

# 시스템 오브 시스템즈의 특성을 고려한 발현 위치 기반 불확실성 요소 분류

## (Manifestation Location-based Classification of Uncertainty Factors Considering Characteristics of System-of-Systems)

신 승 철 \*    현 상 원 \*\*    신 용 준 \*    송 지 영 \*\*    배 두 환 \*\*\*  
(Seungchul Shin) (Sangwon Hyun) (Yong-jun Shin) (Jiyoung Song) (Doo-Hwan Bae)

**요약** 시스템 오브 시스템즈(System-of-Systems, SoS)는 다양한 구성시스템들로 구성된 대규모의 복잡한 시스템을 말한다. SoS의 목표 달성 여부를 확인하기 위해 전체 시스템에 대한 모델링과 검증이 필요하다. 한편, 구성시스템들은 다양한 환경적 요소에 의한 불확실성에 노출되어, SoS의 목표 달성 여부를 정확히 검증하기 위해서는 불확실성을 고려해야 한다. 그러나 SoS의 특성을 고려한 불확실성 요소를 체계적으로 분류한 연구가 부족하다. 따라서 본 논문에서는 SoS의 특성을 고려한 불확실성 요소 분류 체계를 제안한다. 제안하는 분류 체계는 불확실성을 발현 위치를 기반으로 문맥, 구조, 입력 변수, 매니저, 런타임의 다섯 가지로 분류한다. 제안하는 분류 체계는 두 종류의 SoS에 사례 적용된다. 제안하는 분류 체계를 통해 SoS를 검증할 때 발현할 수 있는 모든 불확실성을 충분히 고려할 수 있으며, SoS 대상 성능평가 및 검증 기술 연구의 신뢰를 높일 수 있을 것이다.

**키워드:** 시스템 오브 시스템즈, 구성시스템, 불확실성, 발현 위치 기반 분류 체계

**Abstract** System-of-Systems (SoS) is a large-scale complex system comprising various configuration systems. To check if an SoS satisfies its common goal, it is necessary to model and verify the entire system. Conversely, constituent systems are exposed to uncertainty because of the various environmental factors, so uncertainty must be considered to accurately verify if the SoS goal is achieved. However, there is a lack of research that systematically classifies uncertainty factors considering the characteristics of the SoS. Thus, in this paper, we propose an uncertainty factor classification system considering the characteristics of the SoS. The classification system proposed in this paper classifies uncertainty into five categories based on the manifestation location: the context, the structure, the input variables, the manager, and the runtime. The proposed classification system is applied to two types of the SoS. When the SoS is verified through the proposed classification system, it is possible to fully consider all the uncertainties that may arise and it is expected that this will increase the confidence of the SoS target performance evaluation and the verification technology research.

**Keywords:** system-of-systems, constituent systems, uncertainty, manifestation location based classification

- \* 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 한국소프트웨어종합학술대회에서 '시스템 오브 시스템즈에서의 불확실성 요소 분석: 군집운영 시나리오에서의 사례 연구' 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 한국과학기술원 전산학부 학생(KAIST)

seshin@se.kaist.ac.kr  
(Corresponding author)  
yjshin@se.kaist.ac.kr

\*\* 비 회원 : 한국과학기술원 전산학부 학생

swhyun@se.kaist.ac.kr  
jysong@se.kaist.ac.kr

\*\*\* 종신회원 : 한국과학기술원 전산학부 교수  
bae@se.kaist.ac.kr

논문접수 : 2020년 4월 29일  
(Received 29 April 2020)  
논문수정 : 2020년 7월 3일  
(Revised 3 July 2020)  
심사완료 : 2020년 7월 31일  
(Accepted 31 July 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.  
정보과학회 컴퓨팅의 실제 논문지 제26권 제10호(2020, 10)

## 1. 서론

시스템 오브 시스템즈(System-of-Systems, 이하 SoS)는 개별 구성시스템(Constituent Systems, 이하 CS)들의 상호작용을 통하여, 단일 CS들로는 달성할 수 없는 상위 단계의 목표를 달성하고자 하는 대규모 복잡 시스템이다[1]. SoS는 각 CS에 대한 권한, 운영에 대한 권한, SoS 단계의 목표 달성 여부 등에 따라 지시적(Directed), 상호인정적(Acknowledged), 협력적(Collaborative), 가상(Virtual)으로 분류된다[2]. 예시로 군집운행 SoS에서는 군집을 구성하는 차들은 독립적이며 자체적인 판단으로 운영을 하지만 이러한 차들이 하나의 군집을 이루면서 도로 처리량을 높인다는 SoS 단계의 목표를 달성한다. 군집운행 SoS와 같이 상위 단계의 목표를 강제하지 않지만 각 CS 간의 자발적인 참여를 통하여 달성하고 SoS 단계의 운영 지침을 따르지 않는 SoS를 협력적 SoS라 한다.

SoS에서는 다양한 불확실성 요소가 발현될 수 있다. 예를 들어, 협력적 SoS에서는 내부 CS에 관한 규정과 능력이 투명하게 SoS의 매니저에게 전달되지 않는다. 따라서 SoS의 매니저는 SoS를 구성하는 내부 CS들에 대한 정보가 부족한 상태이다. 이러한 정보의 부족함으로 인해 CS 자체의 운영이 SoS의 목표 달성을 저해하는 불확실성이 발생한다[3].

시스템의 모델링부터 검증까지 불확실성 요소들이 충분히 고려되지 않는다면 SoS의 실패를 유발할 수 있다. 따라서 기존 SoS 연구에서는 불확실성 요소를 고려하여 시스템의 성능을 평가하였다[2,4-7]. 주로 센서 노이즈와 같이 변수 자체에 존재하는 불확실성을 다루었으며[2,4,5] SoS 내부 구조의 변화로 인한 불확실성[6]이나 내부 CS의 독립성으로 인해 CS의 운영을 확신할 수 없어서 생기는 불확실성[7]에 대한 연구가 진행되었다.

선행 연구들에는 SoS에서의 불확실성을 다루는 여러 방법이 제안되었다. 선행 연구 모두 SoS의 특정 분류의 불확실성에 대한 유용한 접근을 진행한 연구임은 분명하지만, 특정 불확실성에 대한 고려만 진행되었다. 이러한 단편적인 불확실성 요소의 고려는 시스템의 성능평가 결과에 대한 신뢰(Confidence)를 충분히 높이지 못한다. 다양한 불확실성을 복합적으로 고려하기 위해 불확실성 요소에 대한 분류는 필요하나 SoS의 불확실성 요소 자체에 대한 체계적인 분류를 진행한 연구를 찾지 못하였다.

본 연구에서는 SoS에서의 불확실성 요소들을 발현 위치(Manifestation Location)를 기준으로 분류하는 분류 체계를 제안한다. 분류 기법의 적용성을 알아보기 위해 두 종류의 SoS 도메인에 사례 적용한다. 본 연구에

서 제공하는 분류 체계는 SoS 검증을 수행할 때 해당 SoS에서 발현할 수 있는 모든 불확실성을 충분히 고려할 수 있게 한다. 또한, SoS 대상 성능평가 및 검증 기술 연구의 신뢰도(Confidence)를 측정하는 간접적인 지표로써 사용될 수 있다.

본 연구의 구성은 다음과 같다. 2장에서 SoS와 유사하며 기존에 불확실성에 대한 고려가 많이 되어왔던 자가 적응 시스템에서의 불확실성과 기존 SoS 연구에서 고려되던 불확실성에 대해 설명한다. 3장에서는 SoS에서의 불확실성 요소의 분류 체계를 소개한다. 4장에서는 사례 적용에 사용하는 두 종류의 SoS 도메인에 대해 설명하며 시나리오를 이용하여 발생할 수 있는 불확실성 요소의 사례를 설명한다. 마지막으로 5장에서는 결론 및 향후 연구를 기술한다.

## 2. 관련 연구

본 관련 연구에서는 자가 적응형 시스템(Self-Adaptive System, 이하 SAS)과 SoS, 불확실성을 다루고 있는 두 개의 서로 다른, 그러나 매우 밀접한 관계를 가지고 있는 시스템의 불확실성에 대해 논의한다. 일반적으로 SAS는 시스템 전체를 top-down 방식으로 개발하는 반면에 SoS는 bottom-up 형식으로 시스템을 구성한다는 측면에서 불확실성에 대한 연구가 진행되어 왔다.

### 2.1 자가 적응 시스템에서의 불확실성

대부분의 SoS는 불확실한 상황에서도 상위의 목표를 달성하는 것을 요구받는다라는 점에서 SAS와 유사하다. SAS는 환경이나 요구사항의 변화에서도 목표를 달성하는 시스템을 말한다[8]. 확장성(Scalability)을 고려하여 복잡성이 늘어나게 돼도 다양한 변화에 대해 적응하여 SAS를 유지하는 것이 중요하다. 불확실성은 SAS를 구현하는 본질적인 특성이기 때문에, 불확실성에 관한 연구는 SAS 분야의 연구자들에 의해 활발히 진행되어 왔다[3,9,10]. SAS의 성능에 직접적인 영향을 끼치는 불확실성에 대한 정의 및 분류를 중심으로 연구가 진행되었으며[3,9] 이후의 불확실성을 고려한 SAS의 연구들에 큰 발판이 되어주었다.

SAS에서는 시스템을 모델링 할 때 관찰할 수 있는 불확실성 요소를 분류하는 기준은 다양하다. 불확실성이 발현되는 위치(Location), 불확실성 자체의 수준(Level), 불확실성의 습성(Nature) 등에 따라 분류를 진행하거나 [3] 불확실성이 일어나는 단계(Level)에 따라라도 분류를 진행한다[9].

기존에 존재하던 시스템들의 협력으로 상위 목표를 달성한다는 SoS의 주된 특성은 적응을 목적으로 내부 시스템들을 적절하게 구성하는 SAS와 구성된 목적이나 CS와의 관계에서 주요한 차이를 보인다. SoS는 SAS와

는 다른 특성들을 추가로 가지기 때문에, SAS에서 고려했던 불확실성 요소들만으로는 충분하지 않다. 따라서 본 연구에서는 SAS에서의 불확실성 요소들을 기반으로, SoS의 특성을 고려한 불확실성 요소들을 추가로 조사하고, 이를 포괄하는 발현 위치 기반 분류 체계를 제안한다.

## 2.2 시스템 오브 시스템즈에서의 불확실성

SoS의 상위 단계의 목표를 다양한 환경 속에서도 안정적으로 달성하기 위해서는 다양한 불확실성에 대한 고려가 필요하다. SoS는 CS들이 블랙박스이고 CS 간의 협력으로 인해 상위 단계의 목표를 달성한다는 특성을 갖는다. CS에 대한 정보가 부족하며 CS 자체가 서로 간의 협력을 사전에 고려하지 않은 채로 상위 시스템을 구성하는 SoS의 경우에는 더욱 많은 불확실성에 대한 고려 역시 필요하다.

불확실성을 고려한 SoS에 대한 연구를 진행한 논문을 다양하게 찾아볼 수 있다[2,4-7]. 기존 연구들은 주로 변수 자체가 존재할 수 있는 불확실성에 집중하고 있다. 변수 자체가 하나의 확률을 표현하여 불확실한 사건을 만들어내거나[2], 부족한 정보로 인해 실제 실행환경에서 받아들이는 입력 변수의 값에 대한 확신이 없어 불확실한 경우[4,5]를 고려했다. 변수 이외에서 찾아볼 수 있는 불확실성을 고려한 연구에서는 내부 구조의 변화로 인한 불확실성[6]과 CS들의 관리 독립성(Managerial Independence)으로 인한 불확실성[7]을 고려했다. 또한, 고려하고자 했던 불확실성에 대한 정의 또는 언급 자체가 불확실한 경우도 다수 존재한다.

선행 연구들은 특정 불확실성 요소를 고려하여 SoS의 목표 달성도 및 성능평가를 진행하였지만, 단편적으로 불확실성을 고려하게 된다면 다른 요소에 의해 발생하는 불확실성에 대한 신뢰성을 보장할 수 없다. 그러나 다양한 관점과 분류 체계가 제안된 SAS에서의 불확실성 연구들에 비해 SoS에서의 복합적인 관점으로 바라본 불확실성에 대한 연구가 부족하다. 따라서 본 연구에서는 SoS에서 발생할 수 있는 불확실성 요소들을 포괄적으로 조사하여 분류하고자 한다.

## 3. 시스템 오브 시스템즈에서의 발현 위치 기반 불확실성 요소 분류

본 장에서는 기존에 다루어졌던 SAS에서의 불확실성에 대한 분류 체계를 기반으로 고안해낸 SoS에서의 불확실성 요소 분류를 설명한다. 특히, 불확실성이 관찰되는 발현 위치는 불확실성을 분류할 수 있는 공정하고 절대적인 지표가 되기에 발현 위치 기반으로 분류한 분류 체계를 설명한다.

SoS에서의 불확실성 요소는 다음과 같이 분류된다. 첫 번째로, 기존의 SAS에서 시스템의 불확실성 요소를

위치에 따라 분류했던 기준[3]인 문맥, 구조, 입력 변수에서 발현되는 불확실성을 SoS에 맞게 재해석한다. 시스템 자체를 구성 요소로 하는 SoS는 선행 연구에서 고려된 기존의 시스템 내의 구성 요소로 인한 불확실성을 포함하며 추가로 CS 단위로도 정의된다. 두 번째로, 기존의 SAS에서는 고려할 수 없었던 SoS의 매니저로 인한 불확실성 및 런타임에서 발현되는 불확실성을 정의한다. SoS에서의 문맥, 구조, 입력 변수, 매니저와 런타임에서 발현되는 불확실성에 대한 정의는 다음과 같다.

**문맥에서 발현되는 불확실성**이란 SoS 자체가 어떠한 CS들로 구성되어 있는가에서 발견할 수 있는 불확실성이다. SoS에 속하지 못한 CS나 SoS에 잘못 속하게 된 CS들이 SoS의 목표 달성에 영향을 주는 경우 발견할 수 있다. 즉, SoS에 속하거나 속하지 못한 CS들을 통해 구분할 수 있다. 그림 1에서 SoS1은 CS1, CS2, CS3만 포함하여 구성되고 CS4를 고려하지 않고 있다. 그러나 CS4가 SoS의 목표 달성에 대한 기여도가 큰 경우 CS4를 고려하지 못한 정보의 부족함으로 인해 불확실성이 발생한다. 마찬가지로 그림 1에서 SoS2는 CS2, CS3, CS4만 포함하고 있으므로 CS1으로 인한 영향을 고려하지 못하게 된다. 두 가지 예시의 SoS는 실제 SoS의 구성 요소들을 모두 고려하지 않았기에 문맥에서 발현되는 불확실성을 가지게 된다.

**구조에서 발현되는 불확실성**은 SoS를 구성하고 있는 CS들이 이루는 구조에서 발견할 수 있는 불확실성이다. SoS는 가지고 있는 정보를 어떤 식으로 구성했는지에 따라 다양한 구조로 표현할 수 있다. 같은 CS들로 구성되어 있어도 다른 구조로 목표 달성이 가능하기에 구조의 차이로 구분할 수 있다. 그림 2에서 SoS는 CS1, CS2, CS3을 모두 포함하여 구성되어 있다. CS1과 CS2 사이의 상호작용은 CS3을 통하여 이루어지는데, 결합 허용성으로 인해 CS3의 다른 표현인 CS3'로도 SoS를 구성할 수 있으며 CS3과 CS3' 중 현실을 더욱 잘 반영한 CS가 존재한다. 여기서 불확실성은 현실을 더욱 잘 반영하여 표현한 CS가 존재하기 때문에 생겨난다.

앞선 두 불확실성의 분류가 각각 SoS가 어떤 CS들을 포함하였는지와 그 CS들이 어떤 구조를 이루는지에 대한 불확실성이었다면, **입력 변수에서 발현되는 불확실성**은 CS 내부에서 발견할 수 있는 불확실성이다. 각 CS의 모델에게 주어지는 입력 값과 이를 조정하는 방법을 통해 구분할 수 있다. 주로 정보의 부족뿐만 아니라 정보 자체의 불확실함이 원인이 된다. 그림 3에서 볼 수 있듯이 CS3에 입력되는 변수에서 발현되는 불확실성이며 대표적인 예시로 센서 소음(Sensor Noise)이나 확률을 입력 변수로 받는 경우를 들 수 있다.

**매니저에서 발현되는 불확실성**은 SoS의 매니저와 각

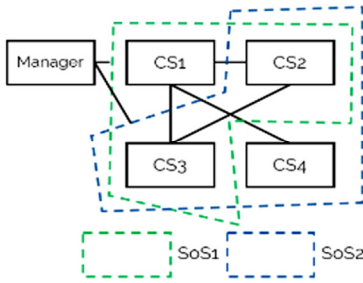


그림 1 문맥에서 발현되는 불확실성  
Fig. 1 Uncertainty manifested in the context

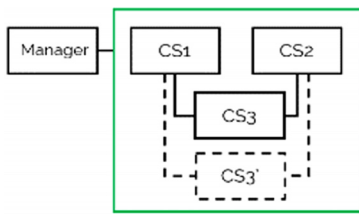


그림 2 구조에서 발현되는 불확실성  
Fig. 2 Uncertainty manifested in the structure

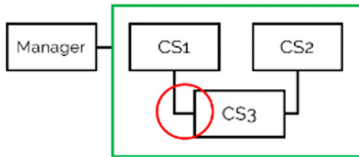


그림 3 입력 변수에서 발현되는 불확실성  
Fig. 3 Uncertainty manifested in the input parameter

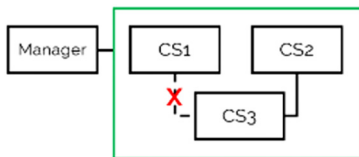


그림 4 매니저에서 발현되는 불확실성  
Fig. 4 Uncertainty manifested in the manager

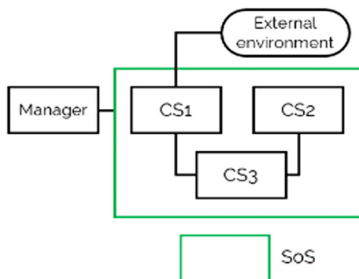


그림 5 런타임에서 발현되는 불확실성  
Fig. 5 Uncertainty manifested in the runtime

CS의 매니저 전체에서 발현되는 불확실성을 말한다. 즉, SoS와 CS의 매니저가 따로 있기에 각각의 운영 방침이 다를 때 일어나는 불확실성을 구분한다. 주로 매니저 간의 소통/정보의 부족과 동적 재구성으로 인해 일어난다. SoS의 특징으로는 각 CS는 블랙박스로 취급되며 서로 독립성을 갖고 있기에 SoS 매니저가 CS에 대한 정보 부족(Lack of Knowledge)을 갖는다. CS가 독립성을 갖기에 CS 단계의 목표에 따라 동적으로 다른 CS와의 연결을 끊을 수 있으며(그림 4), CS 단계의 목표가 SoS 단계의 목표와 충돌하는 경우가 발생할 수 있다. 각 CS의 관리 독립성과 정보 부족은 불확실성의 요소로서 관찰될 수 있다.

마지막으로, 런타임에서 발현되는 불확실성이 존재한다. 런타임에서 발현되는 불확실성은 런타임 환경에서만 관찰되는 불확실성을 말한다. 모델링 단계에서 예측하는 게 어려우며 실제 실행환경에서 주어지는 한정된 자원과 예측할 수 없는 환경과의 상호작용에서 관찰되는 불확실성이다. 그림 5와 같이 런타임 시의 외부 환경으로 인한 불확실성을 이야기하며 가장 대표적으로 행동을 예측하기 힘든 사람을 들 수 있다.

#### 4. 시스템 오브 시스템즈 시나리오에의 사례 적용

본 연구에서 제안하는 불확실성 요소의 분류 체계의 타당성을 확인하기 위해 사례 연구 대상 SoS로 군집운행 SoS와 스마트홈 SoS를 선정했다. 군집운행 SoS는 고속도로에서 운행되는 차들 사이의 간격을 최적화하여 고속도로 전체의 차량 처리량(Throughput)을 높이는 상위 단계의 목표를 달성한다. 이를 위해 각 차량은 적응형 순항 제어(Adaptive Cruise Control) 시스템을 이용해 차량의 속도를 제어하며 군집운행 시스템에서 차량의 행동방침과 속도에 대한 지침을 내려준다. 각 차량은 개별 단계의 목적지와 목표를 갖고 있지만 서로 간의 협력을 통해 상위 단계의 목표를 달성하며 협력적 SoS를 구성한다.

스마트홈 SoS를 구성하고 있는 내부 시스템들은 개별적으로 동작하는 독립적인 시스템들이지만, 서로가 제공하는 기능과 서비스의 상호작용으로 인하여 생산성과 편의성을 최대화하고 비용을 최소화하는 상위 단계의 목표를 달성한다. 구성시스템 자체에 편의성을 목표로 하는 경우가 많기에 생산성에 더욱 집중하면 오락(Entertainment), 보안, 에너지 관리 그리고 건강 관리의 네 개의 기준에 구성시스템으로는 다루지 않던 목표인 상위 목표를 달성한다[11]. 그러나 때로는 스마트홈 SoS의 판단하에 내부 시스템의 운영에 영향을 줄 수 있기에 상호인정적 SoS를 구성한다.

**4.1 군집운행 시나리오에의 사례 적용**

본 장에서는 협력적(Collaborative) SoS인 군집운행 SoS 시나리오에 대한 설명과 그 경우 일어날 수 있는 매니저와 런타임에서의 불확실성에 대해 알아본다.

매니저에서 관찰되는 불확실성은 SoS 매니저와 CS 매니저 간의 정보 및 소통의 부족으로 인해 일어나는 경우가 대표적이다. 서로 다른 군집운행 시스템을 가진 차량들이 하나의 군집을 구성하는 경우에 쉽게 관찰할 수 있다. 그림 6에서 세 트럭은 서로 다른 군집운행 시스템을 가지므로 내부의 제한요소(Constraints)에 차이가 있다. 트럭 1이 군집의 리더(Leader)로서 군집의 기본 동작에 대한 지시를 내리고, 트럭 2와 트럭 3이 군집의 후행 차량(Follower)으로서 해당 지시에 맞는 기동(Maneuver)을 시행한다. 트럭 3이 CS 단계의 목표에 따라 일찍 분기(Split)를 하기 위해 군집의 리더인 트럭 1에 분기 요청을 보내는 상황에서, 트럭 3의 요청 전송 실패 시간제한(Request Failure Time Constraint)이 트럭 1의 요청 인지 시간제한(Request Acceptance Time Constraint)과 응답 전송 시간제한(Response Time Constraint)보다 짧은 경우 트럭 3의 CS 단계 목표는 실패한다.

런타임에서 관찰되는 불확실성은 실제 실행 단계에 있어서 주어지는 제한된 자원의 소모와 주변 환경과의 상호작용으로 인해 발생할 수 있다. 주로 예측할 수 없는 환경요소에 의해 발생하는 경우와 SoS와 환경요소가 한정된 자원을 공유하게 되는 경우를 예로 들 수 있다. 예측할 수 없는 환경요소에는 SoS로 인한 부가효과(Side Effect)가 환경에 미친 영향이 다시 SoS에 영향을 미치게 되는 경우와 예측할 수 없는 사람으로 인한 불확실성 요소를 포함한다. 그림 6에서 나타난 세 대의 트럭은 정방향의 2차선이라는 제한된 도로 자원 중 한 차선의 많은 부분을 소모하고 있다. 따라서 환경요소로 생각될 수 있는 군집운행을 하지 않는 외부 차량들은 차선이 1차선으로 줄어드는 병목현상을 겪게 되며 기존에 가능하던 차량의 추월 자체가 불가능해진다. 따라서 도로상의 상대적 차량의 수가 증가하는 현상이 일시적으로 발생하며 이는 고속도로에서의 차량 처리량을 늘리려는 SoS 단계의 목표 달성에 실패하는 사례이다.

**4.2 스마트홈 시나리오에의 사례 적용**

본 장에서는 상호인정적(Acknowledged) SoS인 스마트홈 SoS 시나리오에 대한 설명과 그 경우 일어날 수 있는 매니저와 런타임에서의 불확실성에 대해 알아본다. 본 장에서 사용하는 스마트홈 SoS 시나리오는 다음과 같다.

스마트홈 SoS의 설정된 목표는 에너지 효율과 거주민의 건강 관리로 가정한다. 스마트 온도 관리 시스템은 가장 쾌적한 온습도를 선정해 거주민이 최적의 상태로 있을 수 있게 한다. 선정된 온습도에 따라 에어컨과 난



그림 6 2차선에서 군집운행 중인 트럭 세 대  
Fig. 6 Three trucks platooning in the two lanes

방기는 상시 방의 온도를 체크하며 스스로의 판단하에 쾌적한 온도를 만들어낸다. 스마트 냉장고는 RFID를 식별하여 음식에 대한 정보를 읽어내고 관리한다. 스마트 홈 SoS는 거주민의 건강 관리를 위해 스마트 냉장고를 이용하여 식품의 상태를 체크하고 소비하는 음식으로 인한 영양 상태 및 열량에 대한 정보를 공유하도록 한다. 또한, 해당 음식을 거주민이 소비했다는 신뢰성을 높이기 위해 스마트 인덕션의 가동과 식기세척기의 가동, 전자레인지의 가동 등에 대한 정보를 토대로 거주민이 하루에 섭취한 열량 및 영양 상태를 기록하고 그에 따른 정보와 권장 식단을 매일 아침 제공한다.

매니저에서 관찰되는 불확실성은 스마트홈 SoS를 구성하는 여러 시스템이 스마트홈 SoS에 종속적인 제품이 아니며 서로 간의 정보가 부족한 상태이기에 발생한다. 스마트 온도 관리 시스템은 스마트홈 SoS에 쾌적한 온도만을 요구받은 상황에서 방 안의 온도를 높이기 위해 난방기를 가동한다. 난방기는 스스로의 에너지 효율을 위해 요구 온도에 근접한 경우 화력을 낮추어 천천히 온도를 높게 된다. 이러한 상황에서 스마트홈 SoS는 요구 온도의 달성이 잘되지 않자 문제가 있다고 판단, 에어컨의 난방을 요구하고 따라서 요구 온도를 달성한다. 각 구성시스템은 요구받은 온도를 달성하며 스스로의 에너지 효율을 관리하지만, 스마트홈 SoS의 입장에서 에너지 효율은 감소한다.

런타임에서 관찰되는 불확실성은 앞서 말했듯이 실행 단계에서 주어지는 제한된 자원의 소모와 주변 환경과의 상호작용에서 발생한다. 특히 스마트홈 같은 경우에는 CS가 아닌 외부의 환경요소 중 가장 예측할 수 없으며 모델링 할 수 없는 사람이라는 환경요소가 필연적으로 얽매어 있다. 따라서 사람의 예측할 수 없는 행동이 스마트홈 SoS에는 불확실성으로 다가오는 경우가 많다. 스마트홈 SoS가 거주민의 단백질 섭취량이 부족하여 고기를 구워 먹도록 안내하여 거주민이 냉장고에서 고기를 꺼내 스마트 인덕션에서 고기를 굽는다. 그러나 예를 들어, 소금과 설탕을 구분하는 용기가 동일하게 생겨 50%의 확률로 잘못된 소금/설탕을 뿌린다고 가정한다. 소금을 착각하여 설탕을 뿌려버렸고 이에 고기를 먹지 못하고 다 버리게 된다. 스마트 냉장고와 스마트 인덕션, 스마트 식기세척기 모두 정상적으로 동작하며 주고받는 정보 역시 문제없다. 따라서 사람의 영양 상태

는 아직 섭취 전임에도 스마트홈은 섭취 후로 인식을 하게 되며 잘못된 정보를 제공하여 스마트홈 SoS가 제공하는 건강 관리의 목표 달성도를 저해하게 된다.

## 5. 결론 및 향후 연구

크고 복잡한 SoS에서 발생할 수 있는 불확실성은 SoS의 실제 성능에 큰 영향을 미칠 수 있는 요인들이기 때문에 SoS 공학에서 반드시 고려되어야 한다. 불확실성에 대해 체계적인 분석 관점을 제공하는 SAS에서의 기존의 불확실성 분류 체계를 확장하여 SoS의 불확실성 요소들을 분석하고 발현 위치 기반으로 총 다섯 가지의 기준(문맥, 구조, 입력 변수, 매니저, 런타임)으로 SoS에서의 불확실성 요소들을 분류하였다. 또한, 제안된 분류 기준을 두 종류의 SoS 시나리오에 사례 적용하여 알아보았다.

향후 연구로는 발현 위치 이외에도 습성이나 수준과 같이 다른 지표를 토대로 한 불확실성 요소의 분류 기법을 만들 예정이다. 이를 통해 SoS를 모델링 할 때 고려해야 할 불확실성 요소들을 더욱 구체화할 것이다. 둘째로는, SoS에서의 불확실성을 모델링하고, 모델링된 불확실성으로 토대로 더욱 명확한 분류 기준을 세울 것이다. 마지막으로 VENTOS[12]와 같은 군집은행 시뮬레이터에서의 불확실성 요소를 고려하는 방안을 생각해 불확실성을 고려한 시뮬레이터를 구현할 예정이다.

## References

- [1] J. Boardman and B. Sauser, "System of Systems - The meaning of of," *2006 IEEE/SMC International Conference on System of Systems Engineering*, Vol. 2006, No. April, pp. 118-123, 2006.
- [2] D. Seo et al., "Modeling and verification for different types of system of systems using PRISM," *2016 IEEE/ACM 4th International Workshop on Software Engineering for Systems-of-Systems (SESoS)*, pp. 12-18, 2016.
- [3] D. Perez-Palacin and R. Mirandola, "Uncertainties in the modeling of self-adaptive systems," *Proc. of the 5th ACM/SPEC international conference on Performance engineering*, pp. 3-14, 2014.
- [4] J. Marvin, B. Morantz, T. Whalen, R. Deiotte, and R. K. Garrett, "8.3.2 Uncertainty Quantification (UQ) in Complex System of Systems (SoS) Modeling and Simulation (M&S) Environments," *INCOSE International Symposium*, Vol. 24, No. 1, pp. 843-858, 2014.
- [5] S. Baldi, M. R. Rosa, P. Frasca, and E. B. Kosmatopoulos, "Platooning merging maneuvers in the presence of parametric uncertainty," *IFAC-Papers-OnLine*, Vol. 51, No. 23, pp. 148-153, 2018.
- [6] Agusdinata, D. Buyung, "Dealing with Complexities

and Uncertainties in a System-of-Systems: Case Studies on Urban Systems," *System of System Failures*, pp. 95-113, 2018.

- [7] J. Song, J. O. Topping, S. Hyun, E. Jee, and D. H. Bae, "Slicing Executable System-of-Systems Models for Efficient Statistical Verification," *2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES)*, pp. 18-25, 2019.
- [8] B. H. C. Cheng et al., "Software engineering for self-adaptive systems: A research roadmap," *Lecture Notes in Computer Science*, Vol. 5525 LNCS, pp. 1-26, 2009.
- [9] A. J. Ramirez, A. C. Jensen, and B. H. C. Cheng, "A taxonomy of uncertainty for dynamically adaptive systems," *2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 99-108, 2012.
- [10] N. Esfahani, E. Kouroshfar, and S. Malek, "Taming uncertainty in self-adaptive software," *Proc. of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, pp. 234-244, 2011.
- [11] T. D. P. Mendes, R. Godina, E. M. G. Rodrigues, J. C. O. Matias, and J. P. S. Catalão, "Smart home communication technologies and applications: Wireless protocol assessment for home area network resources," *Energies*, Vol. 8, No. 7, pp. 7279-7311, 2015.
- [12] M. Amoozadeh, H. Deng, C. N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Vehicular communications*, Vol. 2, No. 2, pp. 110-123, 2015.

### 신 승 철

2019년 한국과학기술원(KAIST) 전산학부 심화과정 학사. 2019년~현재 한국과학기술원(KAIST) 전산학부 석사과정. 관심 분야는 시스템 오브 시스템즈 공학, 소프트웨어 모델링



### 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



송 지 영

2014년 이화여자대학교 컴퓨터공학과 졸업(학사). 2016년 KAIST 전산학부 졸업(석사). 2016년~현재 KAIST 전산학부 박사과정. 관심분야는 소프트웨어 공학, 소프트웨어 검증 및 테스트, 시스템 오브 시스템즈



배 두 환

1980년 서울대학교 공과대학 학사. 1987년 University of Wisconsin-Milwaukee 전산학 석사. 1992년 University of Florida 전산학 박사 졸업. 1995년~현재 KAIST 전산학부 교수. 관심분야는 Software Engineering for System of Systems, Software Safety Modeling and Analysis

Software Safety Modeling and Analysis