

시스템 오브 시스템즈의 오류 위치 추정을 위한 통계적 검증 결과 활용 기법 분석*

현상원^o 신용준 배두환
한국과학기술원 전산학부
{swhyun, yjshin, bae}@se.kaist.ac.kr

Analysis of Utilization Methods of Statistical Model Checking Results for Localizing Faults on System of Systems

Sangwon Hyun^o Yong-Jun Shin Doo-Hwan Bae
School of Computing, KAIST

요 약

시스템 오브 시스템즈(System-of-Systems, SoS)는 독립적인 이종의 구성시스템들로 이루어진 크고 복잡한 시스템이며, 이런 SoS의 목표 달성도 확인을 위해 통계적 모델 검증(Statistical Model Checking, SMC) 기법이 주로 사용되어진다. 하지만 SoS가 특정 목표를 달성하지 못하는 경우, 그 원인을 찾는 것이 단일 시스템에서 오류를 찾는 것보다 더 큰 비용이 소모된다. 이러한 목표 달성 저해 요소를 효율적으로 찾기 위한 방법으로 기존 오류 위치 추정 기법을 확장 적용할 수 있다. 하지만, 이때 SoS의 목표달성도인 SMC의 확률적 결과값에 대한 추가적인 해석 기법이 필요하다. 따라서 본 연구에서는 SoS에 기존의 오류 위치 추정기법을 적용하기 위해 SMC결과를 활용하는 5가지 기법을 소개하고, 재난 대응 SoS 시나리오를 이용하여 해당 기법들에 대한 비교 분석을 진행하였다. 그리고 위치 추정의 효율성을 기반으로 기법들을 비교한 결과, SMC의 결과를 기댓값에 기반하여 활용한 기법이 가장 좋은 성능을 보임을 밝힐 수 있었다.

1. 서 론

시스템 오브 시스템즈(System-of-Systems, 이하 SoS)는 개별 시스템들로는 달성할 수 없는 상위수준의 목표를 독립적인 이종의 시스템들의 상호작용을 통하여 달성할 수 있도록 하는 대규모 복잡 시스템이다[1]. SoS의 대표적 예시로 근래에 많은 관심을 받고 있는 사이버-물리 시스템(Cyber Physical System)을 비롯하여, 대형 재난 사고(Mass Casualty Incident, 이하 MCI)에 대응하기 위한 재난 대응 시스템, 그리고 5G, 사물인터넷을 기반으로 하는 스마트 시티 등을 들 수 있다.

D. Seo 외[2] 연구와 L. Axel외[3]의 연구와 같은 SoS 연구 논문들에서는 위와 같은 SoS의 목표 달성도 검증을 위한 기법으로 통계적 모델 검증(Statistical Model Checking, 이하 SMC) 기법을 사용하였다. SoS의 검증에 SMC 기법을 사용하는 가장 큰 이유는 기존에 사용되는 모델 검증 기법에서 발생하는 상태 급증 문제(State explosion problem)를 완화하기 위해서이다. 하지만 SMC 기법을 이용한 SoS 검증의 단점은 해당 시스템이 특정한 목표를 달성하는지에 대한 정확한 예측(Accurate prediction)이 아닌 근사적 예측(Approximate prediction)을 수행한다는 것으로, 그 결과가 시스템이 특정 목표를 달성할 확률값으로 나온다는 점이다[4].

만약 SoS 모델이 특정 목표를 달성하지 못해서 그 원인을 찾아내야 하는 경우, 크고 복잡한 시스템인 SoS에서 목표 달성 저해 요소의 원인을 매뉴얼하게 찾는 것은 너무 큰 시간적 비용을 필요로 한다. 이러한 비용을 줄이기 위해, 기존 소프트웨어공학분야에서 연구되는 자동화된 오류 위치 추정기술(Fault Localization)의

적용이 한 해결책이 될 수 있다. 하지만 기존 위치 추정 기술들은 목표 프로그램에 테스트 케이스를 실행한 결과인 True/False의 값을 입력받는 반면, SMC를 이용한 SoS의 검증 결과는 True/False의 값이 아닌 확률적 예측값을 가진다. 따라서 본 연구에서는 SoS 모델의 통계적 모델 검증(SMC) 결과를 오류 위치 추정기술에 적용 가능하도록 활용하는 기법들을 소개하고, 해당 기법들에 대한 실험 결과 및 분석을 기술한다.

본 연구의 구성은 다음과 같다. 2장에서 관련 연구들을 소개한 뒤, 3장에서 오류 위치 추정기술에 대한 배경 지식을 서술하고, 4장에서 SMC의 결과를 변환하는 기법들을 소개한 뒤, 5장에서 실험적으로 어떤 기법이 SMC의 결과를 가장 잘 해석하여 변환하였는지를 보이고, 6장에서 결론 및 향후 연구를 기술한다.

2. 관련 연구

G. Ross외 [8]의 연구는 크고 복잡한 시스템에서 발생하는 창발적 실패(Emergent failure)의 원인을 추정하기 위해 스펙트럼 기반 오류 위치 추정 기술 (Spectrum-Based Fault Localization, 이하 SBFL)을 이용하였다. 해당 연구에서는 창발적 실패의 원인을 크게 확률적 원인과 시스템 내부의 원인으로 구분한 뒤, 각 경우에 대한 기법을 제시하였다. 확률적 원인에 대해서는 기존의 SBFL기술에 Fuzzy passing 기법을 적용하여 하나의 테스트 케이스가 성공/실패의 두 가지 결과를 가질 수 있도록 하였으며, 시스템 내부의 원인은 시스템 계층에 추가적인 스케줄 계층을 추가함으로써 해결하였다. 하지만 예시로 사용되어진 시스템이 SoS 수준의 시스템이라고 할 수 없으며, 단순한 테스트 케이스를 이용하였다는 점에서 본 연구와는 다른 타겟 도메인을 가진다.

Y. J. Shin외 [6]의 연구는 SoS가 특정 목표를 수행하지 못했을 때, 그 원인을 찾기 위해 SBFL을 적용하는 기법을 제안했다. 위 연구는 특정 시나리오에서 참여한 SoS의 구성 시스템

* 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No. 2015-0-00250, (SW 스타랩) 모델 기반의 초대형 복잡 시스템 분석 및 검증 SW 개발)과 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임 (2017M3C4A7066212)

(Constituent Systems, 이하 CS)과 그 상호작용을 협력 그래프(Collaboration graph)의 형태로 표현하였다. 그리고 SMC의 결과와 협력 그래프를 입력값으로 SBFL을 적용하여, SoS에 삽입된 오류 CS 및 상호작용의 위치를 추정하는 데 성공하였다. 본 연구에서는 해당 연구에서 사용한 SMC 결과 활용 기법을 5가지 활용 기법 중 한 가지로 사용한다.

3. 배경

소프트웨어의 오류 위치 추정기술은 테스트 케이스의 실패와 그 실패의 원인으로 추정되는 프로그램 유닛의 인과적 관계를 추론하는 기술이다[5]. 대표적으로 사용되는 오류 위치 추정 기술은 SBFL 기법이며, 프로그램 스펙트럼이란 테스트 케이스를 실행할 때 실제 프로그램에서 실행된 프로그램 유닛(코드, 분기문, 등)의 집합을 나타낸다.

Statements	Test cases					*Sus	Rank
	t1	t2	t3	t4	t5		
Input: a,b							
1. if a < 0 :	●	●	●	●	●	0.4	3
2. if b > map_size() :	●					0.2	4
3. print ("Error Input")	●					0.2	4
4. else :			●		●	0.8	1
5. Assign(a,b.10) #BUG			●		●	0.8	1
6. else:		●		●		0.1	5
-							
Testing result	S	S	F	S	F		

[그림 1] 지도에 값을 할당하는 프로그램에 대한 SBFL 예시

위 그림은 프로그램 코드 수준의 SBFL 기법의 예시를 보여준다. 타겟 프로그램은 특정 크기의 지도에 값을 할당하는 프로그램으로, a,b에 대한 입력값을 받는다. 그리고 5번째 코드에 0보다 작은 좌표에 값을 입력하는 오류가 삽입되어있다. [그림 1]에서는 총 5개의 테스트 케이스가 사용되었고, 각 테스트마다 스펙트럼과 함께 그 실행 결과를 갖고 있다. 이와 같은 데이터를 이용하여 각 코드에 대한 의심도를 계산하고, 이를 기반으로 의심되는 코드의 순위를 선정할 수 있다.

SBFL을 SoS 모델에 확장 적용하는 과정은 위 [그림 1]에서 코드 부분을 SoS의 CS들과 CS들의 상호작용으로, 테스트 케이스와 그 결과를 특정 시나리오와 해당 시나리오에 대한 SMC의 결과로 바꾸는 방식이다[6]. 이를 통해 SoS의 목표 달성 저해 원인을 찾는 데 SBFL을 적용하여 모든 CS들과 CS들의 상호작용에 대한 의심도 순위를 계산할 수 있다.

4. SMC 결과 변환 기법

An SMC result of S on δ with Scenario₁

Set of SMC results = $\{p_1, p_2, \dots, p_k\}$

S : SoS Model
 δ : Verification property

- 1) SMC results 집합의 각 p_i 를 이용하여 Success = p_i , Failure = $1 - p_i$ 로 할당
- 2) SMC results 집합의 1분위 값 미만의 p_i 를 Failure로 간주
- 3) SMC results 집합의 2분위 값 미만의 p_i 를 Failure로 간주
- 4) SMC results 집합의 3분위 값 미만의 p_i 를 Failure로 간주
- 5) SMC results 집합을 k-means 알고리즘으로 clustering 했을 때 낮은 평균 값을 갖는 Cluster에 속한 p_i 를 Failure로 간주

[그림 2] SMC 검증 결과 변환 기법

본 장에서는 SoS의 목표 달성 저해 요소 위치 추정을 위한 SBFL 적용을 위해, SMC의 확률적 결과를 SBFL에서 사용할 수 있는 Success, Failure값으로 활용할 수 있는 5가지의 기법을 소개한다.

위 [그림 2]는 본 연구에서 실험할 5가지 SMC 결과의 해석/

변환 기법을 요약하여 보여준다. 기법 1)은 직관적으로 SMC의 결과를 이용하는 기법이다. Y. Shin의 [6] 연구에서 제시한 이 기법은 SMC의 결과를 있는 그대로 해석하여 Success = p, Failure = 1-p의 값을 할당한다. 이는 해당 SMC의 결과를 100번의 시행을 한다고 가정했을 때 p*100번 성공하고, (1-p)*100번 실패할 것이라는 기댓값에 기반한 해석 기법이다.

기법 1)을 제외한 기법 2)~5)는 SMC 결과들의 분포를 이용한 통계적 해석 기법들이다. 기법 2)~4)는 각각 SMC 결과 집합의 1분위, 2분위, 3분위값을 기준으로 해당 분위값보다 작은 SMC 결과들을 실패라고 간주하는 기법이다. 예를 들어 기법 2)를 적용한다면, 1분위 값을 27%라 가정할때, 27%보다 작은 SMC 결과들을 실패로 간주하고 27% 이상의 SMC 결과들을 성공으로 간주한다. 기법 2)~4)는 SMC 결과 분포에 대한 통계적인 성공/실패의 경계선을 각 분위값을 기반으로 설정하는 기법이다. 본 연구에서 1-sigma, 2-sigma, 3-sigma 범칙이 아닌 각 분위값에 기반한 SMC 결과 분포의 경계선을 제시한 이유는 SMC 결과의 분포가 정규분포를 반드시 따른다고 할 수 없기 때문이다. 특정 확률 분포가 정규분포를 따르지 않는 경우, 특정 신뢰구간에 대한 경계값을 분석적으로 구하는 게 불가능하다. 따라서 본 연구에서는 sigma를 이용하여 특정 구간에 대한 경계값을 구하는 기법 대신 각 분위값을 기반으로 하는 경계값을 사용하였다.

마지막으로 기법 5)는 SMC의 결과를 확률값 p에 따라 K-means[7] 알고리즘으로 클러스터링하여 각 클러스터 수준으로 성공/실패를 간주하는 기법이다. 이때, 상대적으로 낮은 평균 값을 갖는 클러스터를 실패 클러스터로 정한다.

5. 실험 및 결과

5-1. 실험 환경

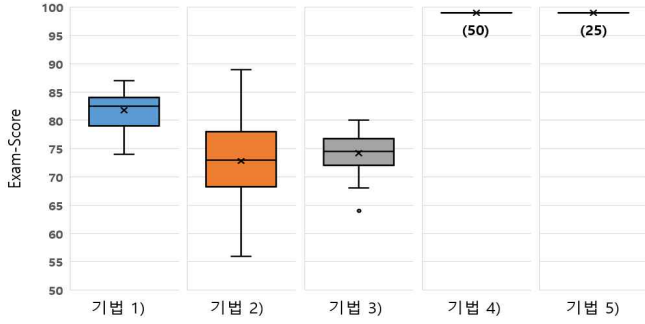
앞서 설명한 5가지 SMC 결과의 해석 기법을 적용하기 위해 본 연구에서는 Y. Shin의[6] 연구에서 사용한 MCI 대응 SoS 및 시나리오를 사용하였다.

해당 SoS에는 총 10대의 로봇, 10대의 드론과 100개의 커뮤니케이션 채널이 구현되어있으며, 한 번의 시나리오를 실행할 때마다 3개의 로봇, 3개의 드론, 그리고 3개의 커뮤니케이션 채널을 랜덤하게 선택한 후 100명의 환자가 발생한 구조 현장으로 투입한다. 이때, SoS 모델에 존재하는 로봇, 드론, 커뮤니케이션 채널에 각각 구조율, 환자 인식률, 지연 시간의 변수들에 대한 인위적 오류를 삽입할 수 있다. 또한 SoS의 구성 시스템들이 이중의 시스템이라는 특징을 보이기 위해 같은 로봇, 드론에서도 구조율과 환자 인식률을 85%~100% 범위에서 랜덤하게 할당한다.

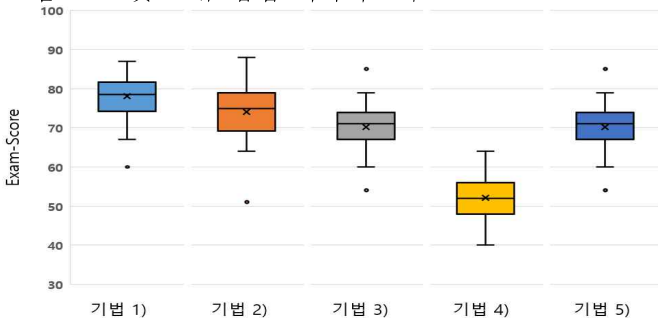
본 연구에서는 앞서 설명한 재난 대응 SoS에 로봇, 드론, 커뮤니케이션의 세 가지 오류를 삽입한 경우에 대하여 각각 200번의 랜덤 시나리오에 대한 SMC를 진행하고, 이 과정을 총 30번 반복하였다. 즉, 로봇 2대의 구조율 감소, 드론 2대의 인식률 감소, 커뮤니케이션 채널 2개의 지연 시간 증가라는 3가지 오류 삽입 경우들에 대해 각각 6,000 개의 SMC 결과를 추출하였다. 그리고 해당 SMC 데이터들에 대해 앞서 설명한 5가지 기법으로 변환한 값들을 이용하여 각각 SBFL 알고리즘을 적용하고, 그 결과를 비교하였다. 이때 해당 SoS 모델에 대한 통계적 검증과정은 SIMVA- SoS[9] 도구를 기반으로 진행하였다. SIMVA-SoS는 해당 재난 대응 SoS 모델을 직접 실행하며, 달성 목표에 대한 시뮬레이션 기반 통계적 검증을 수행한다.

5-2. 실험 결과 및 분석

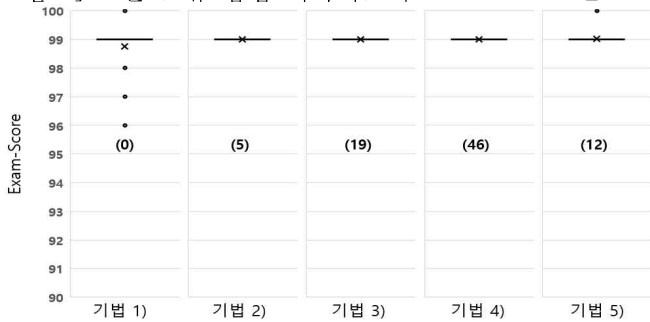
아래 [그림 3]부터 [그림 5]는 각각 로봇, 드론, 커뮤니케이션 오류 삽입 경우들에 대한 SMC 결과를 5가지 기법으로 변환하여 SBFL 알고리즘을 적용한 결과의 분포를 보여준다.



[그림 3] 로봇 오류 삽입 시나리오의 Exam-score 분포



[그림 4] 드론 오류 삽입 시나리오의 Exam-score 분포



[그림 5] 커뮤니케이션 오류 삽입 시나리오의 Exam-score 분포

본 실험에서 SBFL 알고리즘으로는 Tarantula를 사용하였고, 그 결과를 비교하기 위해 위치 추정 알고리즘에서 그 효용성을 보이는데 주로 사용되는 EXAM-score[6]를 이용하였다. 그리고 아래 그림들에서 100점의 EXAM-score를 달성한 기법들의 경우 괄호로 공동 1위의 개수를 표시하였다.

위 그림들로부터 해석할 수 있는 점은 1) 기법을 이용하여 SMC의 결과를 변환했을 때의 결과가 다른 기법들을 적용했을 때 보다 상대적으로 좋은 SBFL 알고리즘의 성능을 보였다는 점이다. 로봇 오류 삽입 시나리오의 경우 기법 4)와 기법 5)를 이용한 결과가 월등하게 좋게 나온 것처럼 보이지만, 실제 결과를 확인해본 결과 의심도 값 1을 가지는 공동 1위가 각각 평균 50개, 25개 발견되었다. 이 같은 결과는 개발자로 하여금 오류의 위치를 추정하는데 도움을 주지 못하기 때문에 좋은 결과라 해석할 수 없다. 드론 시나리오에서는 5가지 기법 중 1) 기법의 결과가 다른 4가지 기법들보다 좋다는 것을 알 수 있으며, 커뮤니케이션 시나리오에서는 모든 기법들을 적용하였을 때 모두 삽입된 오류가 1,2순위였다. 하지만 1) 기법에서는 공동 1위가 존재하지 않았던 반면, 2) 기법부터 5) 기법까지 공동 1위의 개수가 평균 5개, 19개, 46개, 12개가 존재하였다.

이와 같은 결과가 발생하는 이유를 분석하자면, 첫째로 기법 1)이 SMC의 결과를 기댓값의 형태로 SBFL 알고리즘에 입력함으로써 SMC의 결과를 가장 잘 나타낼 수 있었다는 점이며, 두 번째는 통계적으로 성공, 실패의 영역을 나누는 기법들에서

가정한 “오류가 삽입된 경우의 SMC 결과가 상대적으로 낮을 것”이라는 명제가 SoS를 구성하는 이중 시스템들의 상호작용에 의해 부정되었다는 점이다. 이는 오류를 삽입한 단일 시스템에 의해 발생하는 성능 저하 뿐만 아니라, 이중 시스템들의 창발적 상호작용에 의해서도 예상치 못한 성능 저하가 발생될 수 있다는 반증으로 해석할 수 있다.

6. 결론

대규모 복잡 시스템인 SoS는 통계적 모델 검증 기법을 이용하여 특정 목표의 달성도를 확인할 수 있다. 하지만 특정 목표 달성에 실패하여 SoS에 존재하는 오류의 위치를 추정해야 하는 경우 기존의 위치 추정기술의 적용이 필요하며, 본 연구에서는 SBFL을 SoS 모델에 확장 적용하는 과정에서 필요한 SMC의 결과 해석 방안 5가지를 소개하였다. 그리고 MCI 대응 SoS 모델 및 시나리오를 이용한 실험을 통해 이 기법들 중 SMC의 결과를 기댓값에 기반하여 할당한 기법 1)이 상대적으로 좋은 성능을 보임을 밝힐 수 있었다.

하지만 상대적으로 복잡하지 않은 시나리오에 대한 단일 실험만으로 기댓값 기반의 기법이 항상 좋은 성능을 보인다고는 단정지을 수 없으며, 기댓값 기반의 기법은 SBFL 이외의 위치 추정기법을 SoS에 적용하는데 있어 한계점을 가진다. 따라서 더 높은 복잡성을 가지는 SoS 모델 및 시나리오를 이용한 실험을 추가적으로 진행할 예정이며, 해당 기댓값 기반의 기법을 다른 위치 추정기술에 적용하는 연구를 진행할 예정이다.

7. 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] 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.
- [3]: L. Axel et al. “Verifying Systems-of-Systems with Statistical Model Checking.” ERCIM News 103 (2015).
- [4]: K. Kenan et al. “Feedback control for statistical model checking of cyber-physical systems.” International Symposium on Leveraging Applications of Formal Methods. Springer, Cham, 2016.
- [5]: W. E. Wong et al. “A survey on software fault localization.” IEEE Transactions on Software Engineering 42.8 (2016): 707-740.
- [6]: Y. J. Shin et al., “Spectrum-Based faultlocalization on a collaboration graph of a System-of-Systems,” in201914th Annual Conference System of Systems Engineering (SoSE) (SoSE2019), Anchorage, USA, May 2019.
- [7]: Hartigan et al. “Algorithm AS 136: A k-means clustering algorithm.” Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.
- [8]:G. Ross. “ISOLATING THE CAUSES OF EMERGENT FAILURES IN COMPUTER SOFTWARE.” Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach (2018): 263-284.
- [9]:<https://github.com/SESoS/SIMVA-SoS>