

# 이산사건모델들간의 통신을 위한 동기화 가능성 분석

홍기정\* · 김탁곤\*

## Analysis of Synchronization Feasibility for Communicating Discrete Event Models

Ki Jung Hong · Tag Gon Kim

### Abstract

이산사건 시스템을 기술하는데는 다양한 방법론들이 있으며 event의 관점에서 본다면, event loss 방법론과 event lossless 방법론이 있다. event loss 방법론의 경우 때때로 원치 않는 event loss가 발생하여 시스템을 deadlock의 상태로 이끌 수가 있으며, 이러한 것을 피하기 위해 본 논문은 event loss 방식의 동기식 방법론인 DEVS형식론으로 기술된 모델을 본 논문에서 제안하는 TSRG란 방법을 이용하여 event loss의 유무를 해석할 수 있는 기법을 소개하고자 한다. 소개된 해석 기법은 linear programming을 이용하여 해를 구할 수 있으며, DEVS형식론으로 기술된 이산사건 모델들간의 event loss의 유무를 해석함으로써 서로간의 통신 동기화의 가능성 분석을 가능하도록 한다.

## I. 서론

이산사건 시스템을 기술하는데는 다양한 방법론들이 존재한다. 이러한 방법론들간의 차이는 이산사건 시스템들간의 통신은 동기식 혹은 비동기식이 있으며, event loss를 허용하는가에 따라서 동기식이더라도 weak synchronization과 strong synchronization등으로 나눌 수 있으며 비동기식이더라도 event loss를 허용하는 방법론도 존재한다.

동기식이면서 event lossless인 방법론에는 Communicating Calculus System(CCS)등이 있으며, 동기식이면서 event lossy인 방법론에는

DEVS형식론이 있으며, 비동기식이면서 event lossy인 것은 Process Algebra계통의 SDL랭귀지가 있으며, 비동기식이면서 event lossless인것에는 Timed Petri-Net방법론이 있다.[1][2][3][4]

본 논문은 실제 사용되어지는 이산사건 시스템 기술 방법론중 동기식이면서 event loss를 허용하는 DEVS형식론을 기반으로 기술된 모델에 event loss의 발생을 사전에 해석할 수 있는 방법론을 제안하고자 하며, 제안된 방법론을 통해 원치 않는 event loss의 발생으로 인해 우발적인 시스템의 deadlock을 방지할 수 있다.

다음 2장에서는 DEVS기반의 동기화 가능성 분석에 대해서 다루고, 3장에서는 가벼운 예를 들어서 2장의 내용을 설명한다.

\* 한국과학기술원 전자전산학과 시스템 모델링 시뮬레이션 연구실

## II. DEVS기반 동기화 가능성 분석

### 2.1 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.

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

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 수학적 정의는 다음과 같다.

### 2.2 Timed State Reachability Graph

Timed state reachability graph(TSRG)는 이산사건 시스템의 상태 천이를 그래피론으로 표시할 수 있도록 되어있는 도구로써 다음과 같이 표현된다.

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

$$TSRG = \langle X, Y, N, T, E \rangle$$

X: 입력사건집합

Y: 출력사건집합

N: 상태집합 : Node

T:  $N \rightarrow \mathfrak{R}_{(0,\infty)}^+ \times \mathfrak{R}_{(0,\infty)}^+$ : 시간함수

E:  $N \times (X \cup Y) \times Boolean \times N$ : Edge

Boolean: CONTINUE

DEVS모델을 TSRG모델로 변환하는 방법은 DEVS atomic모델기준으로 설명한다면 다음과 같다.

$$TSRG.X = AM.X$$

$$TSRG.Y = AM.Y$$

$$TSRG.N = AM.S$$

$$\begin{aligned}
 TSRG.T &= \{s \rightarrow (0, ta(s)) \mid \forall s \in S \wedge \delta_{int}(s) \neq \phi \\
 &\quad \wedge (\exists x \in X, \delta_{ext}(s, 0, x) \neq \phi)\}
 \end{aligned}$$

$$\begin{aligned}
 \cup \{s \rightarrow (ta(s), ta(s)) \mid \forall s \in S \wedge \delta_{int}(s) \neq \phi \\
 \quad \wedge (\exists x \in X, \delta_{ext}(s, 0, x) = \phi)\}
 \end{aligned}$$

$$\begin{aligned}
 \cup \{s \rightarrow (0, \infty) \mid \forall s \in S \wedge \delta_{int}(s) = \phi \\
 \quad \wedge (\exists x \in X, \delta_{ext}(s, 0, x) \neq \phi)\}
 \end{aligned}$$

$$TSRG.E = \{(s, \lambda(s), false, \delta_{int}(s)) \mid \forall s \in S \wedge \delta_{int}(s) \neq \phi\}$$

$$\cup \{(s, x, true, \delta_{ext}(s, 0, x)) \mid \forall s \in S$$

$$\wedge (\exists x \in X, \delta_{ext}(s, 0, x) \neq \phi$$

$$\wedge e = ta(\delta_{ext}(s, 0, x)) - ta(\delta_{ext}(s, e, x))\}$$

$$\cup \{(s, x, false, \delta_{ext}(s, 0, x)) \mid \forall s \in S$$

$$\wedge (\exists x \in X, \delta_{ext}(s, 0, x) \neq \phi$$

$$\wedge 0 = ta(\delta_{ext}(s, 0, x)) - ta(\delta_{ext}(s, e, x))\}$$

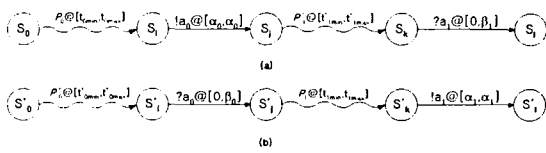
각 atomic모델들을 TSRG로 위의 방법에 따라 변환할 수 있으며 coupled모델들의 경우는 다음과 같은 방법을 통해 각 TSRG들을 합성하여 하나의 TSRG로 변환할 수 있다.

다음의 방법을 이용하여 모델을 변환할 경우 가장 중요하게 고려해야할 사항은 DEVS모델들간의 통신과 TSRG간의 통신방법상의 차이로

문제가 발생할 수 있다는 점이다. DEVS모델들은 weak synchronization을 기본가정으로 모델들이 설계되며 모델간의 event들이 동기화가 이루어질 수도 있거나 혹은 loss될 수 있다는 것을 의미하지만, TSRG모델들은 strong synchronization을 기본가정으로 모델들이 설계되어야 한다. 이것의 의미는 각 event들이 절대로 loss가 발생해서는 안된다는 의미가 된다. 이것을 분석하기 위한 방법을 다음에 제시한다.[6]

$$\begin{aligned}
 AM.X &= CM.X \\
 AM.Y &= CM.Y \\
 AM.S &= \prod_i M_i.S \\
 SM_i &= Select(\{M_i \mid e = M_i.ta(M_i.S)\}) \\
 AM.\delta_{int}(s) &= SM_i.\delta_{int}(s_i) \\
 &\quad \wedge (\forall((\lambda(SM_i.s), M_i.x_j) \in IC, M_i.\delta_{ext}(Q_i, M_i.x_j))) \\
 AM.\delta_{ext}(q, x) &= \forall(x, M_i.x_j) \in EIC, M_i.\delta_{ext}(M_i.x_j) \\
 AM.\lambda(s) &= SM_i.\lambda(SM_i.s_i) \\
 AM.ta(s) &= \min_i(ta(M_i.s_i) - M_i.e)
 \end{aligned}$$

### 2.3 Strong Synchronization Analysis Between Two Models



<그림 1> Synchronization Between Two TSRG

$$\begin{aligned}
 t(P_0) &= \sum_{s_j \in P_0} t(S_j) \\
 t(P'_0) &= \sum_{s_j \in P'_0} t(S'_j) \\
 t_{0min} &\leq t(P_0) \leq t_{0max} \\
 t'_{0min} &\leq t(P'_0) \leq t'_{0max} \\
 \alpha_{0min} &\approx \alpha_0 \leq t(\alpha_0) \leq \alpha_{0max} \approx \alpha_0 \\
 \beta_{0min} &\approx 0 \leq t(\beta_0) \leq \beta_{0max} \approx \beta_0 \\
 t(P'_0) &< t(P_0) + t(\alpha_0) \leq t(P'_0) + t(\beta_0) \\
 \rightarrow 0 &< t(P_0) - t(P'_0) + t(\alpha_0) \leq t(\beta_0)
 \end{aligned}$$

위의 두 그림 사이의 event가 loss되지 않는 조건을 유도해보면 다음과 같다. 일단, 첫 번째 발생한 동기화에 대한 조건식이다. P는 각 노드들의 집합으로써 edge들을 거쳐간다고 가정한다. 즉, 동기화가 발생하지 않은 path라고 한다. 두 번째 발생한 동기화식은 다음과 같다.

$$\begin{aligned}
 t(P_1) &= \sum_{s_j \in P_1} t(S_j) \\
 t_{1min} &\leq t(P_1) \leq t_{1max} \\
 t'_{1min} &\leq t(P'_1) \leq t'_{1max} \\
 \alpha_{1min} &\approx \alpha_1 \leq t(\alpha_1) \leq \alpha_{1max} \approx \alpha_1 \\
 \beta_{1min} &\approx 0 \leq t(\beta_1) \leq \beta_{1max} \approx \beta_1 \\
 t(P'_1) &< t(P_1) + t(\alpha_1) \leq t(P'_1) + t(\beta_1) \\
 \rightarrow 0 &< t(P_1) - t(P'_1) + t(\alpha_1) \leq t(\beta_1)
 \end{aligned}$$

두 번째 이후 i번째의 동기화 식까지 포함하면 다음과 같이 유도됨을 알 수 있다.

$$\begin{aligned}
 t_{imin} &= \sum_{s_j \in P_i} t_{min}(S_j) \leq \sum_{s_j \in P_i} t(S_j) \leq t_{imax} = \sum_{s_j \in P_i} t_{max}(S_j) \\
 t'_{imin} &= \sum_{s_j \in P'_i} t_{min}(S'_j) \leq \sum_{s_j \in P'_i} t(S'_j) \leq t'_{imax} = \sum_{s_j \in P'_i} t_{max}(S'_j) \\
 \alpha_{imin} &\approx \alpha_i \leq t(\alpha_i) \leq \alpha_{imax} \approx \alpha_i \\
 \beta_{imin} &\approx 0 \leq t(\beta_i) \leq \beta_{imax} \approx \beta_i \\
 t(P'_i) &< t(P_i) + t(\alpha_i) \leq t(P'_i) + t(\beta_i) \\
 \rightarrow 0 &< t(P_i) - t(P'_i) + t(\alpha_i) \leq t(\beta_i) \\
 0 &< \sum_{s_j \in P_i} t(S_j) - \sum_{s_j \in P'_i} t(S'_j) + t(\alpha_i) \leq t(\beta_i)
 \end{aligned}$$

이것은 동기화가 발생하는 지점의 수만큼 식이 늘어나게 된다. 이식은 linear programming(LP)으로 해를 얻을 수가 있다.[5]

$$0 < \sum_{s_j \in P_i} t(S_j) - \sum_{s_j \in P'_i} t(S'_j) + t(\alpha_i) \leq t(\beta_i)$$

즉, 위의 조건이 만족되어지는지 검사하는데는 LP를 이용하여 참인지 거짓인지를 판단하면

된다는 것이다. 여기서 제시한 해석 방법을 이용하면 TSRG로 기술된 두 모델사이의 상태천이에 의한 path가 실제로 동기화가 가능한지를 미리 알 수가 있는 것이다. 실제 동기화가 되지 않는다면, 그 path는 서로 통신이 발생할 수 없는 path가 된다는 의미가 된다. 주어진 TSRG에서 도출 가능한 모든 path를 추출하는 방법은 다음과 같은 방식으로 이루어진다.

$$\Gamma(s_0, \nu = \{s_0\}) = \{(e_1, T(s_0)) + \Gamma(s_0, \nu \cup \{s_1\}) \mid s_0, s_2 \in N, \forall e_1 = (s_1, \_, \_, s_2) \in E \wedge s_1 \notin \nu\}$$

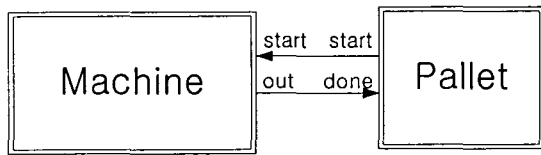
$$\cup \{(e_1, T(s_0)) \mid s_0, s_2 \in N, \forall e_1 = (s_1, \_, \_, s_2) \in E \wedge s_1 \in \nu\}$$

$\Gamma$ 는 초기상태  $s_0$ 에서 출발하여 모든 가능한 cyclic path들을 도출하는 함수이다.

### III. 이산사건 시스템 동기화 분석 예

이장에서는 DEVS모델에서부터 TSRG모델의 변환과 TSRG모델들간의 통신동기화 가능성을 검증하는 과정을 보인다.

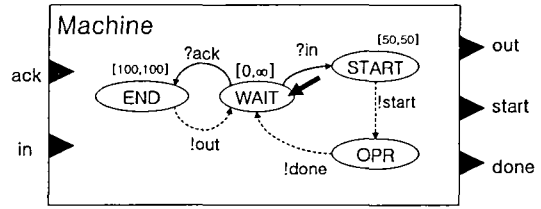
#### 3.1 DEVS모델의 예



〈그림 2〉 Machine/Pallet coupled model

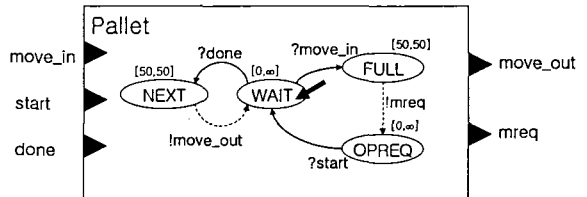
coupled 모델로써 Machine모델과 Pallet모델이 다음과 같이 통신을 하고 있으며 start/start, out/done의 내부 커플링을 하게 되어있다.

Machine모델은 다음과 같이 기술된다.



〈그림 3〉 Machine atomic model

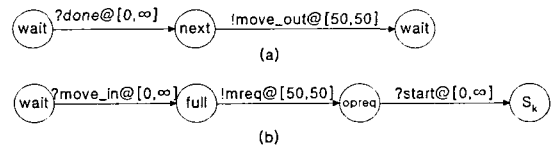
Pallet모델의 기술은 다음과 같다.



〈그림 4〉 Pallet atomic model

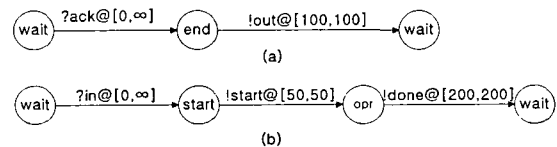
Machine과 Pallet들간에 coupling되지 않은 event들은 EOC/EIC에 coupling되어져 있다고 가정한다.

#### 3.2 Machine/Pallet coupled 모델의 TSRG 변환



〈그림 5〉 Pallet모델의 TSRG Path

Machine과 Pallet의 기술로부터 추출되어지는 path들은 다음과 같다.



〈그림 6〉 Machine모델의 TSRG Path

위의 그림에서 알 수 있듯이 Pallet의 (a)와 Machine의 (a)는 통신이 가능하지만, Pallet의 (a)와 Machine의 (b)는 서로 동시에 발생할 수 없는 path이다. 동시에 발생한다면 strong synchronization의 조건을 만족할 수가 없게 된다. 그리고 마찬가지로 Pallet의 (b)는 Machine의 (b)와 서로 통신할 수 있지만 Machine의 (a)와는 동시에 발생할 수 없는 path가 된다. 이유는 각 패스에 들어있는 event들 중에 machine과 pallet의 내부coupling정보에 의한 event loss의 가능성이다.

### 3.3 Machine/Pallet간 통신동기화 가능성 검사

Pallet과 Machine의 (a)path간의 통신동기화 가능성 검사를 위한 조건들이 다음의 식들로 도출이 된다.

첫 번째 step의 식의 의미는 초기상태에서 동기화조건을 의미하고, 두 번째 step의 식의 의미는 동기화 된 이후 다시 동기화 되는 조건을 의미한다.

$$\begin{aligned}
 &0 < t(ack) \\
 &t(out) = 100 \\
 &0 < t(done) \\
 &t(move\_out) = 50 \\
 &1st, step : 0 < t(ack) + t(out) \leq \max(t(done)) \\
 &2nd, step : 0 < t(ack) - t(move\_out) + t(out) \leq \max(t(done))
 \end{aligned}$$

위의 식에서 첫 번째와 두 번째 조건은 항상 참이 됨을 알 수가 있다. 이 조건이 참이 된다는 의미는 두 path간의 통신에서 event loss가 없다는 의미를 뜻하게 된다.

같은 방식으로 machine (b)path와 pallet (b)path간의 통신동기화 식은 다음과 같이 도출이 된다.

첫 번째 조건은 move\_in이라는 event가 in보

다는 먼저 발생할 경우 항상 참이라는 의미고 이것은 초기 상태의 조건이다. 그리고, 두 번째 조건에서 move\_in은 in보다는 200time unit먼저 발생할 경우 항상 참이라는 의미가 되며 이것은 loop를 형성할 때의 조건, 즉 steady state에서 move\_in이 in보다 200time unit먼저 발생할 경우에만 machine/pallet의 (b)path는 event loss 없이 통신할 수 있다는 의미이다.

$$\begin{aligned}
 &0 < t(in) \\
 &t(!start) = 50 \\
 &t(done) = 200 \\
 &0 < t(move\_in) \\
 &t(mreq) = 50 \\
 &0 < t(?start) \\
 &1st, step : 0 < t(in) - t(move\_in) \\
 &\quad - t(mreq) + t(!start) \leq \max(t(?start)) \\
 &\rightarrow t(move\_in) < t(in) \\
 &2nd, step : 0 < t(in) + t(done) - t(move\_in) \\
 &\quad - t(mreq) + t(!start) \leq \max(t(?start)) \\
 &\rightarrow t(move\_in) < t(in) + 200
 \end{aligned}$$

## IV. 결론

지금까지 이산사건 시스템의 모델들간 event lossless 동기화에 관한 조건과 해석을 할 수 있는 방법을 소개하였으며 이러한 해석이 가능하도록 TSRG를 소개하였다. event lossless동기화를 위한 좀더 복잡한 식들은 일반적인 LP문제로 해를 구할 수 있다.

## 참고문헌

- [1] Robin Milner, *Communication and Concurrency*, Prentice Hall, 1989.
- [2] B.P Zeigler, H.Praehofer and T.G.Kim, *Theory of Modeling and Simulation, 2nd*

- ed. Academic Press, 2000.
- [3] *ITU-T Z.100 SDL Programming Language*, ITU Standard, 1993.
- [4] James L. Peterson, *PETRI NET THEORY AND THE MODELING OF SYSTEMS*, Prentice Hall, 1981.
- [5] Christos H. Papadimitriou and Kenneth Steiglitz, *Combinatorial Optimization*, Dover Publications, 1982.
- [6] Ki Jung Hong and Tag Gon Kim, "Discrete Event Model Verification Using System Morphism", *Proc of IEEE SMC Conference*, p3057-3062, 2001.