Statechart with Timed Shared Resource의 명세 및 검증

[1]김진현, 최진영
고려대학교 컴퓨터학과
(jhkim, choi)@formal.korea.ac.kr

Specification and Verification using Statechart with Timed Shared Resource

Jin Hyun Kim, Jin Young Choi
Dept. of Computer Science, Korea University

요 약

원자력 발전 및 항공 시스템과 같은 실시간 시스템의 설계는 자원 및 시간적인 분석은 상당히 중요한 부분을 차지하고 있다. 이러한 설계는 그 설계단계로부터 철저한 명세 및 검증이 이루어져야 한다. Statechart는 Reactive 시스템을 모델링 하는 도식언어이다. 하지만 기존 Statechart 언어는 자원에 대한 시간적 명세가 분명치 않아 상태로 모든 것을 표현하기에는 복잡하고 용이하지 않다. 또한 이러한 명세 도구는 시스템의 검증은 물론 시간적 행위의 검증은 수월치 한다. 본 논문은 Statechart에 시간을 소모하는 자원을 명세하는 문법을 소개하고 이를 검증하는 기법을 제시하고자 한다.

1. 서론

원자력 발전 계통이나 항공시스템 혹은 의료 기기에 내장되는 시스템은 일반적인 시스템과 달리 신뢰성과 안정성이 절대적으로 필요로 한 시스템이다. 특히 이러한 시스템은 Time-critical 한 시스템으로 자원에 대한 시간적 요소가 시스템의 안정성과 신뢰성에 절대적으로 관련되어 있다. 이러한 시스템의 구현은 그 설계 단계부터 철저한 명세와 검증을 통해 이루어 지고 있다. 실시간 시스템에 대한 다양한 정형적인 명세 도구가 개발되고 있음에도 불구하고 이러한 설계들은 명세를 이해하기 어렵게 만든 수학적이 문자로 표현하기 때문에 시스템 제작에 참여하는 사람들만이 접근하는 경향이 있었다. 따라서 최근에 이르러서는 Statechart[1,2]나 Modechart[3], Timed-automata[4], CRSM(Communicating Real-time State Machine)과 같은 정형적이고 도식적인 언어가 개발되기에 이르렀다.

Statechart 도식적 정형 명세 언어로써 대형 Reactive 시스템을 설계하는데 응용되는 명세 언어이다. 이 언어는 다음과 같은 특징으로 그 효용성을 가지고 있다. 첫째, Statechart는 도식적 언어이기 때문에 설계자와 구현자 혹은 그 외의 다른 사람이 인식하고 이해하기 쉽다. 둘째, Statechart는 계층적 구조와 orthogonal 구조를 가지고 있기 때문에, 일반적인 State diagram보다 훨씬 더 간편하고 단순하게 시스템을 표현할 수 있다. 또한 간단한 문법을 지니고 있기 때문에 다른 도식적 명세 언어 보다 시스템을 표현하기에 훨씬 수월하다. 이러한 편리성 때문에 Reactive 시스템이나 프로토콜 등을 모델링 하는데 다각도로 사용되고 있다. 하지만 실시간 시스템처럼 시간의 흐름을 표현해야 할 경우, Statechart는 이벤트 유도만으로 시스템을 표현하기 때문에 상당히 제한적일 수 밖에 없다. 즉 시간의 흐름에 따라 시스템의 변화를 자세히 묘사하기 때문에 시간에 대한 명확한 문법을 지니고 있지 않은 Statechart를 통해서 실시간 시스템을 명확하게 묘사하는 것은 상당한 부담을 안아야 한다. 또한 Statechart는 공유 자원의 점유에 대한 명확한 의미도 지니고 있지 않다. 즉 특정 순위와 자원의 점유에 대한 표현과 공유되는 자원의 선점에 대한 우선순위에 대한 표현을 가지고 있지 않다. 따라서 시간과, 자원 그리고 자원에 대한 우선순위를 반드시 명세해야 하는 실시간 시스템을 위해서는 Statechart는 적합하지 않을 수 있다. 본 논문에서는 이러한 문제를 해결하고자 보다 확장된 Statechart 인 Statechart with Timed Shared Resource(이하 STSR)를 제안하고 또한 이를 검증하는 기법을 제시한다.

2장에서는 Statechart를 확장한 Statechart with Timed Shared Resource에 대한 시간 및 자원, 자원에 대한 우선순위에 대한 명확한 문법을 설명하고 3장에서는 이를 검증하는 기법을 소개하고 이를 검증하는 기법을 설명한다. 4장에서는 결론 및 향후연구방향을 논하기로 한다.

2. Statechart with Timed Shared Resource

실시간 시스템에서의 시간적 흐름에 따른 자원의 사용을 표현하기 위해서는 다음과 같은 요소들이 필요하다.

첫째로 시간을 표현하기 위한 명확하게 문법이 필요하다. 즉 어떤 상태에서 어느 정도의 시간을 사용할 것인지에 대한 정량적이고 명확한 표현이 필요하다. 둘째로 공유된 자원의 사용에 대한 시간적 표현이 필요하다. 셋째로 자원은 공유할 수 있으며 이러한 공유된 자원을 다수의 프로세스가 점유하려 할 경우 자원의 점유에 대한 우선순위가 표현되어야 한다. 만약 동시에 하나의 자원을 사용해야 할 여러 프로세스가 존재할 경우 이를 조율하기 위한 메커니즘이 필요하다. 실시간 시스템을 설계하기 위해 본 논문에서는 위와 같은 필요에 따라 Statechart를 확장한다.

본 논문에서 소개하고 있는 Satechart with Timed Shared Resource(이하 STSR)은 다음과 같은 정의를 기본으로 한다.

Def 2.1 Statechart with Timed Shared Resource
STSR = {S, s_0 ,→, Σ, TC, R,P}
S : a set of State.
s_0 : a initial state.
→ : ⊆ S ×S : a set of transition.
Σ : S × Σ ×S : a set of Label.
TC : → → tc, a set of time constraints, a function that labels each transition with clock constraint.
R : a set of resources
P : priority

Def 2.2 (State) : S
State는 다음과 같이 정의한다.

- Base state : 시간을 소모하지 않는 Label만을 가진 상태를 의미한다.
- And state : orthogonal state, 두 개 이상의 동시에 수행되는 상태를 의미한다.
- Or state : 두 개 이상의 orthogonal하지 않은 상태들을 의미한다.
- Timed-resource state : 자원을 가지거나 혹은 가지지 않고 시간을 소모하는 상태를 의미한다.
- Timed-state : 자원을 가지지 않고 시간만을 소모하는 상태를 의미한다.
- Link state : 전이의 분기를 유도하는 임시 상태이다.

Def 2.3 (Transition : 전이) : → ⊆ S ×S
상태에서 상태로의 변화를 ***Transition***이라 한다. 하나의 Transition <q,a,q'>에 대해 q, q' ∈ S이고 a ∈ Σ 이라면, q는 a에 의해 q'로 상태를 변화한다.

Def 2.4 (Label) : Σ
Transition을 일으키는 이벤트 및 수행되는 action의 집합을 ***Label***이라 한다. Label은 다음과 같이 표현된다.

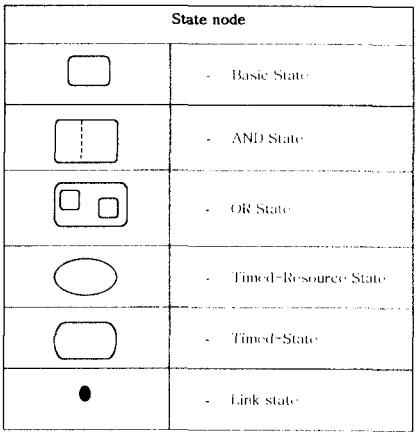
e/a , e ∈ E(Event), a ∈ A(Action)

action에는 다음과 같은 두 가지 수행을 할 수 있다.
첫째로 이벤트의 출력이 있다. 이것은 특정 이벤트를 출력하는 것으로써 모든 이벤트는 broadcasting 된다. 둘째로 assignment가 있다. 이것은 특정 변수의 값을 변화시키는 것이다. 이것은 주로 데이터 값을 변화를 수행한다.

Def 2.5(Time constraint) : TC : → → t , t ∈ TC,
상태에서 자원을 소모하는 전이가 일어날 경우, 자원을 사용하는 시간에 대한 제약조건을 **Time constraint**라 한다. TC는 다음과 같은 형태를 갖는다.

~R : R은 실수, ~ ∈ { >,<,>=,<=,==}

전이의 시간적인 의미는 다음과 같다.
첫째로 Basis state에서 Basic state로의 전이이다. 이 전이는 시간을 전혀 소모하지 않고 instantaneously하게 전이한다. 즉 Basic state에서 일어나는 전이는 전혀 시간을 소모하지 않는다. 하지만 timed state나 timed-resource state로부터 시작하는 전이는 반드시 시간을 소모해야 한다.
{BS, TS, TRS} ∈ S 이고 s ∈ S, bs ∈ BS(Basic state),



State Label	
NODE_NAME	- Node name - 상태의 이름을 명세
{(r,p),…}	- Timed-Resource node label - Timed-Resource node 에 붙는 Label - r : resource 이름 - p : 우선 순위
@name	- Refined state

그림 2. State Label

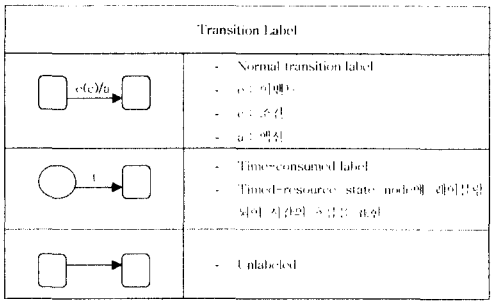


그림 3. Transition Label

ts ∈TS(Timed states), trs ∈ TRS(Timed resource states) 이고 t를 시간이라 할 때,

bs × t ×s , t = 0,
ts × t ×s , t > 0,
tr × t ×s , t > 0,

이다
<그림 1,2,3>은 다음은 각 요소들의 도식적 표현이다.

2.1 Behavioral Semantics of STSR
본 논문에서 제시하는 STSR의 행위에 대한 의미는 비정형적으로 설명하겠다.
본 논문에서 제시하게 될 STSR은 다음과 같이 시간적 모델을 가정한다. 첫째, 상태와 상태의 전이인 Step은 시간을 전혀 소모하지 않는다. 이러한 Step을 단순히 step으로 정의한다. 즉 일반

본 논문에서 제시하게 될 STSRd은 다음과 같이 시간적 모델을 가정한다. 첫째, 상태와 상태의 전이인 Step은 시간을 전혀 소모하지 않는다. 이러한 Step을 단순히 step으로 정의한다. 즉 일반적 상태와 상태로 전이할 때에는 시간을 소모하지 않고 instantaneous하게 전의한다. 둘째, 상태에서 시간을 소모하게 는 상태 노드로 갈 때에는 적어도 한 clock 이라도 소모해야한다. 시간을 명세하는 노드는 소모하는 시간의 양을 명확하게 표현하게 되고 이러한 시간의 진행만이 실시간적 시간의 흐름으로 가정한다. 그리고 이러한 시간적 흐름을 포함한 step을 timed-step으로 정의한다. 또한 특정 자원을 사용할 때에는 반드시 시간을 소모해야 한다. 즉 모든 자원의 점유는 시간적 흐름을 가져야 한다.
모든 이벤트는 broadcasting 되며 Action 부분에는 이벤트의 발생 및 assignment가 올 수 있다.
자원에 대한 경쟁이 일어날 경우 자원에 대한 우선수위가 높은 순서대로 자원이 할당된다. 만약 자원에 대한 경쟁이 일어날 경우 낮은 우선 순위를 가진 프로세스는 자원을 사용할 수 있을 때까지 대기 상태에 들어가게 된다. Transition에 대한 경쟁 관계도 존재하는데 만약 같은 이벤트로 다른 하나의 프로세스 내에서 서로 다른 계층에 존재할 때, 이것은 더 높은 수준의 State의 전이를 먼저 일어나게 한다.

3. STSR의 시간적 검증

본 논문에서 정의한 STSR은 <그림 4>와 같은 과정으로 시간적 검증 및 분석을 수행한다.

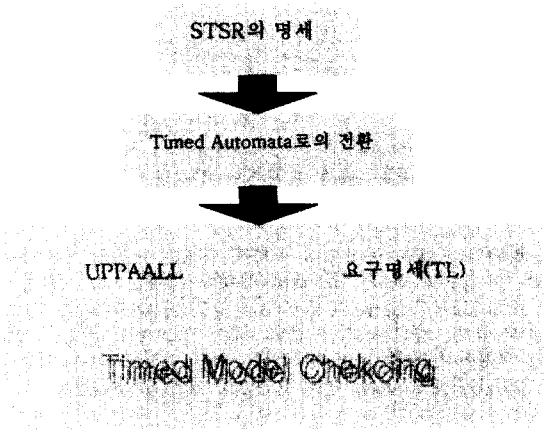


그림 4. STSR의 시간적 분석 및 검증

본 논문에서 제시하고 있는 방법은 우선 UPPAAL 이라는 도구로 STSR을 검증하고자 한다. Timed-automata는 일반적인 오토마타에 시간적인 의미를 부여한 것으로 STSR과 상당히 유사하다. 즉 시간을 소모하는 Transition 에서만 시간을 소모하게 하는 오토마타이다. UPPAAL은 현재 스웨덴의 Uppsala University 와 Aalborg University에서 개발 및 확장하는 도구로 시스템의 시간적인 분석에 이용되는 시간적 모델체커(Timed Model checker)이다.
이 도구는 Timed-automata의 모델링과 시뮬레이션 및 시간적 모델체킹을 수행한다.
본 논문에서 STSR의 명세를 Timed-automata로의 변환은 기술하지 않겠다.
다음 <그림 5>는 STSR로 명세한 Railroad crossing 시스템의 차단기를 설계한 설계명세이다.

4. 결론 및 향후 연구 방향

본 논문에서는 실시간 시스템을 설계하고 분석하는데 적합한 Statechart with Timed Shared Resource 를 제안하고 이를 검증하는 기법을 제시한다.
실시간 시스템은 자원에 대한 시간적 분석이 필요하며 특히 Time-critical 시스템이나 Safety-critical 시스템과 같은 시스템에서는 이에 대한 검증이 절대적으로 필요하다.
따라서 본 논문에서는 현재 가장 잘 알려진 도식적 명세 언어인

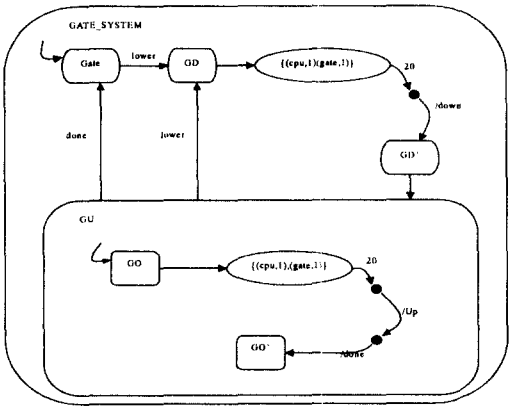


그림 5. STSR로 명세한 Gate 시스템

Statechart를 확장하여 자원과 자원에 대한 시간적인 명세 그리고 자원의 사용의 경쟁관계에서의 우선순위를 명세할 수 있는 STSR을 제안하고 있다. 또한 이를 검증할 수 있도록 Timed-automata로의 변환을 제안하고 있다.
향후 연구 과제로는 STSR의 Timed-automata로의 변환에 대한 명확한 정의와 변환 자동화 도구의 개발이 수반되어야 할 것이다

5. 참조문헌

[1] David Harel, STATECHART: A VISUAL FORMALISM FOR COMPLEX SYSTEMS, Science of Computer Programming 8 (1987) pp231-274
[2] David Harel and Ammon Naamad, The STATEMAT E Semantics of Statecharts, ACM Trans. Soft. Eng. Method. Oct. 1996
[3] Paul Clements et. Al . Modechart User's Guide, Center for Computer High Assurance Systems. US Naval Research Lab.
[4] Alur. R. and D.L.Dill. A Theory of Timed Automata. Theoretical Computer Science 126, p183-235
[5] H. Ben-Abdallah, I. Lee, and J. -Y. Choi, A Graphical Language with Formal Semantics for the Specification and Analysis of Real -Time Systems, *Proceedings of the 16th IEEE Real-Time Systems Symposium*, 1995.,
[5] Johan Bengtsson and Fredrik Larsson, UPPAAL-A Tool for Automatic Verification of Real -Time Systems, Uppala University, 1996