

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

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

현 상 원 [†] 신 용 준 [†] 배 두 환 ^{**}
(Sangwon Hyun) (Yong-jun Shin) (Doo-Hwan Bae)

요약 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 사용된다. 하지만 SoS가 특정 목표를 달성하지 못하는 경우, SoS내에 존재하는 목표 달성 저해 요소를 효율적으로 찾기 위해 기존 오류 위치 추정 기법을 확장 적용할 수 있다. 하지만, 기존 기법들은 성공, 실패의 테스트 결과만을 사용하기 때문에, SMC의 확률적 결과값을 기존 위치 추정 기법에 사용하기 위한 추가적인 해석 기법이 필요하다. 따라서 본 연구에서는 SoS에 기존의 오류 위치 추정기법을 적용하기 위한 6가지 기법을 소개하고, 재난 대응 SoS를 이용하여 해당 기법들에 대한 비교 분석을 진행하였다. 그 결과, SMC의 결과를 기법값으로 활용한 기법이 모든 시나리오에서 평균적으로 좋은 성능을 보임을 밝힐 수 있었다.

키워드: 시스템 오브 시스템즈, 오류 위치 추정, 통계적 모델 검증, 재난 대응 SoS

Abstract System of Systems (SoS) is a large and complex system comprising independent constituent systems. Statistical Model Checking (SMC) techniques can be used to verify if the SoS achieves its goals or not. However, even if the SoS goal failure is detected using the SMC, finding a root cause of the SoS failure requires more cost than that of a system. One of the candidate solutions for reducing the debugging cost is to apply fault localization techniques on the SoS. However, existing fault localization techniques are designed to utilize testing results of a system. Thus, a method to utilize SMC results is needed to apply existing fault localization techniques to the SoS. In this study, we suggest six utilization methods of SMC results for SoS fault localization, and compare the performance of them on the emergency-response SoS. We found that the method based on the expectation value showed the best performance in the experiment.

Keywords: system of systems, fault localization, statistical model vhecking, emergency-response SoS

- 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2015-0-00250, (SW 스타랩) 모델 기반의 초대형 복잡 시스템 분석 및 검증 SW 개발)
- 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대 정보 컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(NRF-2017M3C4A7066212)
- 이 논문은 2019 한국컴퓨터종합학술대회에서 '시스템 오브 시스템즈의 오류 위치 추정을 위한 통계적 검증 결과 활용 기법 분석'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 한국과학기술원 전산학부 학생(KAIST)
swhyun@se.kaist.ac.kr
(Corresponding author인)
yjshin@se.kaist.ac.kr

^{**} 종신회원 : 한국과학기술원 전산학부 교수
bae@se.kaist.ac.kr

논문접수 : 2019년 8월 23일
(Received 23 August 2019)
논문수정 : 2019년 11월 14일
(Revised 14 November 2019)
심사완료 : 2020년 1월 30일
(Accepted 30 January 2020)

Copyright©2020 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지 제47권 제4호(2020. 4)

1. 서론

시스템 오브 시스템즈(System-of-Systems, 이하 SoS)는 개별 시스템들로만 달성할 수 없는 상위수준의 목표를 독립적인 이종 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 검증 기법들(Exhaustive Model Checking)에서 발생하는 상태 급증 문제(State explosion problem)를 완화하기 위해서이다. 상태 급증 문제란 검증을 진행하는 모델의 크기와 복잡도에 의해 검증하는 모델의 상태 집합이 기하급수적으로 커지는 현상을 말한다. 하지만 SMC 기법은 시뮬레이션 기반의 실행 기록(Execution trace)을 이용하여 해당 모델에 대한 검증을 진행하기 때문에, 이러한 상태 급증 문제를 완화할 수 있다. SMC 기반 SoS 검증은 해당 시스템이 특정한 목표를 달성하는지에 대한 정확한 예측(Accurate prediction)이 아닌 근사적 예측(Approximate prediction)을 수행하며, 시스템의 목표 달성 여부의 검증 결과를 확률 형태로 출력한다[4].

만약 SoS 모델이 특정 목표를 달성하지 못함이 검증되고, 그 원인을 찾아내야 하는 경우, 크고 복잡한 이종의 시스템들로 구성된 SoS에서 목표 달성 저해 요소의 원인을 매뉴얼하게 찾는 것은 너무 큰 시간적 비용을 필요로 한다. 이러한 비용을 줄이기 위해, 기존 소프트웨어공학 분야에서 연구되는 자동화된 오류 위치 추정 기술(Fault Localization)의 적용이 한 해결책이 될 수 있다. 하지만 SMC를 이용한 SoS 통계적 검증 결과를 기존 위치 추정 기술에 적용하는데에는 어려움이 있다. 기존 위치 추정 기법들은 타겟 프로그램에 테스트 케이스를 실행한 결과인 Success/Failure의 값을 입력받는 반면, SMC를 이용한 SoS의 검증 결과는 Success/Failure의 값이 아닌 확률적 예측값을 출력한다. 즉 기존의 오류 위치 추정 기술을 SoS에 적용하기 위해, SMC의 확률적 검증 결과를 변환하기 위한 기법이 필요하다. 따라서 본 연구에서는 SoS 모델의 통계적 모델 검증(SMC) 결과를 오류 위치 추정 기술에 적용 가능하

도록 활용하는 6가지 기법들을 소개하고, 해당 기법들을 다양한 위치 추정 알고리즘에 적용하여 그 효용성을 판단한다. 본 연구에서는 재난 대응 SoS에 대한 실험 결과 및 분석을 기술한다.

본 연구의 구성은 다음과 같다. 2장에서 관련 연구들을 소개한 뒤, 3장에서 오류 위치 추정 기술에 대한 배경 지식을 서술하고, 4장에서 SMC의 결과를 변환하는 기법들을 소개한다. 5장에서는 본 연구에서 제안한 6가지 해석 기법을 MCI-R SoS 시나리오에 적용하고, 3가지 오류 위치추정 알고리즘을 통해 그 효용성을 분석한다. 마지막으로, 6장에서 결론 및 향후 연구를 기술한다.

2. 관련 연구

SoS와 같은 크고 복잡한 시스템에서 발생하는 실패와 그 실패의 원인이 되는 오류를 분석하는 연구들은 분석에 사용한 기법들에 따라 크게 두 부류로 나눌 수 있다. 첫째는, 기존 시스템 분야에서 연구되던 Fault diagnosis 기법을 적용한 연구들이다. Fault diagnosis 기법은 시스템의 각 컴퍼넌트들에서 모니터링을 수행하여 특정 테스트들을 통과하지 못한 컴퍼넌트 모듈을 추정하는 기법이다. H. Jarrah외 [11]의 연구는 Mobile Ad-hoc Network(MANET) 기반의 동적 시스템에서 적용 가능한 Fault diagnosis 기법들에 대해 서술하였다. 이때 시스템 모델은 그래프의 형태로 구성되며, 각 노드와 엣지는 시스템 컴퍼넌트들과 컴퍼넌트들 사이의 논리적 연결을 나타낸다. Diagnosis 과정은 오류가 없다고 가정된 노드에서 테스트 요구 메시지를 모든 노드들에게 전달하고 각 노드들이 해당 테스트의 결과를 공유하여, 최종적으로 테스트 결과가 다른 노드들을 현재 오류를 가지고 있는 상태로 판단한다. L. Qin외 [12]의 연구는 군집 드론, 군집 차량 시스템과 같은 Swarm system에서 오류를 찾아내는 기법들에 대해 서술하였다. 해당 논문에서는 중앙집중형(Centralized), 계층형(Hierarchical), 분산형(Distributed) 아키텍처를 기반으로 하는 Fault diagnosis 기법들을 swarm system에 적용하여 실험하고, 각 기법들의 장단점을 분석하였다. 하지만 위에서 서술한 연구들의 타겟 시스템들은 본 연구에서 집중하고자 하는 SoS와 추상화의 수준이 맞지 않다. 따라서 구성 시스템의 다양성, 동적 재구성과 같은 SoS의 특징들을 고려하지 않았기 때문에, SoS에 적용하기 어렵다는 한계점을 가지고 있다. 또한, 전체 시스템의 검증이 아닌 구성 시스템들 각각의 테스트 결과만을 이용한다는 차이점도 갖고 있다.

다른 연구들은 소프트웨어 공학분야에서 활발히 연구되는 기법인 오류 위치 추정 기법(Fault localization)을 적용하여 시스템상의 오류 유발 인자를 분석하였다. G.

Ross의 [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. A. Arrieta와 [10]의 연구는 소프트웨어 생산 라인(Software product line, 이하 SPL)의 특징(Feature)에 초점을 두어 SBFL 기법을 적용하였다. 기존 SBFL에서 프로그램의 코드 수준으로 위치 추정을 진행한 것과 비슷하게, 해당 논문은 SPL의 특징(Feature)을 단위로 하여 위치 추정을 진행하였다. 이때, 기존 SBFL을 SPL의 특징한 특징들을 가지는 제품(Product)과 그 제품들에 대한 테스트 결과를 이용할 수 있도록 확장하였으며, SBFL의 결과를 이용하여 SPL에서의 디버깅 범위를 추가적으로 제한할 수 있는 Fault isolation technique 또한 제시하였다. 하지만 예시로 사용되어진 시스템이 SoS 수준의 시스템이라고 할 수 없으며, 통계적 검증이 아닌 테스트 결과에 이용하였다는 점에서 본 연구와는 다른 타겟 도메인을 가진다. 마지막으로, Y. J. Shin의 [6]의 연구는 SoS가 특정 목표를 수행하지 못 했을 때, 그 원인을 찾기 위해 SBFL을 적용하는 기법을 제안했다. 위 연구는 특정 시나리오에서 참여한 SoS의 구성 시스템(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. 해당 연구에서는 SMC의 결과를 기댓값의 형태로 해석하여 SBFL에 적용하였으며, 본 연구에서는 해당 해석 기법을 기법 1)로 채택하여 사용하였다.

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 기법으로, 이때 프로그램 스펙트럼이란 테스트 케이스를 실행할 때 실제 프로그램에서 실행된 프로그램 유닛(코드, 분기문, 등)의 집합을 나타낸다.

위 그림은 프로그램 코드 수준의 SBFL 기법의 예시

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
Fig. 1 An example of SBFL with a sample value-assigning program

를 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 1에서는 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 1에서 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

본 장에서는 SoS의 목표 달성 저해 요소 위치 추정을 위한 SBFL 적용을 위해, SMC의 확률적 결과를 SBFL에서 사용 가능한 Success, Failure값으로 활용할 수 있는 6가지의 기법을 소개한다.

아래 그림 2는 본 연구에서 실험할 6가지 SMC 결과의 해석/변환 기법을 요약하여 보여준다. 기법 ①은 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 ①을 제외한 기법 ②~⑤는 SMC 결과들의 분포를 이용한 통계적 해석 기법들이다. 기법 ②~④는 각각 SMC 결과 집합의 1분위, 2분위, 3분위값을 기준으로 해당 분위값보다 작은 SMC 결과들을 실패라고 간주하는 기법이다. 예를 들어 기법 ②를 적용한다면, 1분위

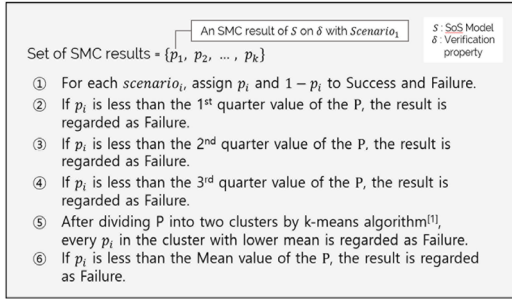


그림 2 SMC 검증 결과 변환 기법
Fig. 2 Utilization methods of SMC results

값을 27%라 가정할 때, 27%보다 작은 SMC 결과들을 실패로 간주하고 27% 이상의 SMC 결과들을 성공으로 간주한다. 기법 ②~④는 SMC 결과 분포에 대한 통계적인 성공/실패의 경계선을 각 분위값을 기반으로 설정하는 기법이다. 본 연구에서 1-sigma, 2-sigma, 3-sigma 법칙이 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.

기법 ⑤는 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. 마지막으로, 기법 ⑥은 기법 ②~④와 비슷하게 SMC 결과 분포의 Mean 값을 기준으로 그보다 작거나 같으면 Failure, 크면 Success로 간주하는 기법이다.

5. 실험 및 결과

5.1 실험 환경

앞서 설명한 6가지 SMC 결과의 해석 기법을 적용하기 위해 본 연구에서는 Y. Shin외[6] 연구에서 사용한 재난 대응 SoS와 그 시나리오를 사용하였다. 실험에 사용된 재난 대응 SoS는 두 종류의 구성 시스템(Constituent System, 이하 CS)인 “구조 로봇”과 “탐색 드론”을 가진다. 구조 로봇은 재난이 발생한 지역을 순회하며 환자들을 발견하고, 일정 확률로 환자들을 구출한다. 탐색 드론은 구조 로봇의 성능을 높여주기 위하여 더 빠른 속도로 재난 지역을 탐사하고, 환자를 인지하는 경우 연결된 구조 로봇에게 환자의 위치를 전달한다.

해당 SoS에는 총 10대의 구조 로봇, 10대의 탐색 드

재난 대응 SoS	재난 대응 SoS 시나리오					의심도*	Rank
	1	2	3	4	5		
CS {							
로봇1 - 0.88	●	●			●	0.53	4
로봇2 - 0.92	●		●			0.44	6
...							
드론1 - 0.45 // fault		●	●	●		0.83	1
드론2 - 0.90	●			●	●	0.51	5
...							
}							
Communication {							
로봇1-드론1		●				0.76	2
로봇1-드론2	●					0.38	7
로봇2-드론1			●			0.69	3
로봇2-드론2					●	0.13	8
...							
}							
목표 달성 확률 (구조율 65%)	0.8	0.47	0.61	0.55	0.94		
검증 결과 활용 (기법 1)	S=0.8 F=0.2	S=0.47 F=0.53	S=0.61 F=0.39	S=0.55 F=0.45	S=0.94 F=0.06	*Tarantula	

그림 3 재난 대응 SoS의 오류 위치 추정 기법 적용 예시
Fig. 3 An example of applying fault localization technique on emergency SoS

론과 100개의 커뮤니케이션 채널이 구현되어있으며, 한 번의 시나리오를 실행할 때마다 3개의 로봇, 3개의 드론, 그리고 1대1 대응인 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 이종의 시스템이라는 특징을 보이기 위해 같은 로봇, 드론에서도 구조율과 환자 인식률을 85%~100% 범위에서 랜덤하게 할당한다.

위의 재난 대응 SoS와 본 연구에서 제시한 SMC 검증 결과 활용 기법을 적용한 예시는 그림 3과 같다. 아래 시나리오는 드론1의 인식률에 오류가 삽입된 SoS이며, 하나의 시나리오마다 SoS에서 랜덤하게 3개의 로봇, 3개의 드론과 각각의 1대1 커뮤니케이션 채널을 선택한다. 본 예시에서는 원활한 설명을 위해 “로봇1”과 “로봇2”, “드론1”과 “드론2”, 그리고 그들의 커뮤니케이션만 표에 표시하였다. 검증 속성으로는 “삽입된 3대의 로봇과 드론들이 65명 이상의 환자를 구조할 수 있는지”를 사용하였다. 이때 검증 결과의 활용을 위해 기법 ①에서 제시한대로 기댓값에 기반하여 Success, Failure값을 각각 할당하여 계산하였으며, 위치 추정 기법의 의심도를 계산하기 위해 Tarantula 알고리즘을 활용하였다. 위 예시의 결과로 오류가 삽입된 “드론1”이 가장 의심스러운 요소로 판단되었다.

본 연구에서는 앞서 설명한 재난 대응 SoS에 로봇, 드론, 커뮤니케이션의 세 가지 오류를 삽입한 경우에 대하여 각각 200번의 랜덤 시나리오에 대한 SMC를 진행하였고, 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 해

당 SMC 데이터들에 대해 앞서 설명한 6가지 기법으로 변환한 값들을 이용하여 각각 SBFL 알고리즘을 적용하고, 그 결과를 비교하였다. 이때 실험에서는 세 가지 SBFL 알고리즘(Tarantula[13], Ochiai2[14], DStar[15])들을 이용하여 SBFL 알고리즘에 따른 각 변환 기법들의 성능 변화를 분석하였다. 재난 대응 SoS에 대한 통계적 검증은 SIMVA- SoS[9] 도구를 기반으로 진행하였으며, 검증 속성은 위의 예시와 같이 “100명의 환자 중 65명 이상의 환자를 구할 수 있는 지”를 이용하였다.

5.2 실험 결과 및 분석

아래 그림들은 각 SBFL 알고리즘에 대한 드론 오류, 로봇 오류, 커뮤니케이션 오류 삽입 경우들의 실험 결과를 나타낸다. 그림 4.1부터 그림 4.3은 Tarantula 알고리즘을 이용한 결과를, 그림 5.1부터 그림 5.3은 Ochiai2 알고리즘, 그림 6.1부터 그림 6.3은 DStar 알고리즘을 이용했을 때의 실험 결과를 나타낸다.

본 실험에서는 위치 추정 결과의 효용성을 판단하기 위해 EXAM-Score[16]를 이용하였다. EXAM-Score는 삽입한 모든 오류를 찾고난 후 확인하지 않아도 되는 code의 수를 수치화하여 위치 추정 알고리즘의 성능을 비교하는데 사용되어왔다. 본 연구에서는 이 지표를 SoS 수준에 맞게 활용하였다. 각 시나리오마다 삽입된 2개의 오류들을 1위부터 검색하여 모두 찾은 뒤, 검사하지 않아도 되는 CS들과 커뮤니케이션 채널의 수를 이용하여 EXAM-Score를 계산하였다. 각 기법들에 대한 성능을 분석해보자면, Tarantula 알고리즘을 적용했을 때(그림 4.1부터 그림 4.3) 드론 오류, 로봇 오류, 커뮤니케이션 오류 삽입의 모든 경우에서 기법 ①이 다른 기법들보다 좋은 성능을 보임을 확인할 수 있다. 기법 ①이 Ochiai2와 DStar 알고리즘을 이용한 로봇 오류, 드론 오류 삽입 시나리오에서도 준수한 성능을 보이는 것 또한 확인할 수 있다(그림 5.1부터 그림 5.2, 그림 6.1부터 그림 6.2).

하지만, 통계적인 경계선을 정하여 Success와 Failure를 구분하는 기법 ②~④의 경우는 특정 시나리오와 알고리즘을 적용했을 때 가장 뛰어난 성능을 보여주지만, 그 경향성이 일관되지 않다. 예를 들어, Tarantula 알고리즘을 적용한 경우(그림 4.1부터 그림 4.3)는 기법 ②, ③, ④의 순서대로 성능을 보이지만, Ochiai2와 DStar 알고리즘을 드론 오류, 로봇 오류 시나리오에 적용했을 때(그림 5.1부터 그림 5.2, 그림 6.1부터 그림 6.2)는 기법 ③이, 커뮤니케이션 오류 삽입 시나리오(그림 5.3, 그림 6.3)에서는 기법 ②가 가장 좋은 성능을 보인다. 기법 ⑤의 경우는 DStar 알고리즘을 드론 오류 삽입 시나리오(그림 6.1)에 적용했을 때만 가장 좋은 성능을 보이며, 기법 ⑥의 경우는 전체적으로 기법 ③과 비슷한 성능을 보인다.

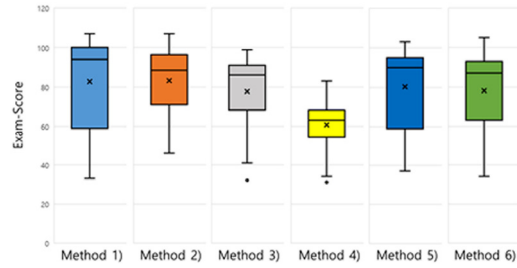


그림 4.1 드론 오류 시나리오(Tarantula)의 EXAM-Score 분포

Fig. 4.1 EXAM-Score Distribution on the drone fault scenario with Tarantula

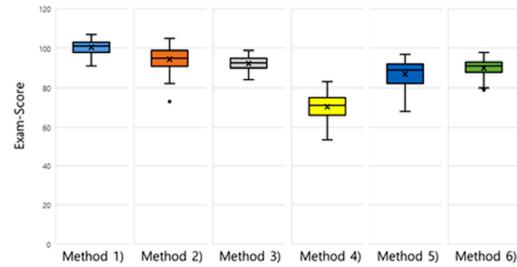


그림 4.2 로봇 오류 시나리오(Tarantula)의 EXAM-Score 분포

Fig. 4.2 EXAM-Score Distribution on the robot fault scenario with Tarantula

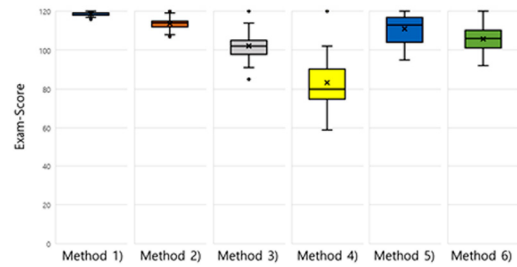


그림 4.3 커뮤니케이션 오류 시나리오(Tarantula)의 EXAM-Score 분포

Fig. 4.3 EXAM-Score Distribution on the communication fault scenario with Tarantula

따라서, 제안한 6가지의 기법들 중 기댓값에 기반한 기법 ①이 본 실험에 사용된 재난 대응 SoS 및 SBFL 알고리즘들에 대해 일반적으로 준수한 성능을 보임을 확인할 수 있었다. 그리고 커뮤니케이션 오류 삽입의 경우에 1분위 값에 기반하여 Success와 Failure 값을 나눈 기법 ②와 K-means 알고리즘을 이용한 기법 ⑤ 또한 좋은 성능을 보임을 확인하였다.

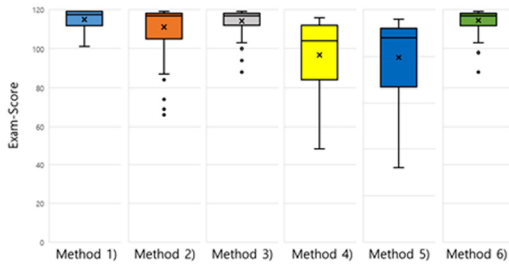


그림 5.1 드론 오류 시나리오(Ochiai2)의 EXAM-Score 분포

Fig. 5.1 EXAM-Score Distribution on the drone fault scenario with Ochiai2

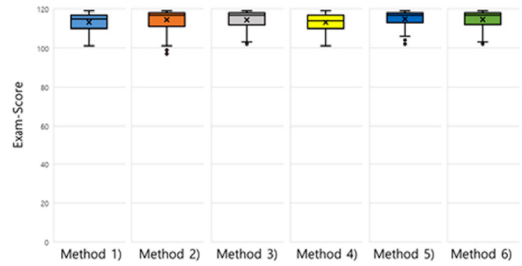


그림 6.1 드론 오류 시나리오(DStar)의 EXAM-Score 분포

Fig. 6.1 EXAM-Score Distribution on the drone fault scenario

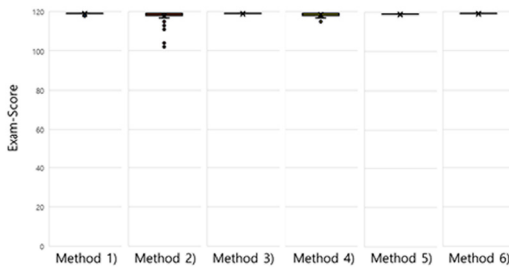


그림 5.2 로봇 오류 시나리오(Ochiai2)의 EXAM-Score 분포

Fig. 5.2 EXAM-Score Distribution on the robot fault scenario with Ochiai2

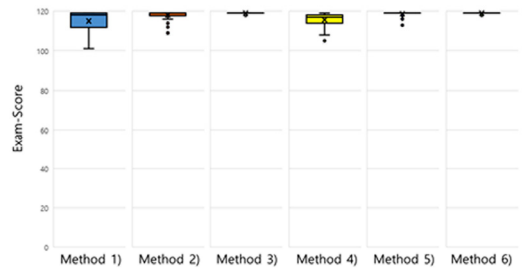


그림 6.2 로봇 오류 시나리오(DStar)의 EXAM-Score 분포

Fig. 6.2 EXAM-Score Distribution on the robot fault scenario

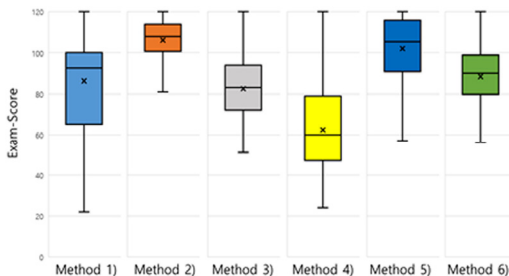


그림 5.3 커뮤니케이션 오류 시나리오(Ochiai2)의 EXAM-Score 분포

Fig. 5.3 EXAM-Score Distribution on the communication fault scenario with Ochiai2

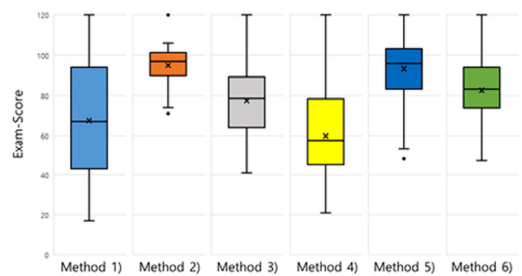


그림 6.3 커뮤니케이션 오류 시나리오(DStar)의 EXAM-Score 분포

Fig. 6.3 EXAM-Score Distribution on the communication fault scenario

6. 결론

대규모 복잡 시스템인 SoS는 통계적 모델 검증 기법을 이용하여 특정 목표의 달성도를 확인할 수 있다. 목표 달성에 실패하여 SoS에 존재하는 성능 저해 요소의 위치를 추정해야 하는 경우 기존의 위치 추정기술을 적용할 수 있다. 본 연구에서는 오류 위치 추정 기술인 SBFL을 SoS 모델에 확장 적용하는 과정에서 필요한

SMC의 결과 해석 방안 6가지를 소개하였다. 그리고 세 가지 SBFL 알고리즘(Tarantula, Ochiai2, DStar)과 제안 대응 SoS를 이용하여 위의 6가지 기법들 중 SMC의 결과를 기댓값에 기반하여 해석한 기법 ①이 평균적으로 좋은 성능을 보임을 확인할 수 있었다.

또한, 커뮤니케이션 오류 삽입과 같은 특수한 경우에 기법 ②와 기법 ⑤ 높은 성능을 보임을 확인할 수 있었다. 실험 결과를 통해 SoS의 통계적 검증 결과를 오류

위치 추정 기법에 적용할 수 있는 가능성을 확인하고 여러 기법들의 성능을 비교할 수 있었다. 본 실험에서 진행한 성능 비교가 단일 SoS 및 세 가지 SBFL 알고리즘들을 이용하여 진행되었기 때문에 일반화하기 어렵지만, 본 연구의 내용을 참조하여 오류의 위치에 따라 적합한 변환 기법을 사용할 수 있다. 추후 연구로는 재난 대응 SoS 뿐만 아니라 사용자에 최적화된 온도 및 습도를 전력량을 최소화하여 유지할 수 있는 스마트 홈 SoS와 자율 주행 및 군집 운행(Platooning) 시뮬레이터를 이용한 스마트 교통 SoS에 오류 위치 추정 기법을 적용하고, 이에 따른 변환 기법 및 오류 위치 추정 모델 및 알고리즘을 제안하는 연구를 진행할 계획이다.

References

- [1] J. Boardman, and B. Sauser, "System of Systems -the Meaning of Of," *IEEE/SMC International Conference on System of Systems Engineering (SoSE)*, 2006.
- [2] D. Seo, D. Shin, Y.-M. Baek, J. Song, W. Yun, J. Kim, E. Jee, and D.-H. Bae, "Modeling and verification for different types of system of systems using prism," *Software Engineering for Systems-of-Systems (SESoS), 2016 IEEE/ACM 4th International Workshop on. IEEE*, pp. 12-18, 2016.
- [3] L. Axel, J. Quilbeuf, and F. Oquendo, "Verifying Systems-of-Systems with Statistical Model Checking," *ERCIM News* 103 (2015).
- [4] K. Kenan, C. Jegourel, A. Lukina, E. Bartocci, A. Legay, S. A. Smolka, and R. Grosu, "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, R. Gao, Y. Li, R. Abreu, and F. Wotawa, "A survey on software fault localization," *IEEE Transactions on Software Engineering*, Vol. 42, No. 8, pp. 707-740, 2016.
- [6] Y. -J. Shin, S. Hyun, Y.-M. Baek, and D.-H. Bae, "Spectrum-Based faultlocalization on a collaboration graph of a System-of-Systems," *2019 14th Annual Conference System of Systems Engineering (SoSE) (SoSE2019)*, Anchorage, USA, May 2019.
- [7] J.A. Hartigan, and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)* Vol. 28, No. 1, pp. 100-108, 1979.
- [8] G. Ross, "Isolating the causes of emergent failures in computer software," *Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach*, pp. 263-284, 2018.
- [9] SE Lab. (2016, June 31). SIMVA-SoS Available: <https://github.com/SESoS/SIMVA-SoS>
- [10] A. Arrieta, S. Segura, U. Markiegi, G. Sagardui, and L. Etxeberria, "Spectrum-based fault localization in software product lines," *Information and Software Technology*, 100 (2018): 18-31.
- [11] H. Jarrah, N. I. Sarkar, and J. Gutierrez, "Comparison-based system-level fault diagnosis protocols for mobile ad-hoc networks: A survey," *Journal of Network and Computer Applications* 60 (2016): 68-81.
- [12] L. Qin, X. He, and D. H. Zhou, "A survey of fault diagnosis for swarm systems," *Systems Science & Control Engineering: An Open Access Journal* 2.1 (2014): 13-23.
- [13] J. A. Jones, M. J. Harrold, and J. T. Stasko, "Visualization for fault localization," *Proc. of ICSE 2001 Workshop on Software Visualization*, 2001.
- [14] L. Naish, H. J. Lee, and K. Ramamohanarao, "A model for spectra-based software diagnosis," *ACM Transactions on software engineering and methodology (TOSEM)* 20.3 (2011): 11.
- [15] W. E. Wong, V. Debroy, R. Gao, and Y. Li, "The DStar method for effective software fault localization," *IEEE Transactions on Reliability* 63.1 (2013): 290-308.
- [16] S. Pearson, J. Campos, R. Just, G. Fraser, R. Abreu, and M. D. Ernst "Evaluating and improving fault localization," *Proc. of the 39th International Conference on Software Engineering*. IEEE Press, 2017.



현 상 원

2018년 한양대학교 컴퓨터공학부 소프트웨어전공 학사. 2019년~현재 한국과학기술원(KAIST) 전산학부 석박통합과정. 관심분야는 시스템 오브 시스템즈의 검증 및 검증 결과를 활용한 결함 위치 추정



신 용 준

2017년 한동대학교 전산전자공학부 학사 2017년~현재 한국과학기술원(KAIST). 전산학부 석박통합과정. 관심분야는 시스템 오브 시스템즈 공학, 소프트웨어 모델링, 자가적응 시스템



배 두 환

1980년 서울대학교 조선공학 학사. 1987년 Univ. Of Wisconsin-Milwaukee 전산학과 석사. 1992년 Univ. Of Florida 전산학과 박사. 1995년~현재 한국과학기술원(KAIST) 전산학부 교수. 관심분야는 소프트웨어 프로세스, 객체지향 프로그래밍, 컴포넌트 기반 프로그래밍, 임베디드 소프트웨어 설계, 관련 지향 프로그래밍, 시스템 오브 시스템즈 소프트웨어 공학