자동화 제어 엔진의 설계 및 구현

서상진**, 류춘렬*, 박지환*
*부경대학교 전자계산학과, **호서전산전문학교

A Design and Implementation of Automation Control Engine

Sangjin Seo**, Chunyeol Ryu*, Jihwan Park*
*Dept. of Computer Science, Pukyong Nat'l Univ., **Dept. of Internet Business, Hoseo Technical Collage

요 약

시스템 자동화를 위한 최근의 소프트웨어 모듈은 각종 외부의 영향에 따라 적절한 대응동작을 이행하지 않는다. 특히 연성 실시간(soft real-time) 동작 제어시 발생하는 동작 마감시간 위반에 따른 피해가 꾸준히 증가하고 있다. 본 논문에서는 실시간 스케줄 가능성 분석을 위해 E-TCPN 모델을 이용하며, 분석 결과를 기반으로 하드웨어 제어를 수행하는 자동화 제어 엔진을 설계 및 구현한다.

1. 서론

자동화 시스템은 각종 제조업 및 제어 분야에서 지향하고 있는 방향이며, 중요한 경쟁력 확보의 수단이 되고 있다. 정부에서도 자동화 산업의 중요성을 인식하여 국립 기술품질원에 자동화 설비 평가 센터 구축을 추진하는 등 많은 노력이 이루어지고 있지만 아직은 연구가 많이 부족한 실정이다[5].

다양한 응용 분야에서 실시간 오동작이나 고장은 환경적, 인적, 물리적으로 막대한 손실을 초래할 수 있기 때문에 이를 최소화하기 위한 전문화된 제어 로직의 개발이 필수 불가결하다[3,4]. 현재 상태를 모니터링 하는 대부분의 실시간 시스템들은 모니터링 관련 모듈이 시스템 코드 레벨에 포함되어 모니터링과 오류에 따른 문제 분석이 매우 어렵다[7].

따라서 본 논문에서는 효율적인 스케줄 가능성 분석 모델인 E-TCPN(Extended- Timed Constraints PetriNet)을 적용한 실시간 자동화 제어엔진을 설계 및 구현한다. 본 논문의 구성은 2장에서 제어동작의 스케줄 가능성 분석을 위한 E-TCPN 모델에 대해 살펴본다. 3장에서는 자동화 제어엔진의 프로토타입을 제시하며, 4장에서는 결론 및 향후 연구과제에 대해서 언급한다.

2. 실시간 스케줄을 위한 E-TCPN

자동화 제어 엔진은 실시간 동작 제어를 위해 스케줄 가능성 분석을 수행한다. 효율적인 분석 모델의 적용[2]을 위해 CCSR[8] 기반의 확장된 패트리 넷인 E-TCPN 모델을 적용한다. E-TCPN 모델은 기존의 스케줄 가능성 분석 모델[1,6]에 비해 프로세스 공간의 명세 구조가 간단하다. 그리고 전체 작업의 동작 과정을 표현되므로 프로세스 공간탐색이 용의하다[8].

2.1 E-TCPN의 정의

CCSR 기반의 확장된 패트리 넷 E-TCPN은 리스트 1과 같이 플래이스와 트랜지션에 대한 최대 및 최소 시간 제약과 시간 간격(duration)을 가진 6-튜블로 정의된다[8].

N={P, T, F, A, D, M}

- N: {P, T, F, A, D, M}인 튜플들의 집합
- P: {P1, P2, ..., Pm}인 플래이스의 집합
- T: {T1, T2, ..., Tn}인 트랜지션의 집합
- F: (P,T) 혹은 (T,P) 순서쌍에서 가능한 아크(arc)의 집합
- A: 단위 시간 D이내에 수행 종료되어야 하는 a(n, ex(d), d) 동기화의 집합
- D: 시간 간격(duration) d의 집합
- M: m-vector를 가진 마킹, 여기서 M_0는 초기 마킹

[리스트 1] E-TCPN의 튜플 정의

2.2 E-TCPN의 명세 구문

프로세스의 작업 처리를 명세하기 위해 확장된 CCSR은 리스트2와 같이 두개의 프로세스 P와 Q에 대한 프로세스 항목으로 표현된다[6].

P:: ni | r(d) | δ(d) | a(n, ex(d), d) | P+Q | P;Q | P||Q | fw(P,d) | e(P,d)

[리스트 2] 확장된 CCSR의 BNF 표현

대문자는 프로세스를 의미하고, 소문자는 프로세스를 구성하는 동작을 표시한다. nil은 프로세스의 종료를 나타내며, d는 수행 제한 시간을 나타내는 단위 시간이다. 그리고 r은 계산시간, δ는 지연시간을 나타낸다. a는 동기화 동작을 나타내며, 동기화 동작 a를 구성하는 첫 번째 인자 n은 동기화 대상 프로세스의 개수를

나타낸다. 두 번째 인자 ex는 동기화 진행 시간을 나타내며, 마지막 인자 d는 전체 동기화 마감시간을 나타낸다. 프로세스 구성자(process constructor)는 프로세스 사이에 기본 동작을 표현하며, +(선택), ;(순차조합), || (병렬조합)이 있다. 시간 구성자(time constructor)는 프로세스가 포함하는 시간제약과 시간 특성을 나타내는데 사용된다. 시간 구성자 sw(start within)는 프로세스 P가 단위시간 d이내에 시작되는 구성자를 나타낸다. fw(finish within) 시간 구성자는 프로세스 P가 단위 시간 d이내에 끝나야 함을 명시하며, e(exactly) 시간 구성자는 정확히 단위 시간 d에 끝나야 됨을 나타낸다.

2.3 실시간 스케줄 가능성 분석

스케줄 가능성 분석이란 실시간 시스템에서 실행되는 프로세스들이 주어진 마감 시간을 만족하면서 원하는 결과를 정확히 산출할 수 있는지를 판단하는 것이다.

본 논문에서는 자동화 제어 엔진의 실시간 시스템 표현을 위해 변형된 E-TCPN 모델을 이용한다. E-TCPN 모델을 기반으로 스케줄 가능성을 분석함으로써 패트리 넷을 이용한 실시간 시스템의 표현과 분석을 수행한다. 만약 스케줄이 불가능한 부분 프로세스가 존재할 경우 스케줄이 불가능한 원인과 위치를 검색한다.

E-TCPN 모델을 이용한 스케줄 가능성 분석을 위해 두번의 처리 단계를 거친다. 첫 번째 처리 단계에서는 각 프로세스들의 스케줄 가능성 분석과 입력 프로세스 전체에 있는 각 트랜지션을 위한 스케줄 가능성을 분석한다. 분석 시 주어진 실시간 시스템 전체의 마감시간은 만족하지만, 부분 프로세스에서 스케줄이 불가능한 경우에 해당되는 프로세스들을 추출한다. 리스트 3의 알고리즘은 이러한 처리 과정을 나타내고 있다.

```
자료 구조 초기화
WHILE(입력 넥이 종료될 때까지)
 트랜지션 객체 적재 및 객체 정보 획득
 순차적으로 구조체의 멤버 변수들의 처리 값 설정
 적재된 트랜지션 객체를 위한 스케줄 가능성 분석
 IF(오류 발생) 오류 메시지 기재
END OF WHILE
```

[리스트 3]각 프로세스의 스케줄 가능성 분석 알고리즘 1

두 번째 처리 단계는 전체 작업 공간에 대해 각 스테이지의 플래이스와 트랜지션을 순차 탐색하여 스케줄 가능성 분석을 수행한다. E-TCPN에 설정된 각 링크 리스트를 탐색하여, 대상 플래이스에서 직전에 수행된 플래이스와 트랜지션의 사용시간의 합과 전체 작업의 마감시간을 비교하여 스케줄 가능한지를 판단한다. 리스트4는 처리과정을 알고리즘으로 나타내고 있다.

```
자료 구조 초기화
WHILE(입력 넥이 종료될 때까지)
 명령어 디코딩
 순차적으로 구조체의 멤버 변수들의 처리 값 설정
 IF(초기 명령어 모드)
  각 프로세스의 제한시간 설정
 ELSE
  SWITCH(명령어 코드)
    CASE "+": 전체 수행 시간 누적
    CASE "||": 두 벡터의 합 연산
    DEFAULT: 전체 수행 시간 누적
  END OF SWITCH
 END IF
 읽기 인덱스 증가
END OF WHILE
FOR(전체 프로세스 수만큼)
 IF(선택프로세스 전체 수행시간<=선택프로세스제한시간)
   시간 구성자가 "e"이면 수행시간과 제한시간이 같아야
   스케줄 가능
   시간 구성자가 "fw"이면 수행시간이 제한시간보다 적
   거나 같으면 스케줄 가능
 ELSE
   스케줄 불가능 판단
 END OF IF
END IF FOR
```

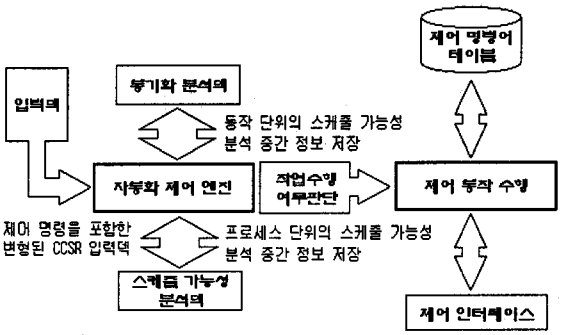
[리스트 4]전체 프로세스 스케줄 가능성 분석 알고리즘 2

3. 자동화 제어 엔진의 설계 및 구현

본 논문의 자동화 제어 엔진은 실시간 스케줄 분석을 통해 신뢰성 있는 하드웨어 제어를 수행하는 독립적인 애플리케이션 객체이다. 본 장에서는 자동화 제어 엔진의 동작 리스트에 대한 실시간 스케줄링과 동작 제어를 위한 제어 엔진을 설계 및 구현한다. 그리고 제어 엔진의 정확한 동작 수행을 검사하기 위해 처리 결과를 분석한다.

3.1 실시간 자동화 제어 엔진의 설계

자동화 제어 엔진은 명세 된 제어 동작을 수행하기 위해 변형된 CCSR 기반의 입력 덱을 읽어서 패트리 넷 객체로 분석 및 변환한다. 분석 과정에서 생성된 중간 정보들은 다음 단계의 스케줄 가능성 분석시 참조된다. 그림1은 자동화 제어 엔진의 전체 수행과정을 위한 처리구성을 나타내고 있다.



[그림1] 자동화 제어 엔진의 구성도

변형된 CCSR로 구성된 입력 덱은 자동화 제어 엔진의 입력으로 사용되기 위해 미리 정의된 입력 포맷으로 명세된다. 각 처리 동작을 나타내는 동작구문은 제어 엔진 내부의 구조체 내에 순차적으로 저장되어 스케줄 가능성 분석시 참조된다. 그림2는 입력 덱의 명령어 포맷을 나타내고 있다.

프로세스 ID	프로세스 이름	계산동작	제어명령 인덱스	동작시간	마감시간

가. 변형된 CCSR 입력 포맷

플래이스 ID	플래이스 이름	동작기호	처리시간	부모객체 리스트	제어명령 인덱스

나. 변환된 플래이스의 출력 포맷

트랜지션 ID	트랜지션 이름	fw 모드	동기화 모드	동기화 입력개수	동기화 시간
동기화 제한시간	부모객체 리스트	부모 링크 전체 인덱스 수			

다. 변환된 트랜지션의 출력 포맷

[그림2] 입력덱의 명령어 포맷

CCSR 입력 명세를 읽어 스케줄 가능성 분석을 위해 패트리 넷 객체로 변환해야 한다. 자동화 제어 엔진의 변형된 CCSR의 패트리 넷 객체 변환 처리 작업이 정상적으로 종료되면, 스케줄 가능성 분석을 수행한다. 패트리 넷의 트랜지션에 대한 스케줄 가능성 분석 결과가 긍정적일 때, 변환된 패트리 넷의 전체 객체 공간에 대한 스케줄 가능성을 분석한다. 전체 프로세스가 스케줄 가능할 때, 자동화 제어 엔진은 제어 동작 수행을 위해 생성된 패트리 넷 객체를 메모리에 적재하여 수행을 시작한다.

3.2 자동화 제어 엔진 구현

자동화 제어 엔진을 동작 테스트하기 위한 입력 덱은 리스트5와 같다.

```
Pa::=e(a^3;(Pc||Pd);r^4,22);
Pb::=fw(a^3;r^5;r^3,18);
Pc::=fw(r^3;r^6;b^3,14);
Pd::=fw(δ^2;r^5;b^3,12);
```

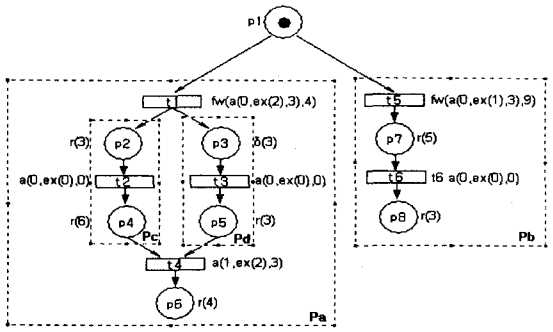
가. CCSR의 프로세스 구문

```
P::=fw(r0(2);Pa+Pb,25);P;
Pa::=e(fw(a(0,ex(2),3),4);Pc||Pd;r1(4),22);
Pb::=fw(fw(a(0,ex(1),2),9);r2(5);a(0,ex(0),0);r3(3),18);
Pc::=fw(r4(3);a(0,ex(0),0);r5(6);a(1,ex(2),3),14);
Pd::=fw(δ(2);a(0,ex(0),0);r6(5);a(1,ex(2),3),12);
```

나. 변형된 CCSR

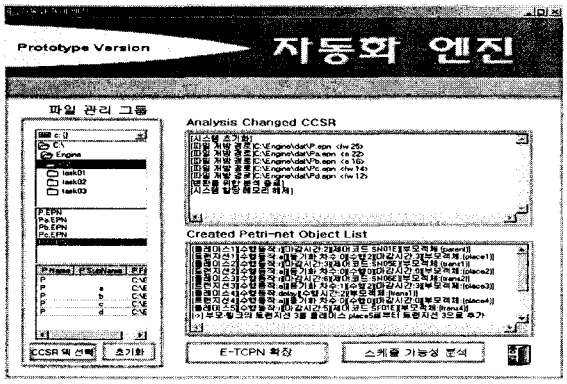
[리스트5] 엔진 수행을 위한 입력덱

입력 덱의 변형된 CCSR의 처리를 위한 프로세스 P는 선택 조합(+)으로 프로세스 Pa와 Pb를 포함하는 주기 프로세스로 표현된다. 병렬 조합(||) 프로세스 구성자를 가지는 Pa는 Pc와 Pd를 포함하는 순차 프로세스이다. 그리고 Pb, Pc, Pd는 모두 순차 프로세스이며, 전체 프로세스 P의 수행 마감 시간은 25 단위 시간이다. 그림3은 입력 덱의 처리 동작을 E-TCPN으로 표현하고 있다.



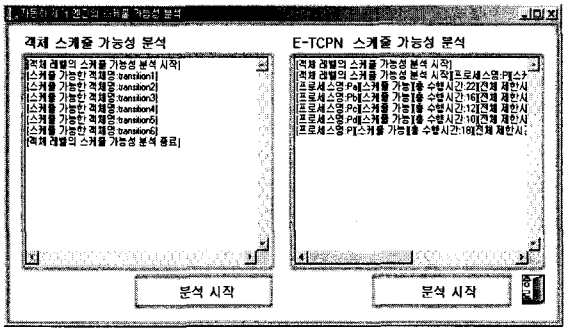
[그림3] E-TCPN으로 표현된 객체 공간

자동화 제어 엔진은 동작 테스트를 위해 변형된 CCSR 입력 덱을 선택적으로 로딩 하여 E-TCPN으로 확장한다. 그림4는 자동화 제어 엔진이 입력 덱 로딩 및 E-TCPN 변환 과정을 보여주고 있다.



[그림4] 자동화 제어 엔진의 시작 화면

변환된 객체에 대한 스케줄 가능성 분석을 위해 자동화 제어 엔진의 "스케줄 가능성 분석" 버튼을 선택하면 스케줄 가능성 분석 결과가 화면에 그림5와 같이 표시된다.



[그림5] 스케줄 가능성 분석 창

3.3 처리 결과 분석

입력 덱의 스케줄 가능성 분석을 위해서 우선 변형된 CCSR 명세를 E-TCPN으로 바꾸어야 한다. 리스트6은 자동화 제어 엔진이 변환한 E-TCPN 객체 공간을 문자열로 나타내고 있다.

```
[플레이스1][수행동작:r][마감시간:2][부모객체:(parent)]
[트랜지션1][수행동작:a][동기화 차수:0][수행2][마감시간:3][부모객체:(place1)]
[플레이스2][수행동작:r][마감시간:3][부모객체:(trans1)]
[트랜지션2][수행동작:a][동기화 차수:0][수행0][마감시간:0][부모객체:(place2)]
[플레이스3][수행동작:r][마감시간:6][부모객체:(trans2)]
[트랜지션3][수행동작:a][동기화 차수:1][수행2][마감시간:3][부모객체:(place3)]
[플레이스4][수행동작:delay][수행시간:2][부모객체:(trans1)]
[트랜지션4][수행동작:a][동기화 차수:0][수행0][마감시간:0][부모객체:(place4)]
[플레이스5][수행동작:r][마감시간:5][부모객체:(trans4)]
[->] 부모-링크의 트랜지션 3을 플레이스 place5로부터 트랜지션 3으로 추가
[플레이스6][수행동작:r][마감시간:4][부모객체:(trans3)]
[트랜지션5][수행동작:a][동기화 차수:0][수행1][마감시간:3][부모객체:(place1)]
[플레이스7][수행동작:r][마감시간:5][부모객체:(trans5)]
[트랜지션6][수행동작:a][동기화 차수:0][수행0][마감시간:0][부모객체:(place7)]
[플레이스8][수행동작:r][마감시간:3][부모객체:(trans6)]
```

[리스트6] 자동화 제어 엔진이 분석한 패트리 넷 객체 공간

확장된 E-TCPN 객체 공간에 대해 대상 프로세스의 트랜지션 객체 동기화 동작에 대한 마감 시간 초과여부를 검사한다. 표1은 변형된 CCSR을 E-TCPN으로 변환하여 스케줄 가능성을 분석한 결과를 요약하고 있다.

[표1]각 프로세스의 스케줄 가능성 분석 결과

프로세스명	패트리넷 객체명	동기화 객체수	수행 시간	마감 시간	스케줄가능 (남은시간)
Pa	t1	0	2	3	가능(1)
Pc	t2	0	0	0	가능(0)
Pc	t3	1	2	3	가능(1)
Pd	t4	0	0	0	가능(0)
Pb	t5	0	1	3	가능(1)
Pb	t6	0	0	0	가능(2)

트랜지션에 대한 스케줄 가능성 분석이 끝나면, 전체 스케줄 가능성 분석을 수행한다. 전체 스케줄 가능성 분석 단계에서는 동기화 객체가 스케줄 가능한 프로세스에 대해서 프로세스 단위의 스케줄 가능성 분석을 수행한다. 표2는 자동화 제어 엔진이 생성한 전체 프로세스들에 대한 스케줄 가능성 분석 결과를 요약하고 있다.

[표2] 전체 프로세스의 스케줄 가능성 분석 결과

프로세스명	스케줄가능성 분석결과	총수행시간	전체마감시간	마감시간 수행조건
P	스케줄 가능	24	25	fw
Pa	스케줄 가능	22	22	e
Pb	스케줄 가능	16	16	e
Pc	스케줄 가능	12	14	fw
Pd	스케줄 가능	10	13	fw

표1,2를 통해 자동화 제어 엔진이 정확하게 동작함을 판단할 수 있다.

4. 결론 및 향후 연구 과제

본 논문에서는 실시간 제어 모니터링을 통해 신뢰성 있는 동작 제어를 위한 자동화 제어 엔진을 설계 및 구현하였다.

기존연구 [7]은 제어 기능이 타이머 기반의 수행 코드로 모듈화 되어 기능의 전문성과 디버깅이 불리하며, 기존연구 [9]는 순차 처리 기반의 모듈이 전체 시스템에 결합되어 기능 확장과 디버깅이 어렵다. 따라서 이 논문에서 제안한 자동화 제어 엔진은 전문화된 객체의 독립성을 보장하여 기능 추가 및 디버깅이 용의하다. 그리고 자원 제어 모니터링 기능은 기존 논문 시스템[7,9]과는 달리 스케줄 가능성 분석을 수행하여 동작 수행을 판단하므로 제한 시간이내에 수행 가능한 경보 시스템과 결함 허용처리를 결정한다. 또 자동화 제어엔진 클래스는 제어와 모니터링에 관련된 기능이 캡슐화 되어 있으므로 재사용성과 시스템 이식이 용이하다. 그러나 입/출력 포맷이 표준화되어 있지 않아 일관성 있는 개발이 어렵다.

향후 연구 과제로는 본 논문의 자동화 제어 엔진을 무선 인터넷 환경에 적합한 경량의 실시간 자동화 제어 엔진으로 이식하는 것이다. 이러한 확장을 통해 분산된 처리 환경에서의 스케줄 가능성 분석 수행은 홈 네트워크와 무선 원격제어 분야에 적용 가능하다.

[참고 문헌]

[1] F.Moller and C. Tofts, "A Temporal Calculus of Communicating System", Proc. of CONCUR '90, LNCS 458, pp. 401-415, 1990

[2] J.E.Coolahan, Jr, and N.Roussopoulos, "Timing Requirement for Time-driven Systems Using Augmented Petrinets", IEEE Trans. Software Eng., vol. SE-9, pp. 603-616, Sept, 1983

[3] Keith A. Butler, Robert Jacob, Bonnie E. John, "Introduction and Overview of Human-Computer Interaction", CHIVAS, April 1994.

[4] Pauli Salo & Leena Thrulin, "Reader's Guide for Connitive Scientists", Stillings et al., 1989.

[5] S.P.Reiss, "PECAN: Program Development System that Support Multiple View", IEEE Trans. on Software Engineering, 13(3):276-285, March 1985.

[6] 김춘배, 최동한, 서동진, 서상진, "분산 실시간 시스템의 스케줄 가능성 분석 및 실시간 프로세스 계산 기술 언어에 관한 연구", 정보통신부 최종 연구 결과보고서, 1999. 8. 26.

[7] 서상진, 박홍식, 박홍복, "Client/Server에 기반한 온실 환경 모니터링에 관한 연구", 한국정보처리학회 제13회 춘계학술대회, Paper-025, 2000. 4.

[8] 최동한, "CCSR Specification Modeling and Schedulability Analysis used E-TCPN", 부경대학교 졸업논문, 1999. 2.

[9] 홍순호, "Automatic Control of Growth Environment for Plant Factory", 서울대학교 박사학위 논문, 1995. 1.