

# E-TCPN을 이용한 CCSR 명세와 스케줄링가능성 분석

최동한<sup>\*</sup>, 박홍복<sup>\*</sup>

<sup>\*</sup>부경대학교

## CCSR Specification and Schedulability Analysis Using E-TCPN

Dong-Han Choi<sup>\*</sup>, Hung-Bog Park<sup>\*</sup>

<sup>\*</sup>Dept. of Computer Science Graduate School of

Pukyong National University

E-mail : dongdong@unicorn.pknu.ac.kr, multiphb@unicorn.pknu.ac.kr

### 요약

CCSR은 CCS에 시간적인 개념이 추가된 실시간 명세 언어로 실시간 프로세스의 정적 분석에 이용되고 있다. CCSR은 프로세스 전이 규칙을 이용하여 프로세스 동작의 관심 부분만 요약하여 분석하기 때문에 전체 태스크(task)의 동작을 분석하는데 적용하기 어렵고, 전이규칙에 따른 상태 변화를 이해하기 어렵다. E-TCPN은 시간 요소가 첨가된 Petri Net의 변형된 형태로 실시간 시스템의 분석과 모델링을 위한 형식적(formal) 방법을 제공한다. 본 논문에서는 CCSR로 표현된 프로세스 동작을 E-TCPN 모델에 적용함으로써 실시간 시스템을 중요한 사건을 중심으로 전체 태스크의 동작 과정을 표현하고 프로세스의 수행 과정을 이해하기 쉽도록 CCSR을 변형하여, E-TCPN 명세 모델에 적용하고, 적용된 E-TCPN으로 스케줄링가능성 분석 알고리즘을 제안하고 구현하였다.

## 1. 서론

실시간 시스템의 정확성은 시스템에서 생성된 결과에 대한 논리적, 시간적인 정확성에 의존하기 때문에 시스템이 정상적인 기능을 수행하기 위해서는 마감시간[1] 이내에 정확한 결과를 산출하여야 한다. 그러므로 실시간 시스템에 주어진 태스크들은 정확한 분석과 이해를 필요로 한다.

실시간 명세 언어 CCSR[6, 10]은 시간적인 개념이 추가된 CCS[6]의 변형된 형태로 프로세스 대수[2]를 이용하여 프로세스 동작의 상호 관련성과 상태 변화를 표현한다.

본 논문에서는 주요 관심 부분만 요약하여 표현하고, 복잡한 전이 규칙의 수행에 따라 다음 상태를 결정하기 때문에 판독하기가 어려운 CCSR로 명시된 실시간 태스크를 시간 개념이 첨가된 TCPN(Time Constraint Petri Net)을[4, 5, 11] 확장한 E-TCPN(Extended Time Constraint Petri Net)을 이용하여 모델링함으로써 전체 태스크의 동작과정을 표현하고, 중요 사건을 중심으로 표현한 실시간 시스템 모델링의 간결성을 판독을 쉽게 해줌으로써 CCSR로 명시된 실시간 태스크의 특성을 보다 쉽게 분석할 수 있도록 하는 CCSR을 E-TCPN으로 적용하기 위한 확장 기법을 제시한다. 또한 적용된 E-TCPN으로 스케줄링가능성을 분석하는 알고리즘을 제안하고 이것을 GUI 환경

에서 Borland사의 C++ Builder 3.0으로 구현하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 실시간 명세 언어 CCSR과 E-TCPN을 소개하고, 3장에서는 CCSR이 E-TCPN에 적용된 모델과 스케줄가능성 분석 알고리즘을 소개하고, 4장에서는 스케줄가능성 분석 알고리즘을 구현 및 분석하고, 5장에서는 본 연구의 결론과 향후 연구 과제에 대해 기술한다.

## 2. CCSR과 E-TCPN

### 2.1 CCSR

CCSR에서 프로세스는 프로세스가 수행하는 명령에 해당하는 동작으로 구성된다. 프로세스의 동작은 프로세스가 수행하는 동작으로 표현된다. 각 동작은 그 동작을 수행하는데 필요한 시간간격  $d$ 를 갖는다. CCSR 프로세스의 구문은 그림 1과 같은 BNF 문법으로 정의된다[6].

$$P ::= \text{nil} \mid \nu^d \mid \delta^d \mid a^d \mid P+Q \mid P;P \mid P\parallel Q \mid \text{sw}(P, d) \mid \text{fw}(P, d) \mid \text{e}(P, d)$$

그림 1 CCSR 프로세스 구문

CCSR은 시간 요소가 첨가된 세 가지 기본 동작  $v^d$ ,  $\delta^d$ ,  $a^d$ 로 구성되고, 프로세스 동작에 대한 +(선택 조합), ;(순차 조합), ||(병렬 조합)도 가능하다. 프로세스가 포함하는 시간 제약과 시간 특성은 세 가지 시간 구성자 sw, fw, e에 의해 구분되고, 역시 마감시간 d를 포함한 시간 요소를 포함하고 있다. 마감시간은 프로세스가 시작된 시간으로부터 상대적으로 계산된다. sw(P, d)는 프로세스 P가 d 단위시간 이내에 시작하여야 하고, fw(P, d)는 프로세스 P가 d 단위시간 이내에 끝나야 한다. e(P, d)는 정확히 d 단위시간 이내에 종료하는 프로세스를 표현하는데 사용된다.

2.2 E-TCPN

시간 간격(duration)을 첨가한 Petri Net의 확장인[3,4] TCPN(Timed Constraints Petri Net)을 확장한 E-TCPN(Extended - TCPN)은 <정의 1>과 같다.

<정의 1>

- $N = (P, T, F, A, D, M)$
- $P : \{P_1, P_2, \dots, P_m\}$ 인 플레이스의 집합
- $T : \{t_1, t_2, \dots, t_n\}$ 인 트랜지션의 집합
- $F : (P, T)$  또는  $(T, P)$  순서쌍에서 가능한 아크(arc)의 집합
- $A : a(n, ex(d), d)$ 의 집합으로 n은 동기화 되어야하는 대상 프로세스의 수, ex(d)는 동기화 처리 시간, d는 동기화가 완전히 처리되고 점화되어야하는 시간 간격
- $D :$  시간 간격(duration) d의 집합
- $M :$  m-vector를 가진 마킹(marking)  $\{M(p_1), \dots, M(p_m)\}$ 의 집합 여기서,  $(M_p)$ 는 플레이스 내 토큰의 수이고,  $M_0$ 는 초기 마킹
- $F_p :$  r(d),  $\delta(d)$ 인 플레이스가 가지는 동작의 집합.
- $F_t :$  fw(P, d)와 a(n, ex(d), d)의 조합인 트랜지션이 가지는 동작의 집합

표 1 CCSR의 프로세스 구문

```
P ::= fw(v2; Pa+Pb, 25);P;
Pa ::= e(a4; (Pc||Pd); v4, 22);
Pb ::= e(a4; v5; v3, 18);
Pc ::= fw(v3; v6; b3, 14);
Pd ::= fw( $\delta^2$ ; v5; b3, 12);
```

표 2 변형된 CCSR

```
P ::= fw(r(2);Pa+Pb, 25);P;
Pa ::= e(fw(a(0,ex(2),3),4);Pc||Pd;r(4),22);
Pb ::= e(fw(a(0,ex(1),2),9);r(5);a(0,ex(0),0);
r(3),18);
Pc ::= e(r(3);a(0,ex(0),0);r(6);a(1,ex(2),5),14);
Pd ::= e( $\delta(2)$ ;a(0,ex(0),0);r(5);a(1,ex(2),5),12);
```

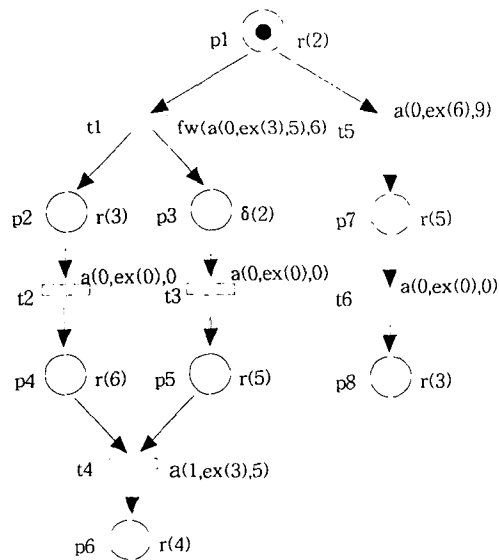


그림 2 E-TCPN에 적용된 전체 프로세스 P

3. E-TCPN의 스케줄가능성 분석

3.1 E-TCPN 적용 모델

CCSR을 E-TCPN에 적용 가능하도록 표 1에 대한 CCSR의 프로세스 구문을 표 2와 같이 변형하여 표현하고 실제 확장된 E-TCPN에 적용하여 모델링한 결과를 그림 2에 나타낸다.

3.2 스케줄가능성 분석

스케줄가능성 분석이란 실시간 시스템에 실시간 프로세스들이 주어진 마감시간을 만족하면서 원하는 결과를 정확히 산출할 수 있는지를 판단하는 것이다.

본 절에서는 실시간 시스템의 표현으로 변형된 E-TCPN 모델을 이용하고, 이 모델을 기반으로 스케줄가능성을 분석함으로써 Petri Net을 이용한 실시간 시스템의 표현과 분석 모두를 제공하는 체계적인 분석 알고리즘을 제안하고, 만약 스케줄

이 불가능한 부분 프로세스가 존재할 경우 스케줄이 불가능한 원인과 위치를 출력하는 알고리즘을 제시한다.

본 논문의 E-TCPN 모델을 이용한 스케줄가능성 분석은 크게 두 단계로 구분할 수 있다.

첫 번째 단계는 전체가 스케줄이 가능하더라도 동기화 객체에서  $ex(d) > D$ (트랜지션의 마감시간)인 경우에는 사실상 스케줄링이 불가능하기 때문에 이러한 조건을 검사하는 알고리즘은 그림 3의 순서도와 같다. 여기서는 모든 동기화 객체에 대해서  $ex(d) > D$ 인 트랜지션이 있는가를 검사하는 것이다. 만약 이러한 트랜지션이 존재 한다면 스케줄링 불가능 메시지를 출력하고, 그렇지 않으면 스케줄링 가능 메시지를 출력한 다음에 전체 스케줄링가능성 분석의 단계로 넘어간다.

두 번째 단계는 전체 프로세스 P에 대한 스케줄 가능성을 분석하는 것이다. 여기서는 각 부분 프로세스가 수행하는 각각의 동작에 대한 누적된 시간과 부분 프로세스의 마감시간을 비교해서 각 부분 프로세스가 스케줄링이 가능하면 수행 경로에 따른 부분 프로세스의 누적된 마감시간과 전체 프로세스에서 부분 프로세스를 제외한 동작의 누적된 시간을 합해서 전체 프로세스의 마감시간과 비교하여 스케줄링이 가능하면 스케줄링 가능성이 가능하다는 메시지를 출력하고, 그렇지 않으면 불가능하다는 메시지를 출력하는 알고리즘을 그림 4의 순서도로 나타내었다. 그림 3의 알고리즘에서 모든 트랜지션이 스케줄링이 가능하고, 그림 4의 알고리즘에서 모든 부분 프로세스의 스케줄링과 전체 프로세스의 스케줄링이 가능할 때 비로소 전체 프로세스가 스케줄링이 가능하다는 것을 의미한다.

알고리즘1

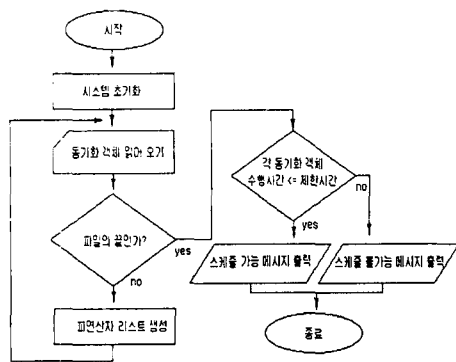


그림 3 각 동기화 객체 스케줄링가능성 분석 흐름도

알고리즘2

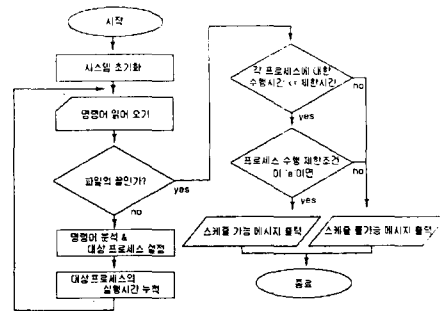


그림 4 전체 스케줄링가능성 분석 흐름도

#### 4. 구현 및 분석

본 장에서는 3장에서 설명한 변형된 CCSR이 적용된 E-TCPN 명세 모델을 기반으로 스케줄가능성을 분석하고, 스케줄이 불가능한 트랜지션이 존재할 경우 스케줄이 불가능한 원인과 위치를 출력해주는 알고리즘의 정확성을 분석하기 위해, 객체 지향 언어인 Borland사의 C++ Builder 3.0을 이용하여 알고리즘을 구현하고 분석하였다.

##### 4.1 구현

실제 프로그램에서 사용된 핵심적인 입력 데이터 저장 구조체는 파일을 오픈한 후 파일의 내용을 읽어서 저장해두는 입력덱 구조체는 표 3과 같고, 플레이스의 모든 정보와 부모 노드와 자식 노드에 대한 정보를 포함하고 있는 플레이스의 정보를 저장하는 구조체는 표 4와 같다. 그리고 트랜지션의 동작이 동기화 동작인지 선택 동작인지를 구별하고 부모 자식의 노드에 대한 정보를 저장하는 트랜지션 구조체는 표 5와 같은 구조로 선언하여 사용한다.

표 3 입력덱 저장 구조체

```

struct InDec_ {
    // 입력덱 저장 구조체
    AnsiString op_code; //명령어 저장
    AnsiString operand[operand_MAX]; //인자저장
    char command[80]; //원시 소스
};
    
```

그밖에 파일 저장 구조체, 시시스템 사용 변수 저장 구조체, 스택 저장 구조체 등이 사용되었다.

표 4 플레이스 정보 저장 구조체

```

struct Place_ {
    // Place 정보 저장 구조체
    AnsiString place_id; //플레이스 명과 id값
    AnsiString action; //동작기호
    int time; //시간
    AnsiString p_link[1]; //부모 객체 링크 리스트
    AnsiString c_link[1]; //자식 객체 링크 리스트
    int p_link_p; //부모 링크 인덱스(수)
    int c_link_p; //자식 링크 인덱스(수)
};
    
```

표 5 트랜지션 정보 저장 구조체

```

struct Trans_ {
    // Trans 정보 저장 구조체
    AnsiString trans_id; //트랜잭션 명과 id값
    bool fw; //fw 유부
    int fw_d; //fw의 제한시간
    bool a; //a의 유부
    int a_n; //a의 입력개수
    int a_ex; //a의 실행시간
    int a_d; //a의 제한시간
    AnsiString p_link[link_MAX]; //부모 객체 링크 리스트
    AnsiString c_link[link_MAX]; //자식 객체 링크 리스트
    int p_link_p; //부모 링크 인덱스(수)
    int c_link_p; //자식 링크 인덱스(수)
};
    
```

아래의 그림 5에서는 프로그램 전체에 관한 시스템 구성을 보여준다. 먼저 변형된 CCSR을 읽어서 구성한 입력력을 각각의 플레이스와 트랜지션에 해당하는 패트리넷 객체로 변환하는 작업을 수행한 후 생성된 객체들에 대해 스케줄가능성 분석을 위해서 두 가지로 분류한 다음 스케줄가능성 분석 단계로 넘어간다.

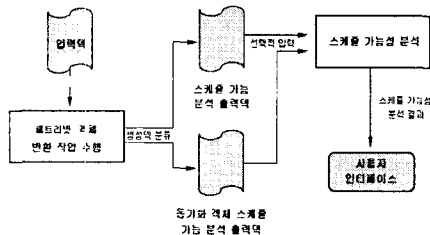


그림 5 전체 시스템 구성도

그림 6의 순서도에서는 각각의 플레이스와 트랜지션에 해당하는 패트리넷 객체로 변환하는 작업을 수행하는 패트리넷 객체 변환 프로그램을 순서도로 나타내고 있다. 각 프로세스에 해당하는 파일의 끝을 만날 때까지 왼쪽의 루프가 수행이 되고 마지막으로 전체 프로세스의 마지막 명령어를 만날 때까지 전체 루프를 수행한다.

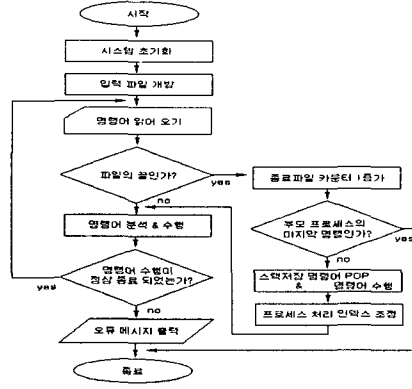


그림 6 객체 변환 처리 흐름도

그림 6의 객체 변환 처리가 끝나고 3장에서 설명된 알고리즘 1과 알고리즘 2를 거치는 과정에서 스케줄링가능성 분석을 완료하게 된다.

4.2 분석

3장에서 E-TCPN에 적용되도록 변형된 CCSR 예제인 그림 2를 구현한 결과를 그림 7에 나타낸다.

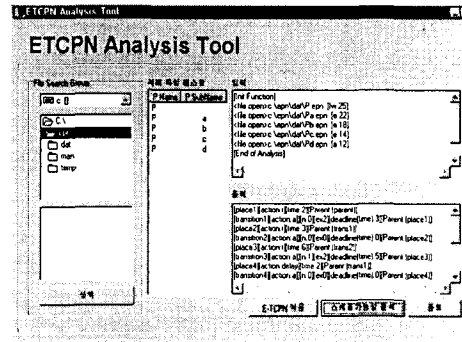


그림 7 메인 윈도우

왼쪽에서 드라이브와 폴더를 선택하고 프로세스들은 처리파일 리스트에 등록을 한 다음에 E-TCPN 확장 버튼을 클릭하면 오른쪽 상단에서 입력상태를 표시하고, 하단에서 적용된 E-TCPN의 플레이스, 트랜지션의 번호와 각각의 동작과 마감시간, 그리고 실행시간과 부모 자식 관계 등에 관한 정보를 표시해준다.

그림 8과 그림 9는 메인 윈도우에서 스케줄 가능성 분석 버튼을 클릭하면 나타나는 스케줄가능성 분석 결과를 보여주는 윈도우이다. 여기서는 두 가지 스케줄가능성을 분석한다. 하나는 동기화 객체에 관한 스케줄가능성을 분석하고 하나는 전체 스케줄가능성을 분석한다. 동기화 객체에 관한 스케줄가능성 분석은 전체가 스케줄이 가능하더라도 동기화 객체에서  $ex(d) > D$ 인 경우를 검사하는 것이다.

그림 8은 스케줄링이 가능한 표 2의 명세를 그대로 입력시킨 것이기 때문에 분석 결과는 스케줄링이 가능한 것으로 판단이 되었고, 그림 9는 프로세스 Pa의 시간을 조정해서 입력으로 주었을 때 스케줄이 불가능하다는 판단을 전체 스케줄링 분석 상태창에서 나타내고 있다.

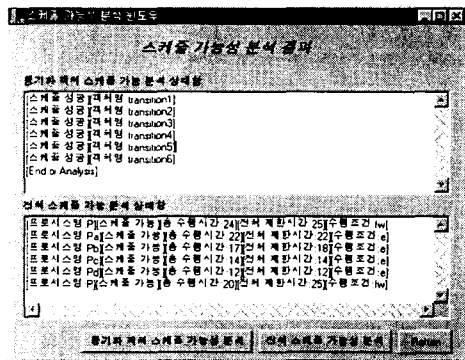


그림 8 스케줄이 가능한 경우

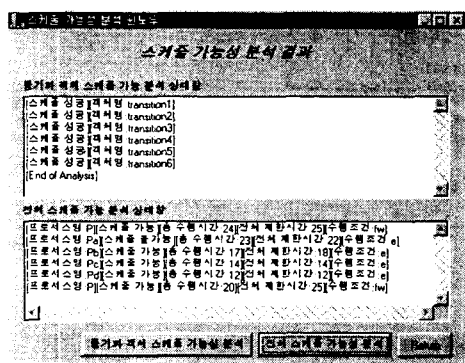


그림 9 스케줄이 불가능한 경우

#### 4.2.1 E-TCPN 적용의 장점

지금까지 제시된 바와 같이 프로세스 대수를 이용한 실시간 시스템은 Petri Net에 시간 요소가 첨가된 TCPN을 확장한 E-TCPN으로 적용하여 이를 기반으로 스케줄가능성을 분석하였다. 실시

간 시스템을 E-TCPN으로 적용함으로써 다음과 같은 장점을 얻을 수 있다.

첫째, CCSR을 이용한 실시간 시스템은 전체 태스크의 모든 동작을 표현하는 것이 아니라 주요 관심 부분만을 요약하여 표현한다. 그러나 Petri Net 모델은 전체 태스크의 동작 과정을 표현함으로써 실시간 시스템의 단계별 실행 동작과 각 단계의 작업 부하를 고려할 수 있다. 이는 작업 부하가 많은 부분 태스크의 실행 시간을 단축함으로써 보다 효율적인 실시간 시스템을 구성할 수 있게 한다.

둘째, CCSR을 이용한 실시간 시스템은 프로세스 전이 규칙에 따라 다음 상태를 결정함으로써 복잡하고 다양한 프로세스 전이 규칙을 수행하여야 한다. 그러나 Petri Net은 사건을 중심으로 프로세스의 상태를 분류함으로써 태스크에 처리되어야 하는 중요 사건을 중심으로 실시간 시스템을 이해할 수 있다.

셋째, 현재의 실시간 시스템은 여러 개의 부분 태스크가 상호 관련되어 수행되고 있다. 이러한 복잡한 실시간 시스템들은 CCSR을 이용하여 표현하기 어렵다. 그러나 Petri Net 모델은 하나의 플레이스나 트랜지션이 하나의 프로세스로 표현되고 있기 때문에 복잡한 실시간 시스템의 상호 동작에 대한 분석을 가능하게 한다.

마지막으로, CCSR을 이용한 실시간 시스템은 프로세스 동작의 정확한 수행 과정이나 수행 과정을 판독하는 것이 쉽지 않다. 그러나 Petri Net의 간결성은 사건을 중심으로 한 표현 방법을 제공함으로써 프로세스의 수행 과정을 판독하는데 용이하다. 또한, 복잡한 실세계 시스템을 모델하고 이를 분석하기 위해 다양한 Petri Net의 형태가 제시되었고, 병행성이 내재된 각종 실시간 시스템의 모델 및 분석에 사용됨으로써 그 우수한 병행성 모델이 입증되었고, 실시간 시스템을 분석하기 위해 Petri Net의 기본 구조에 다양한 구조적 제약 조건과 수행 규칙이 첨가된 TPN(Timed Petri Net)[7], OCPN(Object Composition Petri Net)[8], SPN(Stochastic Petri Net)[9] 등의 다양한 응용이 제시되고 있다.

#### 5. 결론 및 향후 연구

본 논문에서는 프로세스 대수를 이용한 기존의 방법과는 달리 프로세스 대수 표현을 Petri Net에 시간 개념이 첨가된 E-TCPN에 적용할 수 있도록 CCSR의 변형과 TCPN 확장 기법을 제시하고 적용된 E-TCPN으로 스케줄가능성을 분석하고 불가능한 원인과 위치를 출력하는 스케줄가능성 분석을 Borland사의 C++ Builder 3.0으로 구현하였다.

기존의 실시간 태스크의 분석은 프로세스의 관심 부분만 요약하고, 프로세스의 전이 규칙에 따른 상태를 표현함으로써 전체 태스크에 대한 상태를 분석하기 어려웠다. E-TCPN에 적용된 실시

간 태스크의 분석은 전체 태스크의 상태를 표현하고 간결한 표현을 사용함으로써 실시간 시스템의 특징에 대한 분석을 용이하게 하였다.

향후 과제로는 분석 과정에서 발견된 실시간에 스케줄 불가능한 원인을 스케줄 가능하게 명세를 변형하여 스케줄을 가능하게 하는 회복 기법을 찾고 단일 시스템의 환경에서 프로세스 모델링 뿐만아니라 단일 시스템에서의 스케줄 가능성 분석을 분산 실시간 시스템으로 확장하여 적용하고, 프로세스간의 상호 동작과 시간 동작을 시각화하고 프로세스의 동작을 모니터링 함으로써 최적의 수행시간을 평가할 수 있는 실시간 스케줄 가능성 분석기의 설계와 인터프리터 기능을 갖춘 프로세스 계산 지원 시스템의 개발을 목표로 한다.

### 참고 문헌

- [1] A. N. Fredette and R. Cleaveland, "A Generalized Approach to Real-Time Schedulability Analysis," 10th IEEE Workshop on Real-Time Operating Systems and Software, 1993.
- [2] X. Nicollin and J. Sifakis, "An Overview and Synthesis on Timed Process Algebras," Proceedings of the REX workshop on Real-Time: Theory and Practice, LNCS 600, 1991.
- [3] G. Bruno, A. Castella, I. Pavesio and M.P.Pescarmona, "A New Petri Net Based Formalism for Specification, Design and Analysis of Real-time Systems," Proceeding Real-Time Systems Symposium, pp. 294-301, 1993.
- [4] J. E. Coolahan, Jr. and N. Roussopoulos, "Timing Requirement for Time-Driven Dystems Using Augmented Petri Nets", IEEE Trans. Software Eng., vol. SE-9, pp. 603-616, Sept. 1983.
- [5] J. P. Tsai, Yoadong Bi, J. H. Yang and A. W. Srith, "Distributed Real-Time Systems.", A Welly-Interscience publication, 1996.
- [6] R. Milner, Communication and Concurrency, Prentice-Hall, 1989.
- [7] W. M. Zuberek, "Timed Petri Nets and Performance Evaluation of System"Proceedings of the 1998 IEEE International Conferece on System, Man, and Cybernetics- Volume 1, 278-283
- [8] Guan SU, Yu HY, Yang JS "A Prioritized Petri Net Model and its Application in Distriuted Multimedia Systems" IEEE Transaction on Computers, V.47 N.4, , 19980401
- [9] Xie XL, "Perturbation Analysis of Stochastic Petri Nets", IEEE Transactions on Automatic Control, V.43 N.1 , , 19980101.
- [10] 최정률, 김춘배, 박홍복, "프로세스 대수를 이용한 실시간 프로세스의 스케줄 가능성 분석 알고리즘", 한국정보처리학회 '98 추계 학술발표논문집 제5권 제2호, 1998. 10
- [11] 최동한, 서동진, 김춘배, 박홍복. "TCPN 명세 모델링을 위한 CCSR 확장과 적용 기법", 한국 정보처리 학회, 99 춘계 학술발표 논문집 제6권 제1호, pp.1227-1230, 1999. 4