실시간 시스템의 Fault Tolerance를 위한 모니터링 기법

심재환° 김진현 양진석 최진영
고려대학교 컴퓨터학과 정형기법 연구실
{jhsim°, jhkim, jsyang, choi}@formal.korea.ac.kr

Monitoring Method for Fault Tolerance of Real-time System

Jae-Hwan SIM° Jin-Hyun KIM Jin-Seok YANG Jin-Young CHOI
The Computer Theory and Formal Methods Lab, Dept. Computer Science, Korea University

요 약

안전성이 강조되는 실시간 시스템에서 시스템이 시간의 제약을 만족해야만 한다. 실시간 시스템에서 오류는 잘못된 응답 뿐 만이 아니라 시간적으로 늦은 응답에 대해서도 오류로 분류를 할 수가 있다. 이런 오류들을 모니터하기 위해서 본 논문에서는 커널에 Timed Conformance Monitor를 모듈로 추가하였다. Timed Conformance Monitor를 통해서 실시간 태스크가 시간의 제약을 만족하는지를 분석하고 또한 분석 결과에 따라 오류를 처리할 수 있는 Fault Handler를 추가하여 실시간 시스템에 대한 Fault Tolerance를 보장해 줄 수 있다.

1. 서 론

자동차 그리고 항공, 로봇, 원자력, 철도 등의 분야에서 실시간 시스템이 적용되고 있다. 실시간 시스템의 특징은 대부분 안전과 밀접한 관계를 가지고 있고, 엄격하게 시간의 제약을 지켜야 한다는 것이다. 실시간 시스템에서는 태스크들이 시간적인 제약을 가지고 있고, 태스크들이 시간적 제약을 만족하는지를 확인하는 것이 중요하다. 현재 실시간 시스템의 시간적인 부분에 대한 많은 연구가 진행되고 있다. 예를 들어 Worst Case Execution Time(WCET)를 분석하여 실시간 태스크의 스케줄링 가능성을 분석하는 것, EDF Scheduler등을 이용해서 태스크의 데드라인을 맞춰 스케줄링을 하는 것, Timed Conformance Relation을 이용하여 실시간 시스템의 Conformance Test를 하는 것 등이 그 예이다.

그러나 위의 연구들은 다음과 같은 몇 가지의 제약사항을 가지고 있다.

- 대부분의 실시간 시스템은 일회적인 작동을 하고 종료 되는 것이 아니라 계속적으로 환경과 반응을 하며 수행되는 반응형 시스템이다. 따라서 일정 수 이상의 테스트만을 가지고 그 시스템이 시간적인 제약을 만족하는 지를 보장해 주기는 어렵다.
- 많은 연구들이 실시간 시스템을 시스템 전체적인 관점에서 시간 제약을 분석한다. 그래서 커널과 태스크 사이의 상호작용에 대한 Conformance는 분석할 수 없다.
- 오류의 발생시 Fault Tolerance를 보장해 줄 수 없다.

따라서 설계 단계에서 시간을 표현할 수 있는 Timed Automata를 이용해서 실시간 태스크를 설계하고, 이 설계를 모니터에 적용시켜 시간적 제약에 따른 Conformance Relation을 분석하여 오류가 있다면 이를 처리해 줄 수 있는 기법이 필요하다. 따라서 본 논문에서는 실시간 시스템의 Fault Tolerance를 보장해 줄 수 있는 Timed Conformance Monitor를 제시한다.

2. 본 론

2.1 Timed Automata(TA)

실시간 태스크의 설계와 모니터에 이용되는 Timed Automata A는 다음과 같이 정의된다.

$A=(L, l_0, X, Act, E)$ 여기서, L은 location의 집합이고, $l_0 \in L$는 초기 location이고, X는 시간 변수의 집합이고, Act는 입력과 출력의 동작(Action)의 집합이다. 그리고, $E \subseteq L \times G(X) \times Act \times L$인 E는 Edge들의 집합이다. 여기서 $G(X)$는 location 간의 전이를 위한 시간적 조건이다. $G(X)$는 $x \sim k$의 형태로 표현 된다. 여기서 $\sim \in \{\leq, \geq, =, <, >\}$이다. 그리고, 입력 동작의 집합 Act_{in}과 출력 동작 집합 Act_{out}와 시간에 의한 동작 Act_t에 대해서 $Act = Act_{in} \cup Act_{out} \cup Act_t$이다.

2.2 Conformance Relation

Timed input output Conformance Relation(tioco)는 임의의 시간에 구현(Implementation)에서 발생하는 출력이 같은 시간에 설계 명세(Specification)에서 나올 수 있는 출력에 포함되는지의 관계를 말한다. 즉, 구현에서 발생시키는 출력이 설계 명세를 만족하는지를 판단하는 것을 말한다. tioco는 다음과 같다. A_I는 구현, A_S는 설계 명세 $Seq(A)$는 Timed Automata A의 모든 Sequence의 집합, 그리고, $out(A, \rho)$를 Automata A의 Sequence ρ이후의 출력 이라고 한다면, tioco는 다음과 같다.

$$A_I \text{ tioco } A_S \equiv \forall \rho \in Seq(A_S),\ out(A_I, \rho) \subseteq out(A_S, \rho)$$

2.3 모니터의 구성

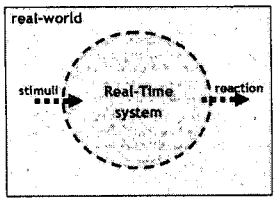

그림 1(a)

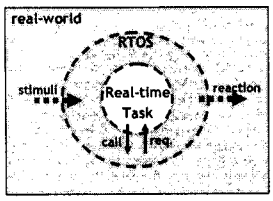

그림 1(b)

실시간 시스템을 바라보는 관점은 그림1의 두 그림과 같다. 그림1(a)는 실시간 시스템의 전체 하나의 시스템으로 보고 전체 시스템과 외부와의 반응과 응답에 대해서 실시간 시스템을 분석할 수 있다. 하지만, 이런 경우 실시간 시스템 내부에서 커널과 실시간 태스크와의 입출력 관계를 상세히 분석하기가 어렵다. 본 논문에서는 실시간 시스템을 그림1(b)와 같이 보고 커널과 태스크 간의 입출력에 대한 모니터를 하는 데에 초점을 맞추었다.

이러한 관점에서 커널에 2개의 모듈이 추가 되었고, 2개의 모듈은 다음과 같다.

- **Timed Conformance Monitor:** Timed Conformance Monitor는 태스크와 커널의 입출력을 Hooking하여 실시간 태스크가 설계 명세와 Conformance Relation을 만족하는지를 모니터하는 모듈이다.
- **Fault Handler:** Fault Handler는 실시간 태스크가 설계 명세와 Conformance Relation이 만족하지 않을 때, 오류를 처리해주는 모듈이다.

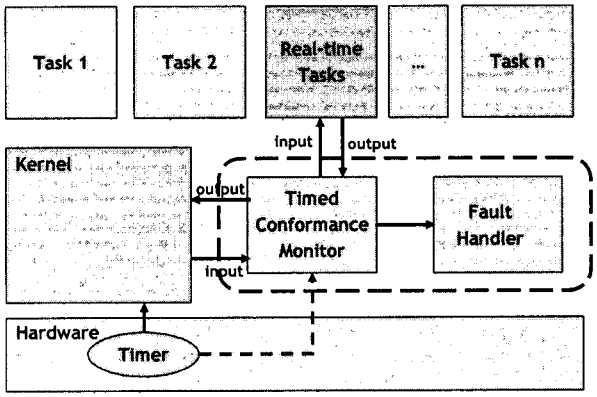

그림 2 Timed Conformance Monitor

그림 2는 Timed Conformance Monitor의 개략적인 구조이다. 시간의 제약을 가진 실시간 태스크는 커널을 통해서 외부 환경에 반응을 하게 된다. 외부의 자극을 하드웨어와 커널을 통해서 태스크에게 입력을 주고 처리 결과의 출력은 커널을 통해 외부 환경으로 전해지게 된다. 이 때 실시간 태스크는 커널과의 입출력에서 시간적인 제약을 가지게 되는데 Timed Conformance Monitor는 커널과 실시간 태스크간의 입출력을 Hooking하여 앞에서 언급된 실시간 태스크의 tioco를 분석하여 태스크가 태스크의 설계 명세대로 올바르게 동작하는지를 모니터한다. 그리고 만약 실시간 태스크가 설계 명세의 시간적 제약을 만족하지 못했을 경우에 Fault Handler가 상황에 맞는 처리를 해주게 된다. 실시간 시스템은 Timed Conformance Monitor를 통해서 실시간 태스크의 설계 명세의 시간 제약을 만족하였는지 분석 할 수 있고, Fault Handler를 통해 Fault Tolerance를 보장해 줄 수 있다. Timed Conformance Monitor와 Fault Handler가 태스크 수준이나 시스템 외부 모니터로 추가 된 것이 아니라, 커널 모듈로 추가가 되어서 생기는 장점은 다음과 같다.

- 태스크 수준으로 추가했을 경우 CPU의 제어권을 쉽게 얻을 수 없어 모니터와 Fault Handle을 하기 어렵다.
- 외부에 모니터를 두어 시스템을 모니터 할 경우 모니터와 시스템간의 통신상의 지연 때문에 정확한 시간의 평가가 어렵다.

2.4 모니터의 실행

실시간 태스크는 Timed Automata를 이용하여 설계하게 된다. 이 설계를 바탕으로 실시간 태스크를 구현하게 된다. 그리고 Timed Automata로 설계된 설계 명세는 Timed Conformance Monitor의 자료구조로 들어가서 Timed Conformance Monitor가 실시간 태스크와 설계 명세간의 Conformance Relation을 분석하는데 이용이 된다. 중요한 사항은 모니터내의 설계 명세는 모든 Location에서의 정보를 관리하는 것이 아니라, 현재 Location의 입출력 정보만 갖도록 설계 명세가 전이할 수 있도록 설계 하였다. 그 이유는 다음과 같다.

- 모든 **Location**에서의 **I/O**정보를 다 보관한다면, 상태폭발의 일으킬 수 있다.
- 임베디드 시스템에서 중요한 자원이 메모리를 절약할 수 있다.

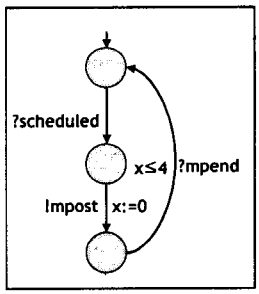

그림 3(a) 설계 명세

```
void TaskA()
{
    while(1)
    {
        ...
        A= ...
        kMboxPost(mbox1, A);
    }
}
```

그림 3(b) 구현

그림 3(a)는 구현하고자 하는 실시간 태스크의 설계 명세이다. 이 태스크는 커널로부터 스케줄링을 받아 메일박스에 메시지를 보낸 뒤 4 tick안에 다른 태스크가 메시지를 받아가는 태스크를 Timed Automata로 명세한 태스크 이다. 그림 3(b)는 이 명세대로 실제 태스크를 구현한 것이다. 구현에서 이 태스크는 A라는 메시지를 내보내기만 하면 된다.

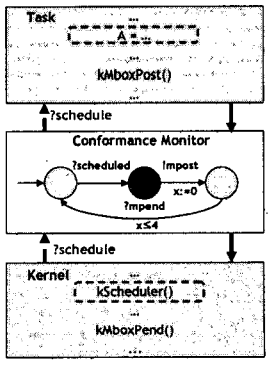

그림 4(a)

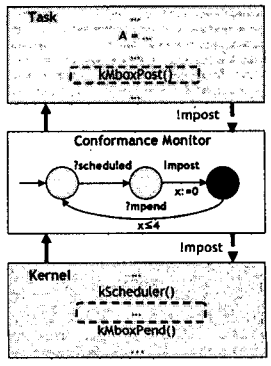

그림 4(b)

그림 4로 실제 모니터가 어떻게 동작을 하는지 설명하겠다.

앞의 Timed Automata로 모델링된 설계 명세는 모니터로 들어간다. 명세대로 스케줄러가 태스크를 스케줄링 해주게 되면, 모니터가 이 정보를 받아서 모니터 내부에 있는 설계 명세를 초기 location에서 다음 location으로 전이를 시켜준다. 그리고 태스크가 수행된 후 메일박스로 메시지를 전해 준다. 이 때 모니터는 이 함수를 Hooking하여 태스크로 부터의 출력이 설계 명세 상에서 현재 위치하고 있는 Location에서 나올 수 있는 출력인지를 판단한다. 만약 잘못된 출력 이었다면 Fault Handler를 통해 오류상황에 대한 처리를 해주고, 정상적인 시점에 나온 출력이라면, 설계 명세의 Location을 해당하는 다음 Location으로 전이 시켜 주고 커널에게 통해 메시지를 전달한다. 이런 과정이 반복되면서, 태스크의 출력이 설계 명세상의 현재 Location으로부터 나올 수 있는 출력인지를 판단하는 것이 Timed Conformance Monitor이다. 이와 같이 앞선 그림 3(a)의 설계 명세에서는 태스크와 커널간의 시간의 제약이 표현되었지만, 구현상의 태스크에는 따로 시간적 제약을 표현해 주기 어렵다. 그러나 Timed Conformance Monitor를 통해 커널과 태스크간의 입출력에 대한 시간적인 분석이 가능하고, 오류상황에 대처하기 쉬워 시스템의 Fault Tolerance를 보장해 줄 수 있다. 특히 선점형 커널에서 태스크 내부 루틴 간의 시간제약을 분석하기에 용이하고, 데드락 상황 시 해제가 용이하다.

모니터를 실행하는 알고리즘은 다음과 같다.

Monitor Execution Algorithm

Initially Z:= $\{<l_0, 0>\}$.
update location:
  while (Z≠Ø) do below :
    if recv(act) ∈ kernel_ouput(Z) then
      send act to implementation
      Z:=update_location(Z, act)
      set time variable
    else if recv(act) ∈ imp_ouput(Z) then
      send act to kernel
      call check conformance
      Z:=update_location(Z, act)
      set time variable
    else if recv(act) == tick then
      decrement time variable
check conformance:
  if recv(act) $\notin$ imp_output(Z) then
    call Fault Handler
    return fail
  else if Z=Ø then
    call Fault Handler
    return fail
  else continue

2.5 Experiment

실시간 태스크의 시간 지연에 의해 생기는 오류를 Timed Conformance Monitor가 잘 분석하고 적절히 Fault Handler가 동작하는지 확인하기 위해서 다음과 같은 실험을 하였다.

- Host : windows2000
- Target : TMS320C32
- Target RTOS : uCOS-II
- 총 태스크의 수 : 15개
- 1 tick의 시간 : 10ms
- 모니터 대상의 태스크 : 1개 (Monitored_Task)
- 설계 명세상의 시간 제약 : 태스크는 매 tick 마다 실행되어야 한다.

이 실험은 태스크의 설계 명세에 태스크가 매 tick 마다 수행되도록 명세하였고, uCOS-II에 이 명세대로 설계한 태스크는 다음의 Monitored_Task와 같다. 이 소스코드에서 Task는 스케줄링을 받으면, dummy_func()를 호출한 뒤 다음 Tick에서 수행되게 되어 있다. 하지만 사실상 이 태스크는 매 Tick마다 Ready 상태가 될 뿐 매 tick에 실행된다는 보장은 없다. 이것은 나머지 14개의 태스크의 상태와 인터럽트 상태에 따라 시간 지연이 생길 수 있다는 것이다.

```
void Monitored_Task()
{
    while(1)
    {
        dummy_func();
        OSTimeDly(1);
    }
}
```

이 태스크가 경성 실시간 태스크라면 시스템에 큰 영향을 줄 수 있다. 이것을 Timed Conformance Monitor로 모니터하여 Fault Handler가 경고를 출력하도록 하였다. 결과는 다음과 같다.

- 실험 총 시간 : 48시간
- 결과 : 총 78회의 경고가 출력

3. 결론 및 향후 연구 과제

실시간 시스템은 시스템이 시간적 제약을 만족하는지가 매우 중요한 시스템이다. 그리고 실시간 시스템의 대부분의 안전필수 시스템에 적용되고 있다. 따라서 시스템이 시간적 제약을 만족하지 못하면, 재산, 인명에 크나큰 피해를 안길 수 있다. 그래서 실시간 시스템에서 설계상의 시간적인 제약조건을 만족하는지에 대한 많은 연구가 진행되어 왔다. 그러나 대부분의 연구가 테스트라는 방법으로 이루어져 왔고, 이런 방법으로는 무한히 동작하여야 하는 실시간 반응형 시스템이 시간적인 제약 사항을 만족하는지 보장해 줄 수 없다. 그래서 본 논문에서는 실시간 시스템을 커널수준에서 모니터하여 분석하고 문제가 발생하게 되면 Fault Handler를 통해 적절한 처리를 해 줄 수 있도록 하였다. 이 기법을 통하여 커널과 태스크간의 상호작용에서 시간적인 제약을 체크할 수 있어서 단지 태스크가 제시간에 스케줄링 되는지 뿐만 아니라 태스크 내부루틴의 입출력에 대한 시간적인 평가가 가능하게 되었다. 또한 이런 기법은 Fault Tolerant시스템에 적용이 가능하고, 태스크가 데드락에 걸렸을 때 이를 해제할 수 있다. 이를 통해 안전성이 높은 실시간 시스템을 개발 할 수 있다.

하지만, 아직은 시간을 시스템의 타임 틱 단위의 디지털 클럭으로만 시스템을 모니터링하고 오류를 처리할 수 있다. 따라서 향후 연구 과제는 아날로그 시간에 대한 연구를 진행할 예정이다.

4. 참고 문헌

[1] Moez Krichen, Stavros Tripakis, "Black-box Conformance Testing for Real-Time Systems", In *SPIN'04 Workshop on Model Checking Software,* 2004

[2] Marius Mikucionis, Kim G. Larsen, Brian Nielsen, "Online On-the-Fly Testing of Real-time Systems", http://www.brics.dk, 2003