DBpia

論 文

ATSN을 이용한 Ada Tasking 실행 시간 복잡도 표현에 관한 연구

正會員 李 秉 馥* 正會員 劉 哲 中* 正會員 金 容 成* 正會員 張 玉 培*

A Study on Representation of Ada Tasking Execution Time Complexity using ATSN

Byong Bog Lee*, Cheol Jung Yoo*, Yong Sung Kim*, Ok Bae Chang* *Regular Members*

要 約

Marked Petri Net(MPN) 모델은 최대 병행 활성화 랑데뷰 수에 따른 통신 복잡도를 분석할 수 있게 해준다. 그러나 이 모델은 시간과 확률 속성(Time and Probability characteristics)을 명시할 수 없음으로 통신 복잡도의 새로운 개념인 실행 시간 복잡도를 분석할 수 없다.

본 논문에서는 이러한 실행 시간 복잡도를 효율적으로 분석하기 위해 MPN에 새로운 제약 조건인 net 절감 법칙, 실행 시간, 그리고 확률 조건을 도입한 Ada Tasking Structure Nets(ATSN)을 제안한다. 끝으로 ATSN 모델을 이용하여 통신 복잡도의 분석 효과를 보인다.

ABSTRACT

Marked Petri Net(MPN) is used to analyze communication complexity with respect to the maximum number of concurrently active rendezvous. But, because it cannot represent timed & probability characteristics explicitly, it is not used to analyze the new execution time of complexity with respect to the communication.

Thus, to effectively analyze that execution time complexity, we propose Ada Tasking Structure Nets(ATSN) introduced restrictive conditions : net reduction rules, execution time, and probability. Finally, we show the powers of analyze of communication complexity with ATSN.

I. 서 론

소프트웨어 시스템이 복잡·대형화 됨에 따라 소프트웨어 위기를 해결하기 위한 여러가지 소프트웨어 공학적 접근 방법론이 연구되고 있다. 이러한 연구 중 하드웨어 발전과 더불어 주목받고 있는 분산형 소프트웨어(distributed software) 개념이 전문가 시스템(expert system), 내장형 소프트웨어(embedded software), 병행 개발(parallel development) 등의 분야에 도입되면서 신뢰성있는 소프트웨어 설계를 위한 복잡도 척도(complexity measures)에 대한 연구

*全北大學校 電子計算學科
Dept. of Computer Science, Chonbuk National Univ.
論文番號 : 93-70

www.dbpia.co.kr

들이 이루어지고 있다. 이러한 연구 중 S.M.Shatz가 Ada Tasking 통신 복잡도(Communication Complexity)를 측정하기 위한 추상화 도구로서 MPN (Marked Petri Net)[SHAT88b]을 소개한 이후 McCabe의 Cyclomatic Number[McCa76]에 관한 MPN 분석 방법이 F.D. Paoli에 의해 연구 발표되었다. 특히, S.M.Shatz[SHAT88b]와 Flavio De Paoli [FLAV90]가 이용한 MPN은 최대 병행 활성화 랑데뷰 수에 따른 통신 복잡도와 제어 흐름 복잡도(Control Flow Complexity)의 한 종류로 볼 수 있는 McCabe의 Cyclomatic Number를 효율적으로 모델링하여 분석할 수 있게 해준다.

그러나, MPN은 시간 개념을 명시할 수 없으므로 통신 복잡도의 새로운 개념인 Ada Tasking 랑데뷰 소요 시간의 상대적인 실행 시간 복잡도(이후, 실행 시간 복잡도라 한다)를 분석할 수 없다. 따라서, 이러한 통신 복잡도와 최대 병행 활성화 랑데뷰 수를 효율적으로 분석·측정하기 위해 MPN에 net 절감 법칙과 실행 시간 그리고 확률 개념을 도입한 ATSN(Ada Tasking Structure Net)을 제안하고 복잡도 측정이 용이하도록 ATSN으로 부터 TRG(Timed Reachability Graph)와 DG(Decision Graph)를 생성한다. 그리고 제안한 ATSN과 DG는 Shatz가 제안한 MPN과 톱니 그래프[SHAT88b]와는 구분되는 효율적인 복잡도 측정 모델임을 보인다.

Ⅱ. MPN과 Ada Tasking 통신 복잡도

이 장에서는 시간을 고려하지 않은 MPN(Marked Petri Net)과 Shatz가 제안한 통신 복잡도 모델에 대해 논한다.

2.1 MPN

MPN의 구조 C는 다음과 같이 플래이스의 유한 집합 P, 트랜지션의 유한 집합 T, 그리고 아크의 유한 집합 A로 A는 입력 아크와 출력 아크로 구성된다. 즉,

$C = (P, T, A, \mu_0)$

여기서, $P = \{p_1, \cdots, p_n\}$ 플래이스의 유한 집합
$T = \{t_1, \cdots, t_m\}$ 트랜지션의 유한 집합
$A = \{P \times T\} \cup \{T \times P\}$ 방향을 갖는 아크의 유한 집합
$\mu_0 = \{\mu_1, \cdots, \mu_n\}$ 초기 마킹 집합
이다[PETE81, FLAV90, SHAT88a, SHAT88b].

프로세스는 조건(condition)과 사건(event)및 그들간의 관계를 서술하는 규칙(rule)들로 구성된다고 생각할 수 있으며 MPN 그래프에서 이들 조건과 사건을 각각 플래이스(원으로 표기)와 트랜지션(선분으로 표기)으로 나타낸다. MPN의 일반적인 특성은,

1) 플래이스와 트랜지션 그리고 트랜지션과 플래이스로의 연결은 유향 아크로 표기하며 실행 상태를 나타내는 토큰은 MPN의 플래이스에 검은점으로 표기한다.
2) 트랜지션은 자신에게로 입력되는 모든 플래이스가 토큰을 보유하고 있어야 점화될 수 있으며 트랜지션이 점화되면 자신의 각 입력 플래이스로부터 토큰을 제거하여 출력 플래이스로 이동시킨다.
3) MPN의 모든 플래이스에 존재하는 토큰들을 벡터로 표기한 것을 마킹이라 하며, 마킹 μ는 $\mu : P \rightarrow N$, $\mu(P_i) = \mu_i$로 정의되는 함수이다. 여기서, N은 음이아닌 정수의 집합이고 μ_i는 플래이스 P_i에 존재하는 토큰의 개수를 의미한다.

마킹 μ는 $|P| = n$이고 각각의 μ_i가 음이아닌 정수이라면 n-벡터 $\mu = \{\mu_1, \cdots, \mu_n\}$으로 표기한다. 또한 MPN 그래프에서 모든 플래이스에 대해 하나 이하의 토큰이 존재하면 safeness라고 부른다. 이러한 경우에 마킹 집합의 요소는 0 또는 1로만 표현된다. 또한, 토큰의 위치와 개수는 트랜지션의 점화(firing)에 의해 바뀌며 이는 MPN의 상태를 변형시키는 역할을 한다. 마킹 μ인 MPN에서 트랜지션 $t_i \in T$가 점화되어 마킹 μ'에 도달한다면, $\mu \xrightarrow{t_i} \mu$: 라 표기하고 μ : 는 μ로 부터 "도달 가능(reachable)하다"고 한다. 이러한 도달 가능성 집합 사이의 관계는 추이(transitive) 관계를 갖는다.

따라서 MPN 구조 C에서 초기 마킹 μ_0로 부터 "도달 가능"한 모든 마킹들의 집합을 $R(\mu_0)$로 표기하고 마킹 μ_0에 대한 도달 가능성 집합(reachability set)이라고 한다.

2.2 S.M.Shatz의 모델

S.M.Shatz[SHAT88b]는 Ada Tasking의 복잡도를 평가하기 위해 새로운 계산 모델을 제안했다. 모델의 기본 원리는 서로 다른 가중값(weighted val-

www.dbpia.co.kr

ue)을 갖는 지역 복잡도(Local Complexity)와 통신 복잡도(Communication Complexity)를 조합하는 방법이었으나 가중값에 대한 정의는 상수가 아닌 임의의 변수를 사용했다.

여기서, 지역 복잡도(LC)는 Halstead의 Software Science[HALS77] 등과 같은 기존의 척도로 계산할 수 있다고 했으며 통신 복잡도(CC)는 MPN 그래프의 마킹 도달성 집합으로 부터 유도했다. 이러한 유도는 하나의 랑데뷰가 시작(동기화)될 때 토큰의 개수가 전체 타스크(task) 수보다 하나 증가하고 랑데뷰가 종료하면 전체 타스크 개수와 일치하는 MPN의 safeness한 성질을 이용한 것이다. 즉, 도달성 집합 중 최대 토큰을 갖는 마킹 벡터는 랑데뷰 수와 관련이 있으므로 최대 병행 활성화 랑데뷰 수는 최대 토큰 수에서 전체 타스크 수를 뺀 값이며, 랑데뷰 관계를 톱니 그래프로 표기함으로써 병행 활성화 랑데뷰 수를 쉽게 계산할 수 있다.

Ⅲ. 실행 시간 복잡도 표현을 위한 ATSN

Ada Tasking 실행 시간 복잡도를 효율적으로 모델링하고 분석하기 위해 MPN에 시간과 확률 속성을 도입한 ATSN 모델을 정의한다.

[정의 1] ATSN(Ada Tasking Structure Net) 구조

$ATSN = (T, P, A, Dt, Pr, \mu_0)$

$P = \{p_1, \cdots, p_n\}$ 플래이스의 유한 집합
$T = \{t_1, \cdots, t_m\}$ 트랜지션의 유한 집합
$A = \{P \times T\} \cup \{T \times P\}$ 방향을 갖는 아크의 유한 집합
$Dt = \{dt(t_1), \cdots, dt(t_m)\}$ 트랜지션에 대한 실행 시간 집합
$Pr = \{pr_1, \cdots, pr_m\}$ 트랜지션에 대한 확률 집합
$\mu_0 = \{\mu_1, \cdots, \mu_n\}$ 초기 마킹 집합
$\mu : P \rightarrow N$(단, $\mu(p_i) = \mu_i$, $i = 1, 2, \cdots, n$, N은 음이 아닌 정수)

여기서, ATSN의 각 트랜지션(t_i)에 대한 실행 시간($td(t_i)$ 또는 td)은 시작 시점과 종료 시점을 각각 $interval_{begin}$과 $interval_{end}$로 표기하는 time interval로 정의하며, 트랜지션의 점화는 늦어도 $interval_{end}$ 시점 까지는 종료하여야 한다. 만일 그렇지 못한 경우에는 정체 현상(∞ 시간)을 갖는 교착 상태(deadlock)가 된다.

만일 트랜지션 점화가 성공한다면 트랜지션(t_i)의 입력 플래이스들로 부터 토큰의 제거는 $tp(interval_{begin})$ 시점에 발생하고, 출력 플래이스들에게로의 토큰 첨가는 $tp(interval_{end})$ 시점에서 발생한다. 그리고 실행 시간 조건은 어떤 수식 또는 상수 값 등으로 정의할 수 있으며 트랜지션의 확률은 특정 점화 조건 또는 같은 입력 플래이스에서의 트랜지션에 대한 경합 상태와 관련이 있기때문에 그 값은 직접 주어지거나 시간 조건과 관련하여 주어진다. 이 확률은 경합 플래이스 p에서 p의 출력 트랜지션이 강한 경합 관계이면 $\sum_{i=1}^{m} Pr_i = 1$를 가지며 그 밖의 트랜지션에 대해서는 $pr = 1$이 성립한다.

그리고 E_c와 E_e는 각각 랑데뷰 시작 트랜지션과 종료 트랜지션을 의미하며 마킹 집합은 ATSN의 실행 상태로 나타낸다. 또한, 트랜지션에 대한 제약 조건은 Event, Time point, Constant-time delay, 경합 상태, 그리고 트랜지션에 대한 확률 등과 관련한다.

첫째, Event 관계는 병행 시스템에서 동일 프로세스 내에 event 집합의 원소 a, b, c, $d \in E$가 존재한다고 가정 할 때, a가 b보다 선행("happened before")하면 $a \rightarrow b$로 표기한다[KIM92].

[정의 2] Event 관계

1. 순 차 관 계 : 동일 프로세스의 a, $b \in E$일 때, a가 b보다 선행하면 $a \rightarrow b$로 표기한다.
2. 통 신 관 계 : 서로 다른 프로세스의 event 집합의 원소인 a, b가 있을 때, a가 전송 메세지이고 b가 수신 메세지 이면 $a \rightarrow b$로 표기한다.
3. 병 렬 관 계 : 서로 다른 프로세스의 event 집합의 원소인 a, b가 있을 때, not($a \rightarrow b$) and not($b \rightarrow a$)이면 $a \| b$로 표기한다.
4. 동 기 화 관 계 : 서로 다른 프로세스의 event 집합의 원소인 a, b가 있을 때, ($a \rightarrow b$) and ($b \rightarrow c$) and ($a \| b$)이면 $a \|_c b$로 표기한다.
5. 추 이 관 계 : ($a \rightarrow b$) and ($b \rightarrow c$)이면 $a \rightarrow c$가 성립한다.
6. 동 등 관 계 : ($a \rightarrow b$) and ($b \rightarrow a$)이면 $a \equiv b$가 성립한다.

www.dbpia.co.kr

7. 경 합 관 계 : c가 a, b를 점화가능 상태로 만들 때, (c→a) and (c→b) and (a||b)이면 a?b로 표기한다.
8. 교착상태관계 : (a$||_c$b) and {c$||_d$a, c$||_d$b}이면 a*d로 표기한다.

둘째, Time point 관계는 a, b, c, d∈E가 time point tp(a), tp(b), tp(c)에서 동작하려면 다음을 만족해야한다.

[성질 1] Time point 관계
1. 순차, 통신 관계 : a→b이면 {tp(a) < tp(b)}
2. 병 렬 관 계 : a||b이면 {tp(a) ≠ tp(b), tp(a) = tp(b)}
3. 동 기 화 관 계 : a$||_c$b이면 tp(c) = max{tp(a), tp(b)}
4. 추 이 관 계 : tp(a) < tp(b)이고 tp(b) < tp(c)이면 {tp(a) < tp(c)}
5. 동 등 관 계 : a≡b이면 {tp(a) = tp(b)}
6. 경 합 관 계 : a?b이면 (tp(c) < tp(a)) and (tp(c) < tp(b))
7. 교착상태관계 : a*b이면 tp(c) < (tp(d) = ∞)

여기서, time point는 [정의 2]에 정의된 event 점화시에 플래이스로의 토큰 이동 결장 순간을 의미함으로,

1) 순차, 통신, 추이 관계의 경우 : event의 점화 우선 순위는 time point로 표현할 수 있기 때문에 선행 event는 후행 event보다 time point값이 작다.
2) 병렬 관계의 경우 : 서로 다른 프로세스의 event가 병행적으로 점화함으로 서로 다른 time point 값을 갖거나 동일한 값을 갖을 수 있다.
3) 동등 관계의 경우 : event의 점화 우선 순위가 동일함으로 각각의 time point 값은 같다.
4) 경합 관계의 경우 : [정의 2]의 경합 event 관계에 따라 event 점화 우선 순위는 c가 a와 b보다 선행함으로서 c의 time point 값은 a와 b보다 반드시 작고, a와 b의 time point 값은 서로 다르다.
5) 교착 상태의 경우 : [정의 2]에 정의된 교착 상태의 event 관계에서 a, b, c는 통신 관계를 갖음으로 time point 값은 임의의 정수값 갖는다. 그러나 a와 b의 점화에 의해 a와 b의 입력 플래이스에는 토큰이 존재하지 않기 때문에 d의 점화는 불가능한 상태인 ∞값을 갖는다.

세째, Interval 관계는 X가 time interval이라할 때, tp(X_{begin})은 X의 시작 time point이고 tp(X_{end})는 X의 종료 time point가 된다. 그리고 트랜지션의 실행 시간 td(X)는,
td(X) = tp(X_{end}) − tp(X_{begin})로 정의된다.

[성질 2] 트랜지션 실행 시간 관계
1. 일 치 관 계 : td(A) = td(B) ⇔ {tp(A_{begin}) = tp(B_{begin})} and {tp(A_{end}) = tp(B_{end})}
2. 선 행 관 계 : td(A)[td(B) ⇔ {tp(A_{end}) < tp(B_{end})}
3. 연 속 관 계 : td(A) ↕ td(B) ⇔ {tp(A_{end}) = tp(B_{begin})}
4. 동시시작관계 : td(A) ↑ td(B) ⇔ {tp(A_{begin}) = tp(B_{begin})} and {(tp(A_{end} > tp(B_{end})), (tp(A_{end}) < tp(B_{end}))}
5. 내 포 관 계 : td(A)《td(B) ⇔ {tp(A_{begin}) > tp(B_{begin})} and {(tp(A_{end}) < tp(B_{end}))}
6. 동시종료관계 : td(A) ↓ td(B) ⇔ {(tp(A_{begin}) > tp(B_{begin})), (tp(A_{begin} < tp(B_{begin}))} and {tp(A_{end}) = tp(B_{end})}
7. 경 합 관 계 : td(A) ↔ td(B) ⇔ {(tp(A_{begin}) = tp(B_{begin})), (tp(A_{begin}) ≠ tp(B_{begin}))} and {((tp(A_{end}) < tp(B_{end})), (tp(A_{end}) > tp(B_{end}))}
8. 교착상태관계 : td(A) ⇔ {(tp(A_{begin}) ≠ ∞)} and {(tp(A_{end}) = ∞)}

여기서, A와 B를 time interval이라 할 때, A와 B의 트랜지션 실행 시간 time point의 관계는,

1) 일치, 선행 관계의 경우 : 일치 관계는 A와 B의 시작 time point 값과 종료 time point 값이 각각 일치하는 경우이며, 선행 관계는 A와 B의 종료 time point 값에 의해 결정됨으로 선행 time interval은 후행 time interval의 종료 time point보다 작은 값을 갖는 경우이다.
2) 연속 관계의 경우 : 선행 time interval의 트랜지션 실행 시간의 종료 time point와 동일 값을 갖는 임의의 time interval의 시작 time point가 존재하는 경우이다.
3) 동시 시작, 동시 종료 관계의 경우 : 동시 시작 관계는 A와 B의 시작 time point 값이 같고 종

www.dbpia.co.kr

료 time point 값이 서로 다른 경우이며, 동시 종료 관계는 A와 B의 시작 time point 값은 서로 다르고 종료 time point 값은 같은 경우이다.

4) 내포 관계의 경우 : A와 B 중 time interval의 선행 시작 time point가 후행 time point 값을 갖는 경우이다.
5) 경합 관계의 경우 : [정의 2][성질 1]의 경합 관계 특성에 따라 A와 B의 time point 값과는 상관없이 종료 time point 값이 작은 interval의 트랜지션이 선택적으로 점화하는 경우이다.
6) 교착 상태 관계의 경우 : [정의 2][성질 1]의 교착 상태 특성에 따라 A와 B의 time point 중 어느 하나가 ∞ 값을 갖음으로 인해 트랜지션의 점화가 불가능한 경우이다.

네째, 경합 상태는 플래이스 P_i에 대한 출력 트랜지션이 $t_1, \cdots, t_k \in T$라고 할 때, 강한 경합 상태(tightly conflict status)가 되려면 [성질 3], 느슨한 경합 상태(loosely conflict status)가 되려면 [성질 4]을 모두 만족해야만 한다.

[성질 3] 강한 경합 상태

1. $\{\mu(P_i) \leq |O(P_i)|\}$ (단, $|O(P_i)|$는 플래이스 P_i의 출력 트랜지션 수이고, $\mu(P_i)$는 플래이스 P_i에 존재하는 토큰의 수이다)
2. 단일 랑데뷰 내에서 경합 상태를 갖는 경우
3. 교착 상태가 아닌 경우

[성질 4] 느슨한 경합 상태

1. $\{\mu(P_i) \leq |O(P_i)|\}$ (단, $|O(P_i)|$는 플래이스 P_i의 출력 트랜지션 수이고, $\mu(P_i)$는 플래이스 P_i에 존재하는 토큰의 수이다)
2. 서로 다른 랑데뷰가 경합 상태를 갖는 경우
3. 교착 상태가 아닌 경우

여기서, ATSN의 트랜지션에 대한 시멘틱 해석에 의해 경합, 무경합, 그리고 교착 상태로 분류한다. 경합 상태는 트랜지션의 부분 집합 $\{t_i\} \subset T_c$에 속하지 않는 다른 트랜지션 $\{t_j\} \subset T_c$의 점화를 불가능하게 하는 경우이며 이 때 트랜지션 $T_c \subset T$를 경합 트랜지션 집합이라 부르며, 경합 상태가 단일 랑데뷰 내에서 일어나는가 아니면 서로 다른 랑데뷰에 의해 일어나는가에 따라 강한 경합 상태와 느슨한 경합 상태로 구분한다. 무경합 상태는 마크 상태 μ에 대하여 경합 상태인 트랜지션의 부분 집합이 존재하지 않는 경우이며, 교착 상태는 트랜지션의 확률이 0이고 실행 시간이 ∞이므로 점화를 행하지 못하는 경우이다.

다섯째, 트랜지션에 대한 확률은 강한 경합 상태의 경우, 느슨한 경합 또는 무경합 상태의 경우, 그리고 교착 상태의 경우로 분류한다(단, $t_1, \cdots, t_k \in T$).

[성질 5] 트랜지션 확률

1. 강한 경합 상태의 경우 : 베이지안 정리

$$Pr_i = \frac{Pr(s_i) \cdot Pr(s/t_i)}{\sum_{j=1}^{k} Pr(s_j) \cdot Pr(s/t_j)}$$

2. 느슨한 경합 또는 무경합 상태의 경우

$Pr_i = 1$

3. 교착 상태의 경우

$Pr_i = 0$

Pr_i : t_j의 트랜지션 확률 (단, $j = 1, 2, \cdots, K$)
$Pr(s_j)$: t_j가 선택될 확률 (단, $j = 1, 2, \cdots, K$)
$Pr(s/t_j)$: t_j가 성공할 확률 (단, $j = 1, 2, \cdots, K$)

ATSN 모델에서 트랜지션에 대한 확률 조건은 세 가지로 분류할 수 있으며 확률 값은 [정의 2][성질 1, 2]의 논리적 시간 환경을 고려하여 상수값으로 표기한다.

1) 강한 경합 상태의 경우 : $Pr(t_i)$가 선택될 확률 "1 / 경합 트랜지션의 수"과 논리적 시간 환경으로 부터 경합 트랜지션 t_i가 성공할 확률($Pr(s/t_i)$)을 계산하여 트랜지션 t_i의 확률 $Pr_i = Pr(t_i/s)$을 계산할 수 있다. 이와같은 계산 방법은 트랜지션이 성공한다는 조건 하에서 성공의 원인이 트랜지션 t_i일 경우에 해당함으로 베이지안 정리를 이용한다.
2) 느슨한 경합 또는 무경합 상태의 경우 : 경합 트랜지션의 특수한 경우로써 트랜지션이 점화 가능 조건만을 만족하면 점화할 수 있으므로 트랜지션에 대한 확률 조건(Pr_i)은 1이 된다.
3) 교착 상태의 경우 : 트랜지션에 대한 실행 시간이 ∞이고 확률 조건(Pr_i)이 0인 특수한 경우이다.

www.dbpia.co.kr

Ⅳ. ATSN 모델링

Net의 분석을 위해 도달성 트리와 행렬 방정식을 이용하는 사례 연구가 있었으나 이 방법은 복잡한 도달 상태와 부정확한 정보를 초래하기 때문에 정보를 변형시키지 않는 범위 내에서 net 절감 법칙을 적용하여 분석을 단순화하는 연구가 있었다[SHAT90a][KIM92]. 본 논문에서는 이러한 절감 법칙[SHAT90a]과 시간 조건 [정의 2][성질 1, 2] 그리고 확률 조건 [성질 3, 4, 5]을 도입하여 net의 결정성 능력과 분석 능력을 향상시킨 문장 구조의 ATSN 모델에 대해 설명한다.

4.1 Ada Tasking 문장 구조의 ATSN 모델링

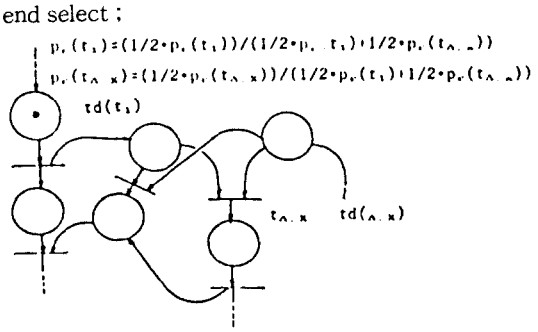
Ada Tasking의 실행 시간 복잡도를 계산하기 위해서는 랑데뷰와 관련된 문장을 고려해야만한다. 랑데뷰 관련 문장은 entry_call과 selective_wait 문장 구조로 구성된다.

4.1.1 entry_call 문장 구조

[conditional-entry-call, CEC]는 entry_call 문에 해당하는 트랜지션(tA.X)이 즉각적으로 점화할 수 없으면 랑데뷰의 활성화는 취소되고 else 문(statement)이 실행(t1)된다. 따라서, CEC는 강한 경합 관계(t1과 tA.X)를 갖는다(그림 1(a)).

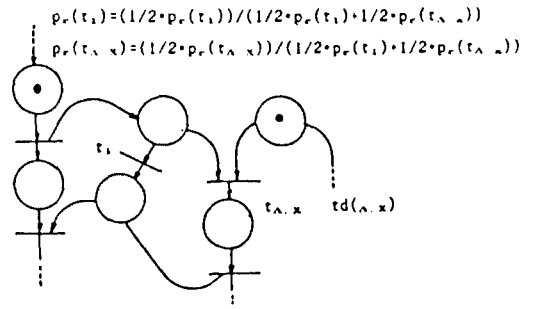
[timed-entry-call, TEC]는 delay 문에 선언된 시간(t)동안 랑데뷰가 활성화(tA.X)되지 않으면 delay 다음 문장에 해당하는 트랜지션(t1)이 점화한다. 따라서, TEC는 강한 경합 관계(t1과 tA.X)를 갖는다(그림 1(b)).

```
  ...
select
  entry A.X(parameter) ;
else
  ...
end select ;
```



(a)conditional-entry-call

```
  ...
select
  entry A.X(parameter) ;
or
  delay t ;
  ...
end select ;
```



(b)timed-entry-call

(그림 1) Entry_call 문장 구조에 대한 ATSN 모델
(Fig. 1) ATSN models for Entry_call structures

4.1.2 selective_wait 문장 구조

[non-select, NS]는 서로 독립적인 accept A~end A과 accept B~end B를 갖는다. 따라서, 트랜지션 tA와 tB는 무경합 관계를 갖는다(그림 2(a)).

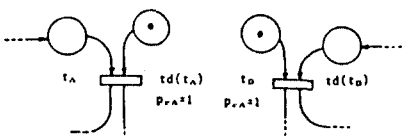
[selective-wait-normal, SWN]은 호출하는 task 환경과 accept 문의 조건에 따라 트랜지션의 점화가 결정되기 때문에 트랜지션 tA와 tB는 느슨한 경합 관계를 갖는다(그림 2(b)).

[selective-wait-guard, SWG]는 guard 조건을 갖는 select 문으로써 호출 task의 환경과 when 절의 조건(G1,G2)에 따라 트랜지션이 점화되기 때문에 트랜지션 tA와 tB는 느슨한 경합 관계를 갖는다(그림 2(c)).

[selective-wait-delay, SWD]에서 accept B~end B가 선택되려면 delay 문에 선언된 시간(t) 후에 실행된다. 따라서, 트랜지션 tA과 tC는 느슨한 경합 관계를 갖는다(그림 2(d)).

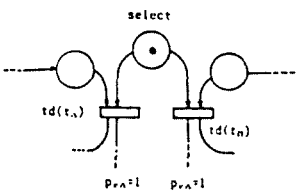
```
accept A do
  ...
end A ;

accept B do
  ...
end B ;
```


www.dbpia.co.kr



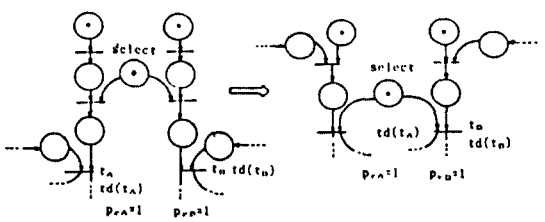
(a) non-select

```
select
  accept A do
    ...
  end A ;
or
  accept B do
    ...
  end B ;
end select ;
```



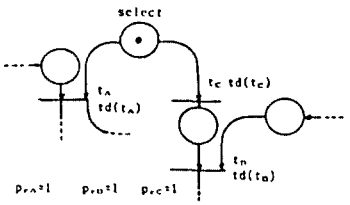
(b) selective-wait-normal

```
select
  when G1 ⇒
  accept A do
    ...
  end A ;
or
  when G1 ⇒
  accept B do
    ...
  end B ;
end select ;
```



(c) selective-wait-guard

```
select
  accept A do
    ...
  end A ;
or
  delay t ;
  accept B do
    ...
  end B ;
end select ;
```



(d)selective-wait-delay

(그림 2) Selective_wait 문장 구조에 대한 ATSN 모델
(Fig. 2) ATSN models for Selective_wait structures

V. Ada Tasking 실행 시간 복잡도 분석

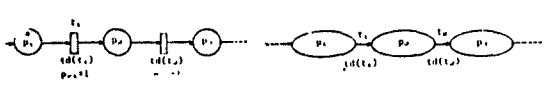
정의된 ATSN을 이용하여 Ada Tasking 실행 시간 복잡도를 분석하기 위해 시간 도달 가능성 그래프(Time Reachability Graph ; TRG)와 결정 그래프(Decision Graph ; DG)를 유도한다. 그리고 이러한 그래프로 부터 통신에 관련된 복잡도를 계산한다.

5.1 ATSN의 시간 도달 가능성 그래프

TRG는 정점(타원형)과 아크(점화 트랜지션) 그리고 확률로 구성되며, ATSN의 초기 상태를 그래프의 시작 정점으로 하여 도달 가능한 다음 상태를 계산함으로써 만들어진다. TRG의 구조는,

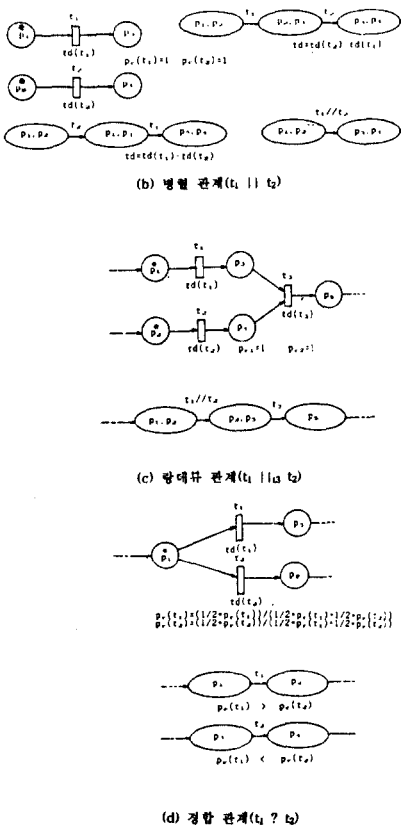
TRG = (V, E, Pr)
여기서, V = 토큰 벡터를 표기한 타원형
E = 점화 트랜지션
Pr = 트랜지션의 확률
이다.

(그림 3)의 정점에 표기된 번호는 트랜지션(E)의 점화에 의한 토큰 도달 상태로써 ATSN의 플래이스들이다.



(a) 순차 관계($t_1 \rightarrow t_2$)

www.dbpia.co.kr



(그림 3) ATSN의 시간 도달 가능성 그래프
(Fig. 3) Time Reachability graphs of ATSN

여기서, TRG의 점화 순서는 시간 환경[정의 2][성질 1, 2]과 확률[성질 3, 4, 5]에 따라 배타적 점화 순서와 동시적 공존 점화 순서로 분류되며, 이 분류 방법은 [DAVI90]의해석에 따른다. 즉, 두 개의 트랜지션을 t_1, t_2라 할 때 $tp(t_1) < tp(t_2)$와 $tp(t_1) > tp(t_2)$는 배타적 점화 순서를 갖으며 $tp(t_1) || tp(t_2)$는 동시적 공존 점화 순서를 갖는다. 그리고 (그림 3 (d))와 같이 경합 상태를 갖는 경우의 점화 순서는 트랜지션 t_1, t_2에 대한 확률값에 의해 순서가 결정된다.

5.2 결정 그래프 생성

ATSN의 랑데뷰 관련 정점만을 고려한 시간 도달성 그래프로부터 중요한 요소만을 모아 DG를 만든다. DG의 구조는,

$$DG = (V, L, D, Pr)$$

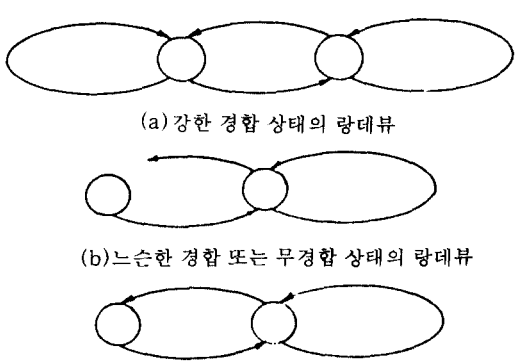
여기서, $V = \{T \mid E_c \cup E_e\}$
T = 트랜지션의 유한 집합
L = 랑데뷰 정점의 시작 트랜지션(E_c)으로부터 종료 트랜지션(E_e)까지의 랑데뷰 경로를 아크로 표현함
D = L을 구성하는데 필요한 소요 시간의 합
Pr = L이 트래버스될 확률

이다.

DG는 랑데뷰 시작 정점(E_c)에서부터 종료 정점(E_e)으로의 연결은 사이클(cycle)을 갖는 방향성 아크(L)로 표기하고, 이 아크를 트래버스(traverse)하는데 소요되는 시간(D)과 확률(Pr)을 DG내에 표기한다.

DG는 통신의 형태에 따라 (그림 4)에서 처럼 강한 경합, 느슨한 경합 또는 무경합, 그리고 교착 상태를 갖는 랑데뷰로 구분한다.

1) 강한 경합 상태를 갖는 랑데뷰 : E_c와 E_e 사이의 트랜지션들이 강한 경합 관계이면 두 개의 서로 다른 사이클(cycle) 아크를 갖는 DG의 정점이 존재한다(그림 4(a)).
2) 느슨한 경합 또는 무경합 상태를 갖는 랑데뷰 : E_c와 E_e 사이의 트랜지션들이 느슨한 경합 또는 무경합이면 하나의 정점에 대해 하나의 사이클 아크만이 존재한다(그림 4(b)).
3) 교착 상태를 갖는 랑데뷰 : E_c로 부터 E_e로의 연결은 사이클을 갖지않는(acycle) 아크로 표기된다(그림 4(c)).



(a) 강한 경합 상태의 랑데뷰
(b) 느슨한 경합 또는 무경합 상태의 랑데뷰
(c) 교착 상태의 랑데뷰

(그림 4) 결정 그래프
(Fig. 4) Decision graphs

www.dbpia.co.kr

여기서, 중첩된 랑데뷰는 [성질 3]의 내포 관계를 갖는 경우로서, 결정 그래프의 외부 정점의 시작 트랜지션(E_c)으로 부터 시작하는 아크 선상에 위치한 내부 정점은 외부 정점에 대한 종료 트랜지션보다 먼저 내부 정점의 종료 트랜지션으로 사이클을 완료해야 함으로, 내부 정점의 아크가 외부 정점에 대한 아크보다 먼저 완성된다.

5.3 ATSN의 실행 시간 복잡도 분석 방법

절감 법칙을 적용하지 않은 시간 도달 가능성 그래프를 분석하면 실행 시간 복잡도를 얻을 수 있으나 이러한 방법은 도달 가능성 상태 공간의 복잡성 문제를 발생시키기 때문에 간단한 분석 방법이 필요하다. 이러한 목적을 위해 본 논문에서는 Shatz가 제안한 톱니 그래프를 향상시킨 새로운 모형의 DG를 이용한다. 그리고 DG를 이용하여 통신 복잡도를 계산한다.

5.3.1 최대 병행 활성화 랑데뷰 수에 따른 통신 복잡도

Shatz는 랑데뷰 event와 활성화 랑데뷰 수를 각각 x-축과 y-축으로하는 톱니 그래프[SHAT88b]를 생성하여 최대 병행 활성화 랑데뷰 수를 얻을 수 있었지만, DG의 정점 수는 랑데뷰 호출에 따른 중첩된 정도를 나타냄으로 최대 병행 활성화 랑데뷰 수를 다음과 같이 계산한다.

최대 병행 활성화 랑데뷰 수
$=\max$(각 랑데뷰에 대한 DG의 정점 수)

5.3.2 실행 시간 복잡도

실행 시간 복잡도를 계산하려면 전체 랑데뷰의 평균 소요 시간에 대한 랑데뷰 소요 시간의 비율을 알아야 한다. 랑데뷰 소요 시간은 DG의 아크를 트래버스하는데 소요되는 시간과 트랜지션 확률을 고려하여 얻을 수 있다.

① 강한 경합 상태를 갖는 랑데뷰의 소요 시간 계산 (Rt_i)

$Rt_i = t_i + \sum(tc_i \times Pr_i + tc_j \times Pr_j)$

(단, t_i와 t_j가 강한 경합 관계인 경우)

② 느슨한 경합 또는 무경합 상태를 갖는 랑데뷰의 소요 시간 계산(Rt_i)

$Rt_i = t_i'$

③ 전체 랑데뷰의 평균 소요 시간 계산(ART)

$ART = \sum Rt_i / n$

④ 각 랑데뷰에 대한 실행 시간 복잡도 계산(RETC)

$RETC_i = Rt_i / ART$

⑤ 랑데뷰에 대한 최대 실행 시간 복잡도 계산(MRETC)

$MRETC = \max(RETC_1, RETC_2, \cdots, RETC_n)$

tc_i, tc_j : i번째 랑데뷰에서 경합 확률의 영향을 받는 소요 시간

Pr_i, Pr_j : i번째 랑데뷰가 갖는 경합 확률

t_i : i번째 랑데뷰에서 경합 트랜지션 이전 까지의 소요 시간

t_i' : i번째 랑데뷰의 소요 시간(단, 강한 경합 관계가 아닌 경우)

Rt_i : i번째 랑데뷰 소요 시간

ATSN 모델의 랑데뷰 소요 시간은 두 가지로 분류하여 계산한다.

1) 강한 경합 상태를 갖는 경우 : 경합 트랜지션의 확률 조건에 따라 랑데뷰 소요 시간이 다르게 계산될 수 있기 때문에 랑데뷰 시작 트랜지션(E_c)으로 부터 종료 트랜지션(E_e)까지의 평균 소요 시간(즉, 기대값 = 확률 × 소요시간)을 계산한다.

2) 느슨한 경합 또는 무경합 상태를 갖는 경우 : 이러한 경우는 트랜지션의 선택이 결정적이기 때문에 트랜지션 E_c로 부터 E_e까지의 소요 시간으로 계산한다.

전체 랑데뷰의 평균 소요 시간은 Rt로 부터 산출된 시간과 전체 랑데뷰의 개수로써 계산할 수 있으며, 각 랑데뷰에 대한 실행 시간 복잡도는 ART에 대한 각 랑데뷰의 소요 시간의 비율로 계산하고, 랑데뷰에 대한 최대 실행 시간 복잡도는 RETC의 계산 값 중 가장 큰 값이 된다.

특히, 랑데뷰에 대한 실행 시간 복잡도는 유지 보수의 가능성 정도를 예측하는 정보로 사용되며 이러한 실행 시간 복잡도의 값 중에서 최대 실행 시간 복잡도는 우선 순위가 가장 높은 유지 보수 영역을 선정하는 기초 자료로 사용된다.

Ⅵ. 모의 실험

제안한 모델의 성능을 평가하기 위해 Shatz의 Gas Station 프로그램[SHAT90a]에 강한 경합 관계를 갖는 entry_call 문장을 첨가한다.

여기서, 프로그램의 코드 번호는 관련 문장을 참조

www.dbpia.co.kr

하기 위하여 사용하였으며 지연 시간(time delay 문에 사용된 시간)과 실행 시간은 트랜지션에 "td"로 표기하였다. 그리고 트랜지션(직사각형)에 대한 dt는 net 절감 법칙을 적용할때 제거된 각 트랜지션의 실행 시간의 합이다.

〈사례 연구 프로그램〉The Extended Gas Station

```
1 task body Customer is
2 begin
3 loop
4   select
5     Operator.Prepay ;
6   or
7     delay 3 ;
8     Pump.Start ;
9     Pump.Finish ;
10    accept Change ;
11 end select ;
12 end loop ;
13 end Customer ;
```

```
14 task body Pump is
15 begin
16 loop
17   accept Activate ;
18   accept Start ;
19   accept Finish ;
20     Operator.Charge ;
21   end Finish ;
22 end loop ;
23 end Pump ;
```

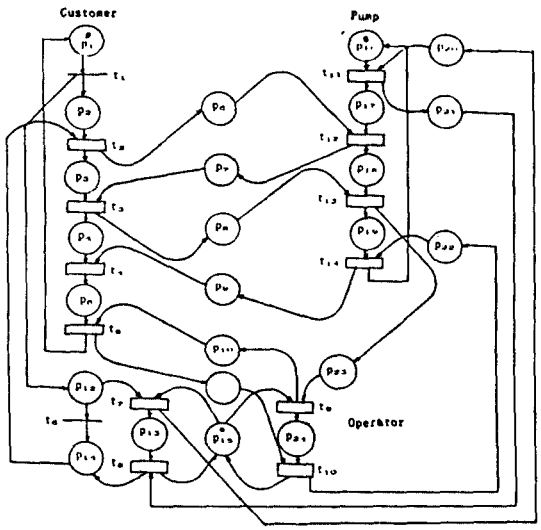
```
24 task body Operator is
25 begin
26 loop
27 select
28   accept Prepay do
29     Pump.Activate ;
30   end Prepay ;
31 or
32   accept Charge do
33     Customer.Change ;
34   end Charge ;
35 end select ;
36 end loop
37 end Operator
```

6.1 절감 법칙을 적용한 ATSN 모델

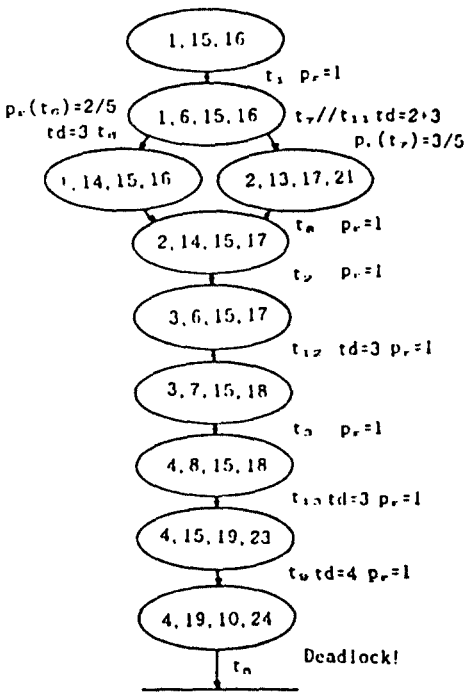
ATSN의 분석을 단순화하기 위해 Shatz가 제안한 net 절감 법칙을 이용하며 이 중에서도 법칙3(Serial-Fusion of Transitions), 법칙4(Parallel Redundant Places), 법칙6(Redundant Accept-Place), 법칙7(Redundant ex-Place)을 적용하되 net의 중요한 성질은 보존한다. 단, 랑데뷰 관련 트랜지션 중에서 E_c와 E_e는 결정 그래프(DG)를 생성하는데 반드시 필요함으로 제거하지 않는다. 편의상 ATSN 모델의 시간 및 확률 속성은 생략하고 시간 도달 가능성 그래프에서 표기하도록 한다.

6.2 ATSN의 시간 도달 가능성 그래프

시간 도달 가능성 그래프는 정점과 아크 그리고 아크에 대한 레이블(실행 시간, 경합 확률)로 구성된(그림 6).



(그림 5) ATSN 모델
(Fig. 5) ATSN model



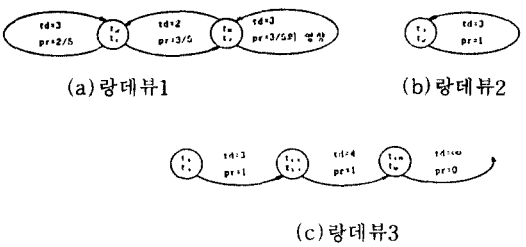
(그림 6) 시간 도달 가능성 그래프(TRG)
(Fig. 6) Time reachability graph

www.dbpia.co.kr

여기서, 강한 경합 관계를 갖는 트랜지션(t_6, t_7)이 점화하면 서로 다른 도달 상태(<1, 14, 15, 16>, <2, 13, 17, 21>)를 갖은 후에 임의의 트랜지션(t_8)에 의해 다음 도달 상태(<2, 14, 15, 17>)는 동일하게 된다. 그리고 프로그램의 교착 상태는 트랜지션의 점화 순서 $t_1 \rightarrow t_6 \rightarrow t_8 \rightarrow t_2 \rightarrow t_{12} \rightarrow t_3 \rightarrow t_{13} \rightarrow t_9 \rightarrow \cdots$ 또는 $t_1 \rightarrow t_7 // t_{11} \rightarrow t_8 \rightarrow t_2 \rightarrow t_{12} \rightarrow t_3 \rightarrow t_{13} \rightarrow t_9 \rightarrow \cdots$ 이후에 발생함을 알 수 있다.

6.3 결정 그래프

실행 시간 복잡도를 계산하기 위하여 TRG로 부터 결정 그래프(DG)를 생성하면 (그림 7)과 같다.



(a) 랑데뷰1 (b) 랑데뷰2

(c) 랑데뷰3

(그림 7) 결정 그래프
(Fig. 7) Decision graph

(그림 7)(a)의 경우는 랑데뷰 시작($E_c = t_1$)과 종료($E_e = t_2$) 사이에 강한 경합 상태를 갖는 트랜지션(t_6, t_7)이 존재하며, ($E_c = t_7$)와 ($E_e = t_8$) 사이에 있는 모든 트랜지션은 이 경합 확률(3/5)의 영향을 받는다. 그리고 (b)는 ($E_c = t_2$)와 ($E_e = t_3$) 사이에 경합 트랜지션이 존재하지 않으며, (c)은 ($E_c = t_9$)와 ($E_e = t_{10}$) 사이에 존재하는 트랜지션(t_5)의 확률이 0(zero)이기 때문에 더 이상 실행을 진행할 수 없다 (교착 상태).

6.4 모의실험 결과

(그림 5, 6, 7)로 부터 Ada Tasking 통신 복잡도를 계산한다.

(1) Ada Tasking 통신 복잡도

- 최대 병행 랑데뷰의 수에 따른 복잡도 계산
  최대 병행 활성화 랑데뷰 수 $= \max(2, 1, 3) = 3$
- 실행 시간 복잡도
  ① 각 랑데뷰에 대한 소요 시간 계산
  $Rt_1 = 3 \times 2/5 + 2 \times 3/5 + 3 \times 3/5 = 4.20$
  $Rt_2 = 3 \times 1 = 3.00$
  $Rt_3 = 3 \times 1 + 4 \times 1 + \infty \times 0 = 7.00$
  ② 전체 랑데뷰의 평균 소요 시간 계산
  $ART = (4.2 + 3.0 + 7.0)/3 = 4.73$
  ③ 각 랑데뷰에 대한 실행 시간 복잡도 계산
  $RETC_1 = 4.20/4.73 = 0.89$
  $RETC_2 = 3.00/4.73 = 0.63$
  $RETC_3 = 7.00/4.73 = 1.48$
  ④ 랑데뷰에 대한 최대 실행 시간 복잡도 계산
  $MRETC = \max(0.89, 0.63, 1.48) = 1.48$

지금까지 살펴본 것처럼 (그림 5, 6, 7)로 부터 최대 병행 활성화 랑데뷰 수와 실행 시간 복잡도 그리고 교착 상태 이전까지의 평균 소요 시간($RT_1 + RT_2 + RT_3 = 14.2$)을 계산할 수 있다. 그리고 최대 병행 활성화 랑데뷰 수로써 통신의 중첩된 정도를 알 수 있으며, 최대 실행 시간 복잡도에 의해 랑데뷰 3이 가장 높은 유지 보수 가능성을 갖음을 알 수 있다. 여기서 유지 보수 가능성의 우선 순위는 Ada Tasking의 통신 부분에 국한된다.

6.5 S.M.Shatz 모델과 ATSN 모델간의 성능 비교

S.M.Shatz 모델과 ATSN 모델의 성능 측정 방법은 개발 대상 시스템을 표현하는 모델링 능력과 모델링된 net 공간으로 부터 정보를 분석할 수 있는 분석 능력을 이용한다.

6.5.1 모델링 능력

S.M.Shatz 모델과 ATSN 모델의 모델링 능력은 net 표현 요소인 제약 조건, net 절감 법칙과 관련된 net 공간 복잡도, 그리고 정보 해석을 지원하는 시맨틱 표현 능력을 갖는다.

(표 1) 모델링 능력의 비교
(Table 1) The comparison for power of modeling

요소 \ 모델	S.M.Shatz 모델	ATSN 모델
제약조건	플래이스, 트랜지션, 토큰	플래이스, 트랜지션, 토큰, 시간, 확률
Net 공간 복잡도	절감 법칙 적용	절감법칙 적용
시맨틱 표현	시맨틱 능력 감소	시맨틱 능력 감소

S.M.Shatz 모델은 조건을 나타내는 플래이스, 점화 가능한 상태에서 점화를 행하는 트랜지션, 그리고 실행 상태를 나타내는 토큰으로 구성된다. ATSN 모

www.dbpia.co.kr

델은 S.M.Shatz 모델에 실행 시간을 나타내는 실행 시간 개념과 트랜지션의 점화 가능성을 나타내는 확률 개념을 도입하여 확장한 새로운 형태의 MPN으로 볼 수 있다.

여기서, ATSN 모델은 모델링 단계에 시간과 확률 개념을 적용함으로서 S.M.Shatz 모델이 표현할 수 없었던 시간 관련 시스템의 표현 능력을 갖는 특징을 가지고 있다. 또한 net 공간의 복잡도를 해결하기 위해 S.M.Shatz 모델에 적용했던 net 절감 법칙을 적용하였으나 두 모델의 시맨틱 표현 능력을 저해하는 요인으로 나타났다.

6.5.2 분석 능력

S.M.Shatz 모델과 net 상태 공간으로 부터 S/W의 혼합적 복잡도의 계산과 최대 병행 활성화 랑데뷰 수의 계산, 그리고 실행 상태의 관련 정보를 유도할 수 있는 분석 도구를 지원한다.

(표 2) 분석 능력의 비교
(Table 2) The comparison for power of analaysis

요소 \ 모델	S.M.Shatz 모델	ATSN 모델
분석도구	TRG, 톱니 그래프	TRG, DG
추론 정보	실행 상태	실행 상태, 시간 관련 정보

S.M.Shatz 모델은 net 상태 공간으로 부터 시스템의 복잡도 계산과 net 공간으로 부터 유도한 톱니 그래프로 부터 최대 병행 활성화 랑데뷰 수를 계산하여 TRG로 부터 실행 상태의 관련 정보들을 얻을 수 있다. ATSN 모델은 S.M.Shatz 모델의 분석 도구로 분석 가능한 정보 뿐만아니라 시간과 확률 제약 조건으로 부터 DG를 생성한 후 DG로 부터 실행 시간 복잡도 등과 같은 시간 관련 정보와 최대 병행 활성화 랑데뷰 수등을 얻을 수 있다.
구로 분석 가능한 정보 뿐만아니라 시간과 확률 제약 조건으로 부터 DG를 생성한 후 DG로 부터 실행 시간 복잡도 등과 같은 시간 관련 정보와 최대 병행 활성화 랑데뷰 수등을 얻을 수 있다.

Ⅶ. 결 론

MPN 모델은 마킹의 도달 상태를 이용하는 최대 병행 활성화 랑데뷰 수에 따른 통신 복잡도는 분석할 수 있으나 이 모델은 시간과 확률 속성(timed & probable characteristics)을 명시할 수 없으므로 이러한 속성을 중요한 요소로하는 실행 시간 복잡도를 분석할 수 없다. 따라서 본 논문에서 제안한 ATSN에 시간과 확률 환경을 정의하여 시간과 확률의 표현 능력을 갖도록 하였으며, ATSN으로 부터 TRG와 DG를 생성하는데 소요되는 분석 비용을 줄이기 위해 S.M.Shatz[SHAT88a]가 제안한 net 절감 법칙을 적용하였으나 생성된 TRG로 부터 복잡도를 측정하려면 점화 순서를 트래버스해야 하는 복잡한 작업이 뒤따르기 때문에 중요한 요소만을 고려하여 결정 그래프(DG)를 유도한 후 이 그래프로부터 최대 병행 활성화 랑데뷰수, 실행 시간 복잡도 등의 통신 복잡도를 계산할 수 있었다.

그리고 시간과 확률의 표현 능력을 갖춘 ATSN 모델과 그에 대한 TRG와 DG의 사례연구로서, 간단한 the extended gas station 문제의 통신 복잡도를 분석·측정함으로써 본 논문에서 제시한 모델들이 실행 시간 복잡도를 표현하는데 효율적임을 보였다. 그 외에도 교착 상태(deadlock), 스케쥴링(scheduling), 그리고 시간과 시스템의 모델링과 분석에 응용될 수 있음을 알았다.

그러나 ATSN 모델의 상태 공간 절감과 분석의 단순화에 지나치게 치중하면 net의 시멘틱 능력을 잃어버릴 수 있는 단점이 있다.

참 고 문 헌

[DAVI90] David A.Watt, Programming Language Concepts and Paradigms, Prentice-Hall, 1990.

[FLAV90] Flavio De Paoli, Sandro Morasca, "Extended Software Complexity Metrics to Concurrent Programs," IEEE, Plazza L. da Vinci 32-20133 MILANO (ITALY), pp.414-419, 1990.

[GRAD87] Grady Booch, Software Components with Ada : Structures, Tools, and Subsystems, Benjamin/Cummings, 1987.

[HALS77] M.H.Halstead, "Element of Software Science," New York : Elsevier North Holland, 1977.

[KIM92] Kim So Yeon, Lee Gang Soo, "A Timed Ada Tasking Net(TATN) Model for Reliability Analysis of Ada

www.dbpia.co.kr

Tasking that involve Delay and Exception Instructions," JCSE:92 Proc. on Software Engineering, Olympic Plaza Hostel, Seoul, Korea, March 1992.

[McCa76] T.J.McCabe, "A Complexity Measure," IEEE Trans. on SE., Vol.2, pp.308-320, Dec. 1976.

[PETE81] J.Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, 1981.

[ROBE85] Rebert G.Clark, Programming in Ada : First Course, Pang-Han, 1985.

[SHAT88a] S.M.Shatz, W.K.Cheng, "A Petri Net Framework for Automated Static Analysis of Ada Tasking Behavior," The Journal of Systems and Software, Vol.8, pp.343-359, 1988.

[SHAT88b] S.M.Shatz, "Towards Complexity Metrics for Ada Tasking," IEEE tran. on Software Engineering, Vol.14, No.8, Aug. 1988.

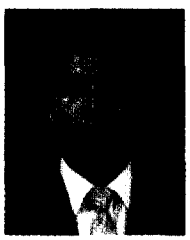
[SHAT90a] S.M.Shatz, S.Tu, T.Murata, "Application of Petri Nets Reduction to Support Ada Tasking Deadlock Detection," Proc. 10'th int'l conf. on Distributed Computing Systems, pp. 96-103, 1990.

[SHAT90b] S.M.Shatz et al., "Design and Implementation of Petri Net based Toolkit for Ada Tasking Analysis," IEEE trans. on Parallel and Distributed Systems, Vol.1, No.4, pp. 424-441, Oct. 1990.

[SOMM89] Ian Sommerville, Software Engineering, Addison Wesley, 1989.

李 秉 馥(Byung Bog Lee) 정회원
1967년 7월 5일생
1991년 2월 : 전북산업대학교 전자계산학과 졸업(이학사)
1991년~현재 : 전북대학교 대학원 전자계산학과 석사과정
※주요관심분야 : 시스템 모델링 및 분석 도구, 분산처리 프로그램

金 容 成(Yong Sung Kim) 정회원
1951년 6월 3일생
1978년 2월 : 고려대학교 수학과 졸업(이학사)
1991년 2월 : 광운대학교 대학원 전자계산학과 졸업(이학박사)
1985년~1986년 : 호서대학교 자연대학 전자계산학과 전임강사
1986년~현재 : 전북대학교 자연과학대학 전자계산학과 부교수
※주요관심분야 : 소프트웨어공학, 컴퓨터 알고리즘, 교육공학 등



劉 哲 中(Cheol Jung Yoo) 정회원
1960년 1월 15일생
1982년 2월 : 전북대학교 전산통계학과 졸업(이학사)
1985년 2월 : 전남대학교 대학원 전산통계학과 졸업(이학석사)
1990년~현재 : 전북대학교 대학원 전자계산학과 박사과정
1982년~1985년 : 전북대학교 전자계산소 근무
1985년~현재 : 기전여자전문대학 전자계산과 부교수
※주요관심분야 : 소프트웨어공학, 전문가 시스템, 컴퓨터 그래픽스, 분산처리 등

張 玉 培(Ok Bae Chang) 정회원
1943년 2월 10일생
1962년 2월 : 고려대학교 수학과 졸업(이학사)
1974년~1980년 : 죠지아 주립대, 오하이오 주립대 박사과정 수료
1988년 : 산타바바라대 박사과정 수료(Ph.D)
1990년~1991년 : 영국 에딘버러대 객원 교수
1980년~현재 : 전북대학교 자연과학대학 전자계산학과 교수
※주요관심분야 : 소프트웨어공학, 전산교육, 인공지능, 수치해석 등

www.dbpia.co.kr