BM-DEVS 모델링과 시뮬레이션 환경에서의 시공간 문제 해결을 위한 시간 논리 적용 기법

안정섭O, 조대호*
O성균관대학교 전자전기컴퓨터공학과,
*성균관대학교 소프트웨어학과
e-mail: sc4217@skku.eduO, thcho@skku.edu*

Temporal Logic Application Technique for Solving Spatio-temporal Problem in BM-DEVS Modeling And Simulation Environment

Jungsub AhnO, Taeho Cho*
ODept. of Electrical and Computer Engineering, Sungkyunkwan University,
*Dept. of Computer Science and Engineering, Sungkyunkwan University

● 요 약 ●

사회적으로 복잡한 문제들이 시공간 형태로 문제 표현이 가능하고 이를 활용하여 문제를 해결하기 위한 연구들이 진행 중이다. 특히, 시뮬레이션 이론 중 하나인 BM-DEVS는 시공간 논리를 적용하여 실세계에서 일어나는 문제들을 시공간 규칙으로써 표현하였고 이를 모델에 적용하여 시스템에서 행위를 모니터링한다. 하지만 BM-DEVS에서는 시스템 차원에서 정의된 시공간 규칙들을 평가하기 위하여 Büchi 오토마타로의 변환과 오토마타를 모델들에 반영할 수 있어야 한다. 이를 위하여 시스템을 구축하는 모델러가 직접 규칙을 오토마타로 변환하는 작업을 해야하며 이에 대한 오토마타를 모델에 적용하기까지는 많은 시간이 소요된다. 이러한 문제를 해결하기 위해 본 논문에서는 모델링의 단순화를 위하여 시공간 규칙을 모델들에 자동적으로 적용하는 방법에 대하여 소개한다.

키워드: 행위 감시 이산사건 시뮬레이션(behavior monitor-DEVS), 지식 변환(knowledge transformation) 시공간 논리(temporal logic)

I. Introduction

IT 기술이 발전함에 따라 컴퓨터 기술과 일상생활과는 매우 중요한 영향을 미친다. 예를들어 군사, 의료, 교통, 은행, 스마트 공장 등과 같은 응용들에서 안전 관련 시스템 고장은 매우 위험한 결과를 가져올 수 있다[1-3]. 이러한 응용뿐만 아니라 컴퓨터 시스템 동작에 대한 정확성 검증은 하드웨어 및 소프트웨어 설계 및 구현 부분에서도 필수적인 요소로 자리 잡고 있다.

다양한 보안 소프트웨어 및 하드웨어 개입을 통해 오류 수를 줄이는데 도움되고 있지만 이러한 시스템들은 프로세스에 대한 오류가 존재하지 않는다는 것을 증명할 수 없으며 시스템에서 시공간적인 문제가 발견된다면 해당 오류를 검출하기 위한 검증 기술들을 도입하여 새로운 보안 시스템을 구축하여야 하므로 많은 사회적 비용이 요구된다. 이러한 문제점을 해결하기 위해 새로운 시뮬레이션 이론인 BM-DEVS (Behavior Monitor Discrete EVent Specification; 이하 BM-DEVS)[4]가 등장하였다. BM-DEVS는 모델별로 행위 모니터링을 위한 시공간 규칙들을 사용할 수 있으며 전문가 시스템과 같이 규칙 관리가 용이하여 보안 지식들을 관리하기 편리하다는 장점이 있다. 하지만 BM-DEVS를 이용한 완전 자동화된 시스템의 구축을 위하여 시공간 규칙을 Büchi 오토마타로 변환과 모델에 적용해 주는 방법이 필요하다.

본 논문에서는 모델에 입력된 규칙들을 Büchi 오토마타로 변환과 적용 방법에 대하여 서술하였다. 본 논문의 구성은 다음과 같다. 2장에서는 제안 방법의 기반인 BM-DEVS와 시공간 규칙에 대하여 설명한다. 3장에서는 제안 방법에 관하여 자세히 다룬다. 마지막으로 4장에서는 결론을 맺는다.

II. Preliminaries

1. Related works

1.1 BM-DEVS

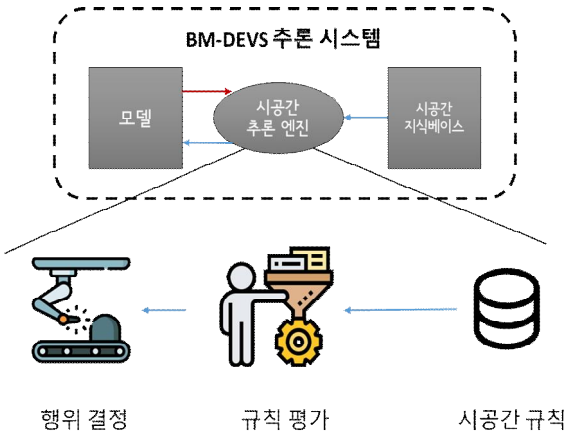

Fig. 1. BM-DEVS 추론 시스템

Fig.1은 BM-DEVS의 추론 시스템을 나타낸다. Cho가 제안한 BM-DEVS는 DEVS[5]를 기반으로한 실제 시스템의 행위 모니터링을 위한 이론이다. BM-DEVS는 모니터링을 통해 모델 내에서 원하는 행동 패턴을 간접적으로 알 수 있다. BM-DEVS에서 행동을 감지하기 위한 동작들은 모델내에 시간 논리(Temporal Logic; 이하 TL)로 표현된다. BM-DEVS의 구성 요소는 다음과 같다.

$$N_{BM\text{-}DEVS} = (N,\ P,\ B,\ Z_{f,p})$$

N은 DEVS의 커플드 모델을 나타내고 P는 상태전이에 대한 집합을 나타내고 B는 TL에서 변환된 Büchi 오토마타를 나타낸다. $Z_{f,p}$는 Büchi 오토마타에 의한 상태 전이 함수를 나타낸다. 이와같이 BM-DEVS는 행위를 추론하기 위해 TL 규칙에서 Büchi 오토마타로 변환이 필요하다. 또한 변환된 오토마타들은 규칙과 1:1 관계로 매칭되어야 한다.

1.2 Temporal Logic

시간 논리는 동시 및 분산 시스템의 검증과 보안 시스템 등 광범위하게 사용되고 있다. 시간 논리를 통해 시간에 대한 문제를 명확히 표현할 수 있다. 시간 논리는 기본 논리 연산자인 ∨(OR), ∧(AND) 뿐만아니라 시간 표현을 위한 ○(Next), ◇(Someday), ∪(Until), □(Always) 등 다양한 연산자를 통해 복잡한 문제를 하나의 식으로 표현한다. 이와같이 현실 세계 속 시공간 문제를 시간 논리로 표현하고 시뮬레이션과 융합하여 시공간 문제를 효과적으로 해결한 연구들이 진행되고 있다[6, 7].

III. The Proposed Scheme

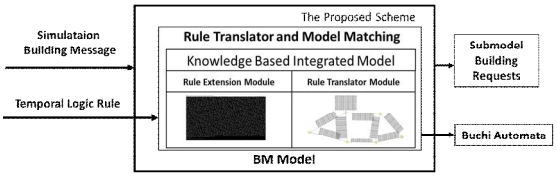

Fig. 2. 제안 방법 개요도

Fig.2는 제안 방법에 대한 개요도를 나타낸다. 제안방법은 시뮬레이션을 진행시키기위한 Building Message를 전달받으면 모델에 정의된 TL 규칙들을 로드하게 되고 제안 방법을 통해 최종적으로 오토마타 형태가 나오게 된다. 또한 하위 모델이 있다면 하위 모델에게 Building 요청 메시지를 보내게 된다. 그러므로 시뮬레이션이 동작하기 전에 모든 TL 규칙에 대한 오토마타 변환 작업이 이루어진다.

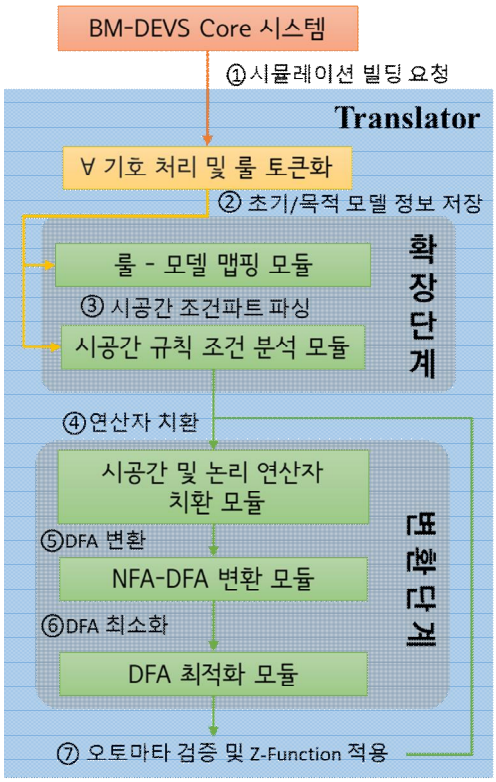

Fig. 3. Büchi 오토마타 변환 기법

제안 방법은 Fig. 3과 같이 시뮬레이션 환경에 시간 논리 적용을 위해 크게 확장단계와 변환단계로 나뉜다. 확장단계 진입 전 모델 내에 정의된 시공간 규칙들에 대한 ∀ 심볼 처리와 규칙 토큰화를 진행한다. ∀ 심볼에 대한 처리로는 시뮬레이션 시작 전 모델 구성 단계에서 사용자에게 모델의 수를 입력받는다. 시공간 규칙은 규칙 토큰화를 통해 조건부와 결과부로 나뉘게 되며 이후 모델 내에 정의된 시공간 규칙을 확장하는 단계로 진입한다. 확장단계에서는 조건부로 나뉜 토큰에서 ∀ 입력에 따라 동적으로 규칙을 확장 시킨다. 시공간 규칙 조건 분석단계에서는 확장된 규칙을 이용하여 시공간 규칙에서

Büchi 오토마타로 변환하기 위한 연산자를 분석하고 변환단계로 진입한다. 변환단계의 초기 단계에서는 S_0와 S_1인 2개의 Vertex로 구성되며 $S_0 \rightarrow S_1$ 상태로 전이하기 위한 조건(Edge)은 전 단계에서 확장된 규칙이 된다. 이후 시공간 규칙 조건 분석단계에서 결정된 연산자에 시공간 연산자가 포함되어 있다면 [8]에 의거하여 오토마타로 각 연산자에 맞도록 치환한다. 이렇게 치환을 한 결과는 NFA (Non-deterministic Finite Accepter) 또는 DFA (Deterministic Finite Accepter) 형태로 나오게 된다. 만약 치환된 결과가 NFA 형태를 가진다면 오토마타를 DFA 형식으로 변환한다. 이후 DFA minimization 알고리즘을 통하여 DFA를 최적화한다. 마지막으로 오토마타 검증단계에서는 오토마타에 표현된 모든 Vertex의 Edge를 순회하며 입력에 시공간 연산자가 있는지 검사한다. 만약 시공간 연산자가 존재한다면 다시 변환단계를 수행한다. 그렇지 않다면 시공간 규칙에 정의된 결과를 통해 Z-function을 적용한다. 최종적으로 변환된 나온 오토마타는 각 $N_{BM\text{-}DEVS}$에 저장되며 이후엔 시뮬레이션 도중 모델의 상태 변화에 의해 상태전이를 한다. 또한 최종 출력인 오토마타 검증을 위해 오토마타에 대한 File Out 기능을 제공한다. 출력된 파일은 [9]에서 제공되는 JFLAP을 이용하여 오토마타를 시각적으로 출력할 수 있다.

IV. Conclusions

현실 세계에서는 수많은 문제들이 시공간 형태로 표현될 수 있다. 시공간 문제를 모델에 표현하여 해결하기위해 BM-DEVS에서 오토마타 변환과 모델과의 매핑이 반드시 필요하다. 본 논문에서는 시공간 논리 기반 규칙들을 모델에 매핑 및 오토마타 변환 방법에 대하여 제안하였다. 향후 연구에서는 오토마타 변환 성능을 높이기 위한 연구를 진행할 것이며 다양한 도메인에서 제안 방법과 시공간 규칙들을 이용하여 제안 방법을 검증할 예정이다.

ACKNOWLEDGMENT

이 논문은 2021년도 정부(과학기술정보통신 부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2021R1A2C2005480)

REFERENCES

[1] Gotarane, Vishal, and Sandeep Raskar. "IoT practices in military applications." 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2019.

[2] Sisinni, Emiliano, et al. "Industrial internet of things: Challenges, opportunities, and directions." IEEE transactions on industrial informatics 14.11 (2018): 4724-4734.

[3] Samad, Abdus, et al. "Internet of vehicles (IoV) requirements, attacks and countermeasures." 2018 5th International Conference on “Computing for Sustainable Global Development. 2018.

[4] Cho, Tae Ho. "Simulation Methodology-Based Context-Aware Architecture Design for Behavior Monitoring of Systems." Symmetry 12.9 (2020): 1568.

[5] Zeigler, Bernard P., Tag Gon Kim, and Herbert Praehofer. Theory of modeling and simulation. Academic press, 2000.

[6] Ahn, Jung Sub, and Tae Ho Cho. "Modeling and Simulation of Abnormal Behavior Detection Through History Trajectory Monitoring in Wireless Sensor Networks." IEEE Access 10 (2022): 119232-119243.

[7] Merz, Stephan, Martin Wirsing, and Júlia Zappe. "A spatio-temporal logic for the specification and refinement of mobile systems." International Conference on Fundamental Approaches to Software Engineering. Springer, Berlin, Heidelberg, 2003.

[8] Fisher, Michael. An introduction to practical formal methods using temporal logic. John Wiley & Sons, 2011.

[9] Rodger, Susan, and T. Finley. "JFLAP." (2015).