시스템 오브 시스템즈의 효율적 검증을 위한 목표 모델 슬라이싱*

현상원○ 송지영 지은경 배두환
한국과학기술원 전산학부
{swhyun, jysong, ekjee, bae}@se.kaist.ac.kr

Goal Model Slicing for Efficient Verification of System of Systems

Sangwon Hyun○ Jiyoung Song Eunkyoung Jee Doo-Hwan Bae
School of Computing, KAIST

요 약

시스템 오브 시스템즈(System of Systems, SoS)는 독립적인 운영과 관리가 가능한 시스템들로 구성된 크고 복잡한 시스템이다. SoS는 시스템 외부 환경, 혹은 시스템 내부적인 요소들에 의해 끊임없이 변화하며, 변화가 발생할 때마다 SoS 목표 모델 내에 있는 모든 목표들의 달성 여부를 반복하여 검증할 필요가 있다. 하지만 변화가 발생할 때마다 모든 SoS 목표에 대해 검증을 진행하는 것은 효율적이지 못하다. 본 논문에서는 SoS 목표 모델 내에서 변화와 관련된 목표들을 추출하는 기법인 SoS 목표 모델 슬라이싱 기법을 제안한다. 기법을 구현한 도구인 SoS 목표 모델 슬라이서는 SoS 목표 모델을 변화와 관련 있는 하위 시스템 수준의 목표로부터 최상위 SoS 목표까지 역방향 슬라이싱을 수행하여, 특정 변화와 관련된 목표들만을 찾아낸다. 기법의 정확성을 보이기 위해 대형 재난 사고 대응 SoS 시나리오를 이용하여 실험을 진행하였다. 그 결과, SoS 목표 모델 슬라이서에 의해 슬라이싱된 검증 목표들은 SoS 변화에 따라 그 검증 결과가 달라졌지만, 그 외의 SoS 목표들은 결과에 큰 영향이 없음을 확인함으로써 기법의 정확성을 보였다.

1. 서 론

시스템 오브 시스템즈(System-of-Systems, 이하 SoS)는 개별 시스템으로는 달성할 수 없는 상위 수준의 목표 달성을 위해 여러 독립적인 이종의 시스템들이 상호작용을 하며 작동하는 대규모 복잡 시스템이다 [1]. SoS의 대표적인 예시로는 대형 재난 사고(Mass Casualty Incident, 이하 MCI)에 대응하기 위한 재난대응시스템, 전쟁 승리 확률 및 전략을 확인하기 위해 사용하는 전쟁 시뮬레이션 시스템, 그리고 스마트팜, 무인 드론 배송시스템과 같은 사물인터넷(Internet of Things, 이하 IoT)을 들 수 있다 [2]. D. Seo 외[3] 연구에서는 예제 시스템들과 같이 크고 복잡한 SoS의 목표 달성 수행 확인을 위해, 그리고 SoS의 복잡성 때문에 발생하는 상태 급증 문제(State explosion problem)를 해결하기 위해 목표를 검증 속성으로 하는 시뮬레이션 기반 통계적 모델 검증을 수행하였다.

SoS는 동적 재구성(Dynamic reconfiguration)과 진화(Evolution)의 특징을 가지고 있어 외부 환경의 변화, 내부 구성 시스템들의 상호작용 등에 의해 변화할 수 있다. SoS에 변화가 발생할 때마다 SoS 모델또한 변화되어야 하며, 변경된 SoS 모델과 SoS 목표 모델내의 모든 목표들을 통계적 모델 검증기에 입력으로 주어 각 목표의 달성여부를 확인할 수 있다. SoS 목표 모델은 SoS 모델 검증에 사용되는 독립적인 모델로써, SoS가 가지는 상위 수준의 공통 목표와 각 구성 시스템(Constituent System, 이하 CS) 수준의 목표, 그리고 각 CS를 구성하는 하위시스템 수준의 세부 목표들로 구성된다.

하지만 통계적 모델 검증의 단점은 시뮬레이션을 기반으로 하기때문에 비용이 많이 든다는 점이다. SoS에 변화가 발생할때마다 이러한 통계적 모델 검증을 SoS 목표 모델 내의 모든 목표들에 대해 반복적으로 수행하는 것은 비효율적이다. 따라서 본 연구는 효율적인 검증을 위해 변화와 관련된 목표를 추출할 수 있는 SoS 목표 모델 슬라이싱 기법 및 도구를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 목표 모델과 슬라이싱에 대한 관련 연구를 서술하고, 3장에서 목표 모델 슬라이싱 기법을 제시한다. 4장에서 실험적으로 기법의 정확도를 보이고, 5장에서 결론 및 향후 연구를 기술한다.

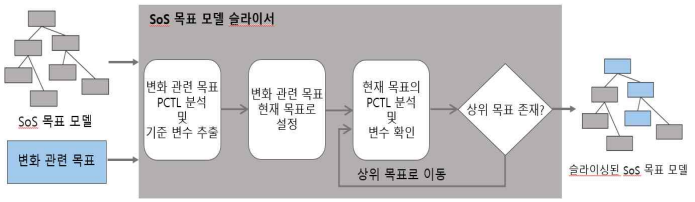
2. 관련 연구

Bordini 외의 논문 [4]에서는 다중 에이전트 시스템 언어(Multi-agent Systems Language), AgentSpeak을 대상으로 하는 속성 기반 슬라이싱 알고리즘을 제안했다. 이 논문에서는 AgentSpeak 언어로 의존성 그래프(Dependency Graph)를 생성하고, 논리 프로그래밍 언어의 속성(Property)을 이용하여 슬라이싱 기준을 만들고 슬라이싱을 진행하였다. 그 결과로 다중 에이전트 시스템의 상태 영역(State-space)을 감소시켜 효율적 검증이 가능함을 보였다. Wikinikoff 외의 논문 [5]에서는 Bordini 외의 논문 [4]이 가지고 있던 문제점을 보완하는 연구를 진행하였다. 이 논문에서는 슬라이싱 과정에서 프로그램의 순서 정보 (Sequencing Information) 를 추가하기 위해 다중 에이전트 시스템 언어의 의미(Semantic)을 기반으로 의존성 그래프를 만들었으며, 더 효율적인 슬라이싱 진행을 위해 목표

* 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No. 2015-0-00250, (SW 스타랩) 모델 기반의 초대형 복잡 시스템 분석 및 검증 SW 개발)과 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임 (2017M3C4A7066212)

속성과 관련 없는 시스템들을 제외하여 슬라이싱을 진행하였다. Shaikh 외의 연구[6] 에서는 크고 복잡한 시스템의 효율적 검증을 위해 시스템의 UML 클래스 다이어그램에 슬라이싱 기법을 적용하였다. 그리고 슬라이싱된 클래스 다이어그램만을 제약 충족 문제(Constraint Satisfaction Problem)로 변환하여 검증을 진행하였다. 하지만 본 연구가 SoS 모델상에 발생한 변화를 효율적으로 검증하기 위해 UML 기반 SoS 목표 모델에 슬라이싱 기법을 적용한것과 달리, 기존 연구들은 다중 에이전트 시스템 모델 자체, 혹은 시스템의 UML 모델 자체에 슬라이싱을 적용하고, 해당 슬라이싱 과정을 더 효율적으로 진행하는 것에 초점을 맞추고 있다.

3. SoS 목표 모델 슬라이서



[그림 1] SoS 목표 모델 슬라이서 실행 과정

본 연구에서 제안하는 목표 모델 슬라이서는 [그림1]과 같은 형태로 진행된다. 먼저, 입력은 트리 형태의 SoS 목표 모델과 SoS 모델에 발생한 변화와 직접적으로 관련된 하위시스템 수준 목표의 이름이다. SoS 목표 모델 슬라이서는 3단계의 과정을 거치며 변화와 관련된 목표들만을 추출하게 된다.

1) 슬라이싱 기준 (Slicing Criteria) 생성

먼저 입력으로 들어온 변화 관련 목표의 이름과 SoS 목표 모델을 분석하여 슬라이싱 기준을 생성한다. 본 연구를 적용한 재난 대응 SoS 시나리오에서는 각 목표가 통계적 검증 도구인 PRISM[7]에서 사용되는 수식인 PCTL(Probabilistic Computation Tree Logic) 형식의 검증 명세를 가진다. 따라서 입력으로 주어진 변화 관련 목표에 대한 PCTL 명세 상의 변수들을 추출하는 방식으로 슬라이싱 기준을 생성하였다.

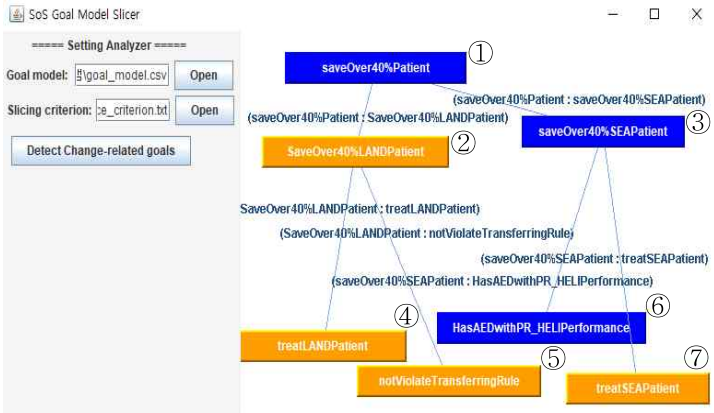
2) 슬라이싱 시작점(Slicing Starting Point) 설정

그 뒤에는 슬라이싱을 위한 시작점을 설정해야 한다. 슬라이싱 시작점은 입력으로 주어진 변화 관련 목표의 이름과 같은 이름을 가지는 SoS 목표 모델상의 목표로 설정하며, 이때 시작점으로 설정된 목표는 SoS 목표 모델상의 잎새 노드(Leaf Node) 중 하나이다.

3) 역방향 슬라이싱(Backward Slicing) 진행

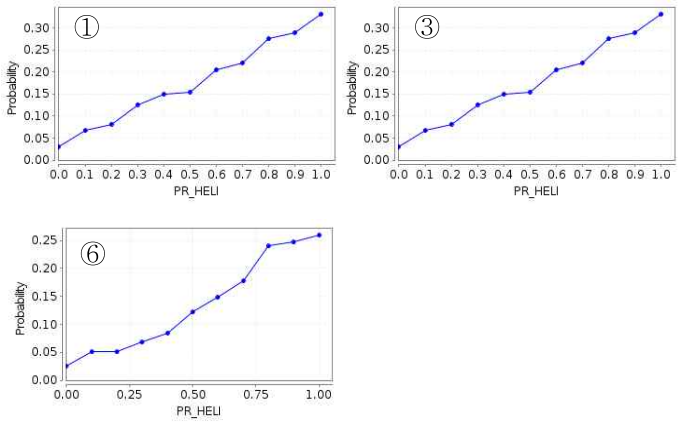
슬라이싱 시작점과 슬라이싱 기준을 이용하여 상위 목표들에 대한 역방향 슬라이싱을 진행한다. 기존 연구에서 데이터 흐름(Data Flow)상에 존재하는 변수를 슬라이싱 했던 것과 다르게 [7], 본 연구에서는 각 목표에 존재하는 목표의 명세를 이용하였으며, 본 연구를 적용한 재난 대응 SoS에서는 각 목표의 PCTL 명세를 이용하였다. 최상위 목표인 SoS 목표까지 역방향 슬라이싱을 진행하게 되면 슬라이싱은 종료되며, 슬라이싱된 SoS 목표 모델을 출력한다.

아래는 앞서 설명한 SoS 목표 모델 슬라이서를 재난 대응 SoS에 적용한 예시에 대한 설명이다. 먼저 SoS 목표 모델은 [그림 2]와 같이 최상위의 SoS 수준 목표를 시작으로, 각 CS가 달성하는 CS 수준의 목표들과 해당 CS를 구성하는 하위 시스템(Subsystem) 수준의 세부 목표들의 트리 형태로 구성하였다. SoS 목표 모델의 구현은 UML 모델을 이용하였으며, 각 목표는 “목표 이름, 목표를 만족시키는 대상 시스템, PCTL 형식으로 명세된 검증 명세, 상위 목표, 하위목표, 목표 상세설명 정보” 를 가지고 있다. 이 모델에서 최상위 SoS 수준의 목표는 전체 구조한 환자가 전체 발생 환자의 40% 이상임을 나타내는 ‘① saveOver40%patient’ 이다. 그 아래는 CS 수준의 목표들을 나타내며, 각각 바다와 육지에서 구조한 환자 수에 대한 목표인 ‘②saveOver40%LANDPatient’ 와 ‘③ saveOver40%SEAPatien-t’ 이다. 마지막으로 하위 시스템(Subsystem) 수준 목표들은 ②의 목표에 대해 ‘④ treatLANDPatient’ 와 ‘⑤ notViolate-TranserringRule’ , 그리고 ③ 목표에 대해 ‘⑥ HasAEDwith-PR_HELIPerformance’ 와 ‘⑦ treatSEAPatient’ 로 구성된다.

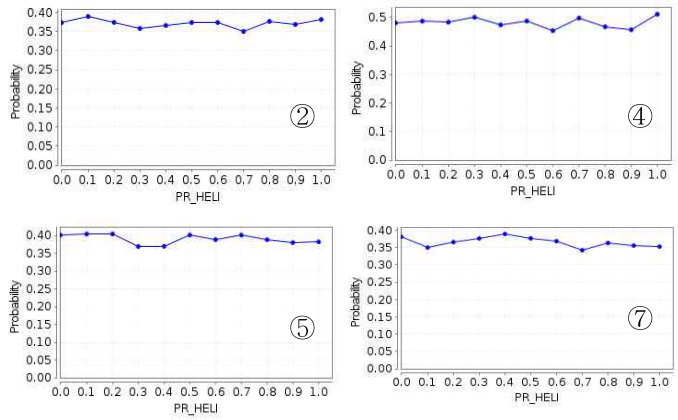


[그림 2] SoS 목표 모델 및 슬라이싱 결과

SoS 목표 모델 슬라이서는 [그림2]와 같은 형태의 SoS 목표 모델과 SoS 모델에서 발생한 변화와 관련된 목표의 이름을 입력으로 받는다. 그 뒤, 앞서 기술한 1)~3)의 과정을 거치며 슬라이싱을 진행하게 된다. 먼저 입력된 변화 관련 목표의 PCTL 명세를 분석하여 변화와 관련 있는 슬라이싱 기준 변수를 추출하는데, [그림2]의 목표 모델과 변화 관련 목표로 “⑥ HasAEDwithPR_HELIPerformance” 이 입력되면 ⑥의 목표가 가지고 있는 PCTL 명세를 분석한다. 해당 목표의 PCTL 명세는 P=? [true U<=10000 (num_of_patient_SEA>=45)]이다. 이때, 슬라이싱 기준 변수로 “num_of_patient_SEA” 를 추출하게 된다.그 뒤, 슬라이싱 시작점을 입력된 목표인 “⑥ HasAEDwith-PR_HELIPerformance” 을 설정한다. 역방향 슬라이싱은 ⑥-③-①의 순서로 진행되며, SoS 수준 목표인 “① saveOver40%-patient” 까지 확인하게 되면, 슬라이싱 된 SoS 목표 모델을 출력하게 된다. [그림2]에서 파란색으로 표시된 목표들이 실제 SoS 모델 슬라이서[1]의 결과로 출력된 변화 관련 목표들을 나타낸다.



[그림 3] 슬라이싱된 목표들의 검증 결과



[그림 4] 슬라이싱되지 않은 목표들의 검증 결과

4. 실험 및 결과

4.1 실험 환경

본 연구에서 제안하는 기법을 평가하기 위해 사용된 SoS 모델과 시나리오는 송지영 외[8] 연구에서 사용된 MCI 재난 대응 시나리오이다. 바다와 육지에서 발생한 MCI 상황에 대해 해상 인명 구조를 위해 5대의 헬기가, 지상 인명구조를 위해 10대의 구급차가 투입된 시나리오를 대상으로 한다. 실험은 Windows 8.1 (64-bit), java jdk v1.8.0 환경에서 PRISM v4.3.1을 이용하여 진행되었다. PRISM[7]은 확률적 특징을 가지는 시스템을 형식 모델링 및 검증(Formal Modeling & Verification)하는데 사용되는 대표적인 도구이다. PRISM에서는 다양한 확률적 모델을 구현할 수 있으며, 해당 모델들에 대해 통계적 모델 검증을 수행할 수 있다. 이 실험에서 SoS 모델은 PRISM을 이용하여 구현되었고, 통계적 검증방식을 이용하였다. 이때 SoS 모델에 삽입된 변화는 해상인명구조 역할을 진행하는 헬기에 AED 키트와 체온 유지 기구가 제공되어, 이와 관련된 변수인 헬기의 구조율(PR_HELI)이 바뀌는 것이다. 그리하여 SoS 모델에서 PR_HELI의 입력값을 조절해가며 [그림2]에 있는 7개의 검증 목표들에 대해 검증을 진행하였다.

1) https://github.com/SESoS/SIMVA-SoS

4.2 실험 결과

7개의 검증 목표에 대한 검증 결과는 크게 두 가지 경향을 나타내었다. SoS 목표 모델 슬라이서를 통해 슬라이싱된 3개의 목표들은 [그림3]과 같이 변화의 입력값이 달라짐에 따라 검증 결과도 달라지는 결과 보였고, 슬라이싱 과정에서 제거된 4개의 검증 목표들은 [그림 4]와 같이 변화의 입력값이 달라짐에도 불구하고 검증 결과는 크게 달라지지 않는 모습을 보였다. 이를 통해 SoS 목표 모델 슬라이싱 기법이 SoS 모델의 변화와 관련된 목표들만 슬라이싱 한다는 것을 확인할 수 있었다.

5. 결론

대규모 복잡 시스템인 SoS는 다양한 요인들에 의해 변화한다. 이런 변화에 대응하여 SoS 모델에 대한 효율적인 검증을 제안하기 위해 본 연구에서는 SoS 목표 모델 슬라이싱 기법을 제안한다. 변화 관련 목표와 SoS 목표 모델을 입력으로, 본 알고리즘은 해당 변화와 관련된 SoS 목표들만을 출력한다. 이는 SoS 모델 검증의 효과성에는 크게 영향을 주지 않으면서도 검증 시간을 감소시켜 더 효율적인 검증을 가능하게 한다. 향후 연구로는 SoS 모델의 변화로부터 관련된 시스템 수준 목표를 찾아내는 기법을 자동화하는 연구를 진행할 예정이다.

6. 참고문헌

[1] J. Boardman et al, “System of Systems -the Meaning of Of,” IEEE/SMC International Conference on System of Systems Engineering (SoSE), 2006.

[2] H. Eric. “Verification and Validation Issues in Systems of Systems.“ arXiv preprint arXiv:1311.3626 (2013).

[3] D. Seo, et al, “Modeling and verification for different types of system of systems using prism,” in Software Engineering for Systems-of-Systems (SESoS), 2016 IEEE/ACM 4th International Workshop on. IEEE, 2016, pp. 12-18.

[4] R. H. Bordini et al, “Property-based slicing for agent verification,” Journal of Logic and Computation, vol. 19, no. 6, pp. 1385-1425, 2009.

[5] M. Winikoff et al “Slicing Agent Programs for more Efficient Verification”.

[6] A. Shaikh et al “UMLtoCSP (UOST): a tool for efficient verification of UML/OCL class diagrams through model slicing”. In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (p. 37). ACM. (2012, November).

[7] M. Kwiatkowska et al, “Prism 4.0: Verification of probabilistic real-time systems,” in Computer aided verification. Springer, 2011, pp. 585-591.

[8] Song, J et al, “SoS GaP Slicer: Slicing SoS Goal and PRISM Models for Change-Responsive Verification of SoS”, In 2017 24th Asia-Pacific Software Engineering Conference (APSEC) (pp. 546-551). IEEE (2017, December)