에서 첫 번째로 처리되어진 부품의 queue(queue D_q).
γ_i , γ_j – m_2 에서 첫 번째로 처리도어진 부분의 queue(queue C_q).
μ_i , μ_j – 다음장치 m_2에서 수행시작(queue P_q).
ν_i , ν_j – 다음장치 m_1 에서 수행시작(queue Q_q).
λ_i , λ_j – 기계 m_1 에서 이탈
ω_i , ω_j – 기계 m_2 에서 이탈

- $$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N} \delta=x_{i(j)} \Leftrightarrow (\delta=x_i)\vee(\delta=x_j)\right] \quad (6)$$

x_i와 x_j는 임의 사건을 뜻하며, (6)식은 전체적인 동작설명에서 유용하게 사용된다.

프랜트 specification

$$\Box\left[\bigvee_{\substack{i=1\\ j=1\\ i\neq j}}^{N} (\delta=\alpha_{i(j)}\vee\delta=\beta_{i(j)}\vee\delta=\rho_{i(j)}\vee\delta=\gamma_{i(j)} \vee\delta=\mu_{i(j)}\vee\delta=\nu_{i(j)}\vee\delta=\lambda_{i(j)}\vee\delta=\omega_{i(j)})\right] \quad (P1$$

위 식은 모든 발생 가능한 사건들의 집합을 표현한다.

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N} \{\delta=\alpha_{i(j)} \Rightarrow (x_{i(j)}=O_{i(j)}\wedge(\bigcirc x_{i(j)})=W)\} \wedge \bigwedge_{\substack{j=1\\ i=1\\ i\neq j}}^{N} (\bigcirc x_{j(i)})=x_{j(i)}\right] \quad (P2$$

위 식은 부품 $i(j)$의 도착 결과를 표현한다: O에서 W로의 상태 천이이다.

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N} \{(\delta=\beta_{i(j)}) \Rightarrow (x_{i(j)}=W\wedge((\bigcirc x_{i(j)})=P \vee(\bigcirc x_{i(j)})=Q)\wedge\triangle(\bigcirc x_{i(j)})=P\wedge\triangle(\bigcirc x_{i(j)})=Q)\}\right]$$

(P3

위 식은 사건 $\beta_{i(j)}$ 후에 두 상태중 한 가지로 확실히 상태 천이한다: $x_{i(j)}$ 의 다음상태는 P또는 Q로써 두 가지 상태중 하나이다.

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N} \{\delta=\rho_{i(j)} \rightarrow (x_{i(j)}=P\wedge(\bigcirc x_{i(j)})=C)\}\right] \quad (P4$$

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N} \{\delta=\gamma_{i(j)} \rightarrow (x_{i(j)}=Q\wedge(\bigcirc x_{i(j)})=D)\}\right] \quad (p5$$

위 식은 사건 $\rho_{i(j)}$ ($\gamma_{i(j)}$)후에 $x_{i(j)}$ 의 상태가 P (Q)에서 C(D)로 천이한다.

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{\delta=\mu_{i(j)}\Rightarrow x_{i(j)}=C\wedge(\bigcirc x_{i(j)})=Q\wedge(L_Q=Z_1\wedge U_Q=2Z_1)\}\right] \quad \text{(P6}$$

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{\delta=\nu_{i(j)}\Rightarrow x_{i(j)}=D\wedge(\bigcirc x_{i(j)})=P\wedge(L_P=Z_2\wedge U_P=2Z_2)\}\right] \quad \text{(P7}$$

위 식은 사건 $\mu_{i(j)}$ ($\nu_{i(j)}$) 후에 $x_{i(j)}$ 의 상태가 C (D)에서 Q (P)로 천이한다: 상태 P (Q)에서 C (D)까지의 전체 수행시간은 최대 $2Z$ 이다.

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{\delta=\lambda_{i(j)}\Rightarrow x_{i(j)}=P\wedge(\bigcirc x_{i(j)})=O_{i(j)}\}\right] \quad \text{(P8}$$

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{\delta=\omega_{i(j)}\Rightarrow x_{i(j)}=Q\wedge(\bigcirc x_{i(j)})=O_{i(j)}\}\right] \quad \text{(P9}$$

위 식은 사건 $\lambda_{i(j)}$ ($\omega_{i(j)}$)후에 $x_{i(j)}$ 의 상태가 P (Q)에서 $O_{i(j)}$ 로 천이한다

$$\Box[O_{i\neq}O_j\wedge O_i\neq W\wedge O_i\neq P\wedge O_i\neq Q\wedge O_i\neq C\wedge O_i\neq D\wedge O_j\neq W\wedge O_j\neq P\wedge O_j\neq Q\wedge O_j\neq C\wedge O_j\neq D\wedge W\neq P\wedge W\neq Q\wedge W\neq C\wedge W\neq D\wedge P\neq Q\wedge P\neq C\wedge P\neq D\wedge Q\neq C\wedge Q\neq D\wedge C\neq D] \quad \text{(P10}$$

위 식은 모든 상태들이 서로 구분(distinct)이 됨을 나타낸다.

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}(x_{i(j)}=O_{i(j)})\right] \quad \text{(P11}$$

위 식은 초기 상태조건을 나타낸다.

페루프 시스템 specification

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}(x_{i(j)}=P\Rightarrow\bigwedge_{\substack{i=1\\ i\neq j}}^{N}x_{j(i)}\neq P)\right] \quad \text{(CL1}$$

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}(x_{i(j)}=Q\Rightarrow\bigwedge_{\substack{j=1\\ i\neq j}}^{N}x_{j(i)}\neq Q)\right] \quad \text{(CL2}$$

부품들이 동시에 같은 장치에서 공정처리 되어질 수 없다: 상호배타

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}(\delta=\mu_{i(j)}\wedge x_{i(j)}=C)\vee(\delta=\nu_{i(j)}\wedge x_{i(j)}=D)\Rightarrow\delta\neq\beta_{i(j)}\right]$$

(CL3

임의 장치에서 첫 번째로 처리되어진 부품이 다음장치로 천이되는 부품은 새로이 공정처리 되어지는 것보다 우선한다: 우선순위

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{x_{i(j)}\neq C\Rightarrow((\delta=\rho_{i(j)}\,\mathrm{P}\,\delta=\rho_{j(i)})\Rightarrow(\delta=\mu_{i(j)}\,\mathrm{P}\,\delta=\mu_{j(i)}))\}\right] \quad \text{(CL4}$$

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{x_{i(j)}\neq D\Rightarrow((\delta=\gamma_{i(j)}\,\mathrm{P}\,\delta=\gamma_{j(i)})\Rightarrow(\delta=\nu_{i(j)}\,\mathrm{P}\,\delta=\nu_{j(i)}))\}\right] \quad ($$

CL5

먼저 도달된 명령에 따라서 부품 i 의 다음상태가 j 의 다음상태보다 앞선다면 부품 i 는 부품 j 보다 전에 보내어진다: 선행조건

$$\Box\left[\left\{\left(\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}(x_{i(j)}=O_{i(j)}\vee x_{i(j)}=W)\right)\wedge\neg\left(\bigwedge_{k=1}^{N}x_k=O_{i(j)}\right)\right\}\Rightarrow\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\delta=\beta_{i(j)}\right]$$

(CL6

모든 부품이 작업장 밖이나 또는 처리되기를 기다리고 있지만, 그것들 모두 작업장밖에 있는 것이 아니라면, 사건 $\beta_{i(j)}$ 는 반드시 발생한다.

$$\Box[((\delta=\gamma_{i(j)}\vee\delta=\rho_{i(j)})\wedge t=T)\Rightarrow(\Diamond(\delta=\nu_{i(j)}\vee\delta=\mu_{i(j)})\wedge t\leq T+2Z_{1(2)})]$$

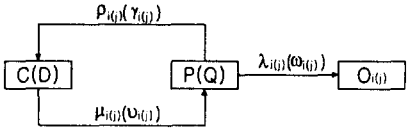
(CL7

동작설명(specification)은 처음장치로부터 pf 부품이 다음장치로 천이될 때 상태 C (D)에서 최대 수행시간을 $2Z$ 로 지정한다. 이때, $Z_1\neq Zz_2$

제어기 specification

제어기의 동작설명을 할 때 저장된 Data를 표현하기 위해 지역변수기호들을 사용한다. 기호 C_q , D_q , P_q , Q_q 는 각각 장치 m_1 과 m_2 에 관련된 queue를 표현한다. 그리고 1부터 N까지의 값(value)을 저장하거나, 공 수열기호 ε 을 표현할 수 있다. 작업장내에서 첫 번째로 처리되어진 부품이 아닌 부품의 수는 c 로 주어진다.

식 ($CL7$ 을 만족하기 위해 우리는 Z_1 (Z_2)와 $2Z_1$ ($2Z_2$)의 단위시간 사이에서 사건 $\mu_{i(j)}$ 와 $\nu_{i(j)}$ 가 강제로 발생하도록 해야한다. 이것은 사건들 $\mu_{i(j)}$ 와 $\nu_{i(j)}$ 이 설계되어진 변환이 발생하는 것으로부터 얻어진다. <그림 4>에서 이것을 보였다.사건 $\rho_{i(j)}$ ($\gamma_{i(j)}$)가 발생할 때, P (Q)로부터 C (D)로 제어기의 상태가 천이한다. 이 후에, 다음사건 $\mu_{i(j)}$ ($\nu_{i(j)}$)이 제어기 등식으로부



< 그림 4 >

터 정해진 최소와 최대 시간간격사이에서 발생할 것이다. 제어기 등식들은 다음과 같다

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{\delta=\rho_{i(j)}\Rightarrow((Y_{i(j)}=P)\wedge(\bigcirc Y_{i(j)}=C)\wedge(\bigcirc C_q)=C_q*i(j)\wedge(\bigcirc c)=c-1)\wedge L_C=Z_1\wedge U_C=2Z_1)\}\right] \quad \text{(C1}$$

$$\Box\left[\bigwedge_{\substack{i=1\\ j=1\\ i\neq j}}^{N}\{\delta=\gamma_{i(j)}\Rightarrow((Y_{i(j)}=Q)\wedge(\bigcirc Y_{i(j)}=D)\wedge(\bigcirc D_q)=D_q*i(j)\wedge(\bigcirc c)=c-1))\wedge L_D=Z_2\wedge U_D=2Z_2)\}\right] \quad \text{(C2}$$

윗 식은 장치 m_1 (m_2)로부터 첫 번째로 처리되어진 부품 i (j)가 온다면, 수(number) i (j)는 queue C_q (D_q)에 저장되며, 그리고 count c 의 값은 1만큼 감소한다. 최소와 최대시간영역을 주목하면, $L_C=Z_1$ (Z_2) 와 $U_C=2Z_1$ (Z_2) 은 각각

상태 $C(D)$에서의 최소와 최대 지연시간이다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{\delta=\mu_{i(j)}\Rightarrow((Y_{i(j)}=C)\wedge(\bigcirc Y_{i(j)}=Q)\wedge i(j)<C_q\wedge(\bigcirc C_q)=C_{q\wedge(\bigcirc c)=c})\}\right] \quad (C3$$

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{\delta=\nu_{i(j)}\Rightarrow((Y_{i(j)}=D)\wedge(\bigcirc Y_{i(j)}=P)\wedge i(j)<D_q\wedge(\bigcirc D_q)=D_q\wedge(\bigcirc c)=c)\}\right] \quad (C4$$

윗 식은 $i(j)$가 queue $C_q(D_q)$의 첫 번째라면, 첫 번째로 처리되어진 부품은 기계 $m_2(m_1)$로 보낸다. 반면에, queue $C_q(D_q)$의 목록과 count c의 값은 상태 유지한다. $Y_{i(j)}$ 값의 변화를 주시하면,

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{\delta=\omega_{i(j)}\Rightarrow((Y_{i(j)}=Q)\wedge(\bigcirc Y_{i(j)}=O_{i(j)}\wedge(\bigcirc D_q)=D_{q-1}\wedge(\bigcirc c)=c)\}\right] \quad (C5$$

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{\delta=\lambda_{i(j)}\Rightarrow((Y_{i(j)}=P)\wedge(\bigcirc Y_{i(j)}=O_{i(j)})\wedge(\bigcirc C_q)=C_{q-1}\wedge(\bigcirc c)=c)\}\right] \quad (C6$$

윗 식은 부품(part)이 장치(machine) $m_1(m_2)$로부터 이탈되면 queue $C_q(D_q)$의 첫 번째 원소가 제거되어진다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{\delta=\beta_{i(j)}\Rightarrow(D_q=\varepsilon\wedge(\bigcirc D_q)=D_q\wedge C_p=\varepsilon\wedge(\bigcirc C_p)=C_p\wedge(\bigcirc c)=c)\}\right] \quad (C7$$

윗 식은 queue C_q와 queue D_q가 공 수열이 될 때의 확률적 선택이다. 그리고 queue의 목록과 count 의 값은 변화가 없다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{\delta=\alpha_{i(j)}\Rightarrow(\wedge(\bigcirc D_q)=D_q\wedge(\bigcirc C_p)=C_p\wedge(\bigcirc c)=c+1)\}\right] \quad (C8$$

윗 식은 어떤 한 부품이 작업장에 도달할 때 c의 값이 1만큼 증가한다. 반면에 queue의 목록은 상태 유지한다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{((\delta=\alpha_i)\wedge(\delta=\alpha_j))\Rightarrow(Y_i=O_i\wedge(\bigcirc Y_i)=W\wedge x_j=O_j\wedge(\bigcirc x_j)=O_j)\}\right] \quad (C9$$

윗 식은 α_i와 α_j가 동시에 발생하면 먼저 사건 α_i가 발생한다. 그리고 나머지는 상태 유지한다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{((\delta\neq\rho_{i(j)})\wedge(\delta=\beta_{i(j)}))\Rightarrow(Y_{i(j)}=W\wedge(\bigcirc Y_{i(j)})=Q)\}\right] \quad (C10$$

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{((\delta\neq\gamma_{i(j)})\wedge(\delta=\beta_{i(j)}))\Rightarrow(Y_{i(j)}=W\wedge(\bigcirc Y_{i(j)})=P)\}\right] \quad (C11$$

윗 식은 사건 $\rho_{i(j)}$나 $\gamma_{i(j)}$가 발생하지 않을 때에는 사건 $\beta_{i(j)}$의 확률적인 선택을 무시하고 선택되지 않은 다른 장치로 부품을 보낸다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{((\delta\neq\rho_{i(j)})\wedge(\delta=\nu_{i(j)}))\Rightarrow(Y_{i(j)}=D\wedge(\bigcirc Y_{i(j)})=P\wedge(\bigcirc P_q)=P_q*i(j)\wedge(\bigcirc c)=c)\}\right] \quad (C12$$

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{((\delta\neq\gamma_{i(j)})\wedge(\delta=\mu_{i(j)}))\Rightarrow(Y_{i(j)}=C\wedge(\bigcirc Y_{i(j)})=Q)\wedge(\bigcirc Q_q)=Q_q*i(j)\wedge(\bigcirc c)=c)\}\right] \quad (C13$$

윗 식은 각각 장치 $m_1(m_2)$가 고장이 나고, 다음 공정으로 가는 사건이 발생했을 때 장치의 고장이 해결 될 때까지 queue $P_q(Q_q)$에 사건들이 저장된다. 그리고 고장이 해결되면 저장된 순서에 의해 다음 공정을 수행한다.

$$\Box\left[\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}\{((\delta\neq\rho_{i(j)})\wedge(\delta\neq\gamma_{i(j)}))\Rightarrow(x_{i(j)}=O_{i(j)}\wedge(\bigcirc x_{i(j)})=O_{i(j)}\wedge(\bigcirc c)=c)\}\right] \quad (C14$$

장치 $m_1(m_2)$가 모두 고장이 나면 부품 $i(j)$는 상태 유지한다.

$$\left[\left\{\bigwedge_{\substack{i=1\\j=1\\i\neq j}}^{N}(Y_{i(j)}=P\wedge Y_{i(j)}=Q)\right\}\wedge(C_q=\varepsilon)\wedge(D_q=\varepsilon)\wedge(P_q=\varepsilon)\wedge(Q_q=\varepsilon)\wedge(C=0)\right] \quad (C15$$

모든 제어기 상태변수의 초기상태를 나타낸다.
이상에서, RTTL로써 선택된 FMS를 효과적으로 모델링을 하였다.

6. 결론

본 논문에서는 실시간 제약을 가진 비결정적 이산사건 시스템에서 상태천이가 결정되어 있지 않고 그 확률적 요소를 알고있는 사건의 상태천이를 실시간 시간논리 구조를 이용하여 모델링함에 있어 기존의 모델[1]와는 다르게 서로 다른 두 가지 부품(part)이 두 종류의 장치(machine)에서 유기적으로 공정처리되는 복잡한 시스템을 사용하여 효과적으로 모델링을 하였다. 앞으로 더욱 더 간편한 모델링 방법과 다른 공정과의 합성에 대한 연구가 이루어져야 될 것이다.

참고 문헌

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

[2]Silva Jr.,Braz I. & Mendes,Rafael S.,*"Genralized real-time temporal logic applied on control of nondeterministic discrete e- vents dynamic systems"* Proc.31th IEEE Conf.D&C, Fueson,Arizona, pp.3369-3373, 1992.

[3]Lehmann,D. & Shelar,S.,*"Reasoning with Time and Chance"*,Information & Control, 53, pp.165-198

[4]Ostroff,J.S.,*"Temporal logic for Real Ti- me Systems"*,Press Limited and Wiley, New

York,1989.
[5]J.L.Peterson,"*Petri Net Theory and the Modeling of Systems*",Prentice-Hall,Englewood Cliffs,1981.
[6]J.G.Thistle & W.M.Woham,"*Control problems in a temporal logic frame work*",Int.J. Control,vol.44,pp.943-976,1986.
[7]P.J.Ramadage & W.M.Wonham,"*Supervisory c- ontrol of a class of discrete event process*",SIAM J. Contr. Optimiz.,vol.25,no.1, pp.206-23-,1987.
[8]Z.Manna & A.Pnueli,"*Verification of concurrent programes:A Temporal proof system*", Foundations of Computer Science IV,Mathem- atical Centre Tracts 159,Mathematish Centrum,Amsterdam,pp.163-255,1983.
[9]J.Y.Lin & D.Ionescu,"*Output feedback control for a class of nondeterministic discreteevent systems*", ACC, San diego, California,May 23-25,1990.