

저자 (Authors) 신승철, 현상원, 송지영, 배두환
Seungchul Shin, Sangwon Hyun, Jiyoung Song, Doo-Hwan Bae

출처 (Source) [한국정보과학회 학술발표논문집](#), 2019.12, 278-280 (3 pages)

발행처 (Publisher) [한국정보과학회](#)
The Korean Institute of Information Scientists and Engineers

URL <http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09301560>

APA Style 신승철, 현상원, 송지영, 배두환 (2019). 시스템 오브 시스템즈에서의 불확실성 요소 분석: 군집운행 시나리오에서의 사례 연구. 한국정보과학회 학술발표논문집, 278-280.

이용정보 (Accessed) KAIST
143.248.47.***
2020/03/06 10:18 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

시스템 오브 시스템즈에서의 불확실성 요소 분석: 군집운행 시나리오에서의 사례 연구*

신승철[○] 현상원 송지영 배두환

한국과학기술원 전산학부

{scshin, swwhyun, jysong, bae}@se.kaist.ac.kr

Analysis of Uncertainty Factors in System of Systems: Case Study in Platooning Scenario

Seungchul Shin[○] Sangwon Hyun Jiyong Song Doo-Hwan Bae

School of Computing, KAIST

요 약

대규모 복잡 시스템인 시스템 오브 시스템즈 (System-of-Systems, SoS)는 다양한 구성시스템 (Constituent System, CS)들로 복잡하게 구성되어 있다. 기존에 연구되어 왔던 시스템 자체의 특성으로 인해 관찰되는 불확실성 요소 이외에도 CS들의 독립성과 구성으로 인해 여러 불확실성 요소들이 관찰된다. 불확실성 요소를 고려하지 않는 경우 SoS의 성능에 악영향을 미칠 수 있다. 그러나 SoS에 존재하는 불확실성 요소들을 체계적으로 분석하는 연구가 부족하다. 따라서 본 논문에서는 SoS의 불확실성 요소를 관찰되는 위치 (Location)에 따라 문맥, 구조, 입력 변수, 매니저, 런타임의 5가지로 분류하는 체계를 소개한다. 본 논문에서 제안하는 분류 체계에 대한 이해를 돕기 위해 SoS인 군집운행 (Platooning) 시나리오에 적용하여 관찰되는 매니저와 런타임의 불확실성 요소에 대해 알아본다. 본 연구에서 제시한 SoS의 불확실성 분류 체계를 통해 SoS의 모델링 과정에서 기존에 고려되지 못했던 불확실성 요소들을 고려하여 현실에 더욱 가까운 결과물을 보일 수 있다.

1. 서 론

시스템 오브 시스템즈 (System-of-Systems, 이하 SoS)는 다양한 시스템들의 상호작용을 통하여 개별 구성시스템 (Constituent systems, 이하 CS)으로는 달성할 수 없었던 상위 단계의 목표를 달성할 수 있게 하는 대규모 복잡 시스템이다[1]. 상위 단계인 SoS 단계의 목표를 가지면서 SoS 단계의 운영 지침을 따르지 않는 SoS를 협력적 SoS (Collaborative SoS)라 한다[2]. 협력적 SoS는 각 CS에 상위 단계의 목표가 강제되지 않지만, CS들의 자발적인 참여로 인해 상위 단계의 목표가 달성된다. 협력적 SoS에서는 내부 CS에 관한 규정과 능력이 투명하게 매니저에게 전달되지 않는다. 이러한 정보의 부족함과 같은 요소들에 의하여 SoS 내에서 불확실성이 발생할 수 있다. 이들을 고려하지 않는 경우 SoS의 성능에 악영향을 미칠 수 있다. 그러나 SoS의 복잡함으로 인해 불확실성에 대한 고려가 더욱 힘들다.

SoS와 유사한 시스템으로 자가 적응 시스템 (Self-Adaptive System, 이하 SAS)이 있다. SAS는 환경이나 요구사항의 변화에서도 목표를 달성하는 시스템을 말한다[10]. 불확실성에 관한 연구는 SAS 분야의 연구자들에 의해 활발히 진행되어 왔다. 그러나 기존 SAS에서 연구되어 왔던 불확실성 요소의 분류체계들에 포함될 수 없는 불확실성 요소들이 존재한다. 따라서 본 연구에서는 SoS에서의 불확실성 요소들을 위치 (Location)를 기준으로 분류하는 분류 체계를 제안한다. 분

류 기법의 적용성을 알아보기 위해 SoS인 군집운행 시나리오에 적용하여 불확실성 요소를 확인한다.

본 연구에서 다루는 SoS로 군집운행 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.

본 연구의 구성은 다음과 같다. 2장에서 SoS에서의 불확실성 요소의 분류체계를 소개한다. 3장에서는 SoS인 군집운행 시나리오를 이용하여 발생할 수 있는 불확실성 요소의 사례를 설명한다. 마지막으로 4장에서는 결론 및 향후 연구를 기술한다.

2. 시스템 오브 시스템즈에서의 불확실성 요소 분류

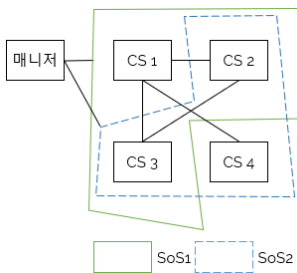
본 장에서는 기존에 다루어졌던 SAS에서의 불확실성에 대한 분류 체계 소개와 이를 기반으로 고안해낸 SoS에서의 위치 기반 불확실성 요소 분류를 설명한다.

SAS에서는 시스템을 모델링 할 때 관찰할 수 있는 불확실성 요소를 위치에 따라 문맥 (Context), 구조 (Structure), 입력 변수 (Input Parameter)로 분류했다. 그러나 협력적 SoS는 단일 시스템이 아닌 다양한 개별 시스템이 독립적으로 상호작용하여 다른 불확실성 요소들을 관찰 할 수 있다. 따라서 SAS에서의 분류 체계를 그대로 적용할 수 없으며 SoS에서의 새로운 분류 체계가 필요하다.

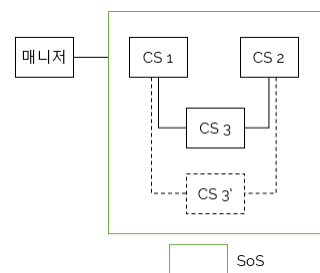
* 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No. 2015-0-00250, (SW 스타랩) 모델 기반의 초대형 복잡 시스템 분석 및 검증 SW 개발)과 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임 (2017M3C4A7066212)

[표 1] SoS 불확실성 분류 체계를 이용한 기존 시스템 불확실성 문헌 분석

문헌	불확실성				
	문맥	구조	입력 변수	매니저	런타임
F.S. Hover 2009[6]	X	X	O	X	O
N. Esfahani FSE/2011[7]	X	O	O	X	O
A.J. Ramirez SEAMS/2012[3]	O	X	O	X	X
P. Diego ICPE/2014[4]	O	O	O	X	X
S. Baldi IFAC/2018[5]	X	X	O	X	X
F. Acciani ECC/2018[8]	X	X	O	X	X



[그림 1.1] 문맥 불확실성



[그림 1.2] 구조 불확실성

SoS에서의 불확실성 요소는 다음과 같이 분류된다. 첫 번째로, SoS는 시스템 자체가 가지는 불확실성 요소를 포함한다. 기존의 SAS에서 시스템의 불확실성 요소를 위치에 따라 분류했던 기준을 SoS에 맞게 재해석하여 문맥, 구조, 입력 변수에 따른 불확실성을 정의한다. 두 번째로, 기존의 단일 시스템에서는 고려할 수 없었던 SoS의 매니저로 인한 불확실성을 새로 정의한다. 마지막으로 앞선 네 가지 위치에서 발견될 수 없는 런타임에서 발견되는 불확실성을 새로 정의한다. SoS에서의 문맥, 구조, 입력 변수, 매니저와 런타임에서 발견할 수 있는 불확실성에 대한 정의는 다음과 같다.

문맥에서 관찰되는 불확실성이란 실제 시스템의 요소들이 시스템의 모델에 포함되지 않아서 발생하는 불확실성이다. 따라서 문맥으로 인한 불확실성은 현실에 비한 모델의 완성도를 고려한다. [그림 1.1]에서 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 1.1]에서 SoS2는 CS2, CS3, CS4만 포함하고 있으므로 CS1으로 인한 영향을 고려하지 못하게 된다. 두 가지 예시의 SoS는 실제 SoS의 구성 요소들을 모두 고려하지 않았기에 문맥에서 관찰되는 불확실성을 가지게 된다.

구조에서 관찰되는 불확실성은 고려되어야 하는 정보를 구성한 형태에서 관찰할 수 있는 불확실성이다. 따라서 현실을 얼마나 자세하게 반영했는지를 고려한다. [그림 1.2]에서 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.

입력 변수에서 관찰되는 불확실성은 모델에게 주어지는 입력값과 이를 조정하는 방법에서 발견되는 불확실성이다. 정보의 부족뿐만 아니라 정보 자체의 불확실함이 원인이 되며 대표적인 예시로 센서 소음 (Sensor noise)을 들 수 있다.

매니저에서 관찰되는 불확실성은 SoS의 매니저와 각 CS의 매니저 전체에서 관찰되는 불확실성을 말한다. 주로 매니저 간의 소통/정보의 부족과 동적 재구성으로 인해 일어나는데, 대표적으로 협력적 SoS에서 SoS 매니저의 각 CS 매니저가 정한 CS의 내부 정책에 대한 정보 부족 (Lack of Knowledge)에 의해 발생한다. 각 CS는 SoS를 구성하지만, 블랙박스로 취급되며, 서로 독립성을 갖고 있다. 따라서 CS 단계의 목표에 따라 동적으로 다른 CS와의 연결을 끊을 수 있으며, CS 단계의 목표가 SoS 단계의 목표와 충돌하는 경우가 발생할 수 있다. 각 CS의 관리 독립성 (Managerial Independence)에 의해 발생하는 정보 부족은 불확실성의 요소로서 관찰될 수 있다.

앞선 네 가지 위치에서 발견되는 불확실성이 아닌 나머지는 런타임에서 발견되는 불확실성으로 구분한다. 런타임에서 발견되는 불확실성은 런타임 환경에서만 관찰되는 불확실성을 말한다. 미리 모델링 할 수 없으며 실제 실행환경에서 주어지는 한정된 자원과 예측할 수 없는 환경과의 상호작용에서 관찰되는 불확실성이다.

위 [표 1]은 기존 불확실성 관련 연구들에서 본 연구에서 제시한 5가지 불확실성 요소들을 고려했는지를 보이는 표이다.

첫 번째 연구는 불확실성을 고려하여 시스템을 디자인하는 기본적인 지침에 관한 연구로 변수와 측정값에서의 불확실성, 시스템 자체의 실행환경 속에서 만날 수 있는 불확실성에 대해 다루고 있다[6]. 두 번째 논문은 자가 적응형 소프트웨어에서 불확실성을 어떻게 다루는지에 대해 기술했다. 불확실성을 크게 세 형태로 정리했는데, 시스템 변수에서의 불확실성과 분석 모델이 시스템의 행동에 대한 단순 가정을 함으로써 오는 불확실성, 그리고 사용자의 선호도로 인해 나오는 불확실성에 대해 다루고 있다[7]. 세 번째 논문은 동적 적응 시스템에서의 불확실성에 대해 용어를 정의하는 템플릿을 정형화한 논문으로 불확실성을 발생하는 개발 단계를 기준으로 요구사항, 디자인, 그리고 실행 단계에

따라 나누었다[3]. 네 번째 논문은 자가 적응 시스템에서의 불확실성을 모델링 하는 방법에 대해 다루며 자가 적응 시스템에서 관찰되는 불확실성을 여러 가지 방법으로 분류해놓았다[4]. 다섯 번째 논문은 군집운동에서 변수에 불확실성이 있는 경우의 합병 기동에 대해 다룬 논문이며 변수에서 관찰되는 불확실성에만 집중했다[5]. 여섯 번째 논문은 적응형 순항 제어에서의 신뢰할 수 없는 네트워크에 대해 패킷 로스를 줄일 방법에 관해 기술한 논문이다. 여기서는 변수에 대한 노이즈 등의 불확실성에 집중하여 기술하였다[8].

이처럼 SoS의 특성으로 인해 발견되는 매니저에게서의 불확실성과 런타임에 대한 불확실성에 대한 고려는 기존 연구들에서는 체계적으로 이루어지지 않고 있다.

3. 군집운동 시나리오에의 사례 적용

본 장에서는 매니저와 런타임에서의 불확실성은 어떠한 양상으로 나타날지에 대해 군집운동 시나리오에 대한 설명과 그 경우 일어날 수 있는 매니저와 런타임에서의 불확실성에 대해 알아본다.



[그림 2] 4차선 고속도로에서 군집운동 중인 트럭 세 대

매니저에서 관찰되는 불확실성은 SoS 매니저와 CS 매니저 간의 정보 및 소통의 부족으로 인해 일어나는 경우가 대표적이다. 서로 다른 군집운동 시스템을 가진 차량들이 하나의 군집을 구성하는 경우에 쉽게 관찰할 수 있다. [그림 2]에서 세 트럭은 서로 다른 군집운동 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 환경요소는 SoS로 인한 부가효과 (Side Effect)가 환경에 미친 영향이 다시 SoS에 영향을 미치게 되는 경우와 예측할 수 없는 사람으로 인한 불확실성 요소를 포함한다. [그림 2]에서 나타난 세 대의 트럭은 정방향의 2차선이라는 제한된 도로 자원 중 한 차선의 많은 부분을 소모하고 있다. 따라서 환경요소로 생각될 수 있는 군집운동을 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 레벨의 목표 달성에 실패하는 사례이다.

4. 결론 및 향후 연구

크고 복잡한 SoS에서 발생할 수 있는 불확실성을 분류하고 고려하는 것은 SoS의 실제 성능에 큰 영향을 미치는 요인들이기 때문에 모델링 과정에서 반드시 고려되어야 한다. SoS와 비슷한 특성을 공유하는 SAS에서의 기존 연구내용들은 구분할 수 없는 불확실성 요소들을 분석하고 이를 군집운동 시나리오에서의 사례를 통해 알아보았다. 향후 연구로 첫째로는 SoS에서의 불확실성에 대한 사례 연구를 다양한 도메인에 적용해보고, 위치 (Location)이외에도 습성 (Nature)이나 단계 (Level)와 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 VENTOS[9]와 같은 군집운동 시뮬레이터에서의 불확실성 요소를 고려하는 방안을 생각해 불확실성을 고려한 시뮬레이터를 구현할 예정이다.

참고 문헌

- [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." IEEE/ACM 4th International Workshop on Software Engineering for System-of-Systems (SESos), 2016.
- [3] J. Ramirez et al., "A taxonomy of uncertainty for dynamically adaptive systems." Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. IEEE Press, 2012.
- [4] D. Perez-Palacin et al., "Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation." Proceedings of the 5th ACM/SPEC international conference on Performance engineering. ACM, 2014.
- [5] S. Baldi et al., "Platooning merging maneuvers in the presence of parametric uncertainty." IFAC-PapersOnLine, 2018.
- [6] S. Franz et al., "System design for uncertainty." Mass. Inst. Technol, 2009.
- [7] N. Esfahani et al. "Taming uncertainty in self-adaptive software.", Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering. ACM, 2011.
- [8] A. Francesco et al., "Cooperative adaptive cruise control over unreliable networks: an observer-based approach to increase robustness to packet loss." 2018 European Control Conference (ECC). IEEE, 2018.
- [9] Amoozadeh et al., "Platoon management with cooperative adaptive cruise control enabled by VANET." Vehicular Communications 2.2, 2015.
- [10] B. Cheng et al., "Software Engineering for Self-Adaptive Systems: A Research Roadmap." Software Engineering for Self-Adaptive Systems, LNCS Hot topics, 2009