

교통 분야 시스템 오브 시스템즈 대상

SUMO의 모델링 적합성 분석

윤원경, 송지영, 지은경, 배두환

KAIST 전산학부

{ wkyun,jysong,ekjee,bae}@se.kaist.ac.kr

A Suitability Analysis of SUMO

for Modeling System of Systems in Traffic Domain Example

Wonkyung Yun, Jiyoung Song, Eunyoung Jee, Doo-Hwan Bae

School of Computing, KAIST

요 약

시스템 오브 시스템즈(System of Systems, SoS)는 기능적 독립성과 관리적 독립성을 갖는 시스템들의 집합이다. 개별 시스템의 목표보다 더 높은 레벨의 목표를 달성하기 위한 복잡한 시스템의 필요성이 증가함에 따라 SoS의 연구가 점차 활발해지고 있으며, 다양한 상황 하에서 SoS가 실제로 목표를 달성할 수 있는지 확인하기 위한 검증의 필요성도 증가하고 있다. 따라서 본 연구에서는 많은 SoS 연구에서 사례로서 사용되는 교통 여제를 대상으로 하여, 교통 상황의 미시적 시뮬레이션 및 검증을 위해 Multi-Agent System(MAS) 분야에서 쓰이는 Simulation of Urban MObility(SUMO)를 활용하였다. SUMO를 통해 예제 시나리오를 모델링하며, 8가지의 SoS 특성을 기준으로 SUMO의 SoS 모델링 적합성을 분석하였다.

1. 서론

경제, 사회의 각 분야에서 정보화가 이루어짐에 따라 시스템의 규모가 점차 성장하고 있다. 단일 시스템의 기능으로는 해결할 수 없는 복잡한 문제가 발생하고 있으며 이를 해결하기 위해 여러 시스템의 협업이 증가하고 있다. 이러한 추세와 맞물려 시스템 오브 시스템즈(System of Systems, SoS)의 개념이 등장하였는데, SoS는 자율성을 갖는 독립적인 개별 시스템(Constituent System, CS) 간의 협력을 통해 개별 시스템이 해결할 수 없는 고차원적 목표를 달성하기 위한 시스템들의 집합이다. 이 때, SoS를 구성하는 각각의 시스템은 개별 목표를 달성하기 위해 기능을 하는 동시에 공동의 목표를 달성하기 위해 시스템간 혹은 SoS 레벨의 관리자와 상호작용을 한다[8].

고차원적 목표를 달성하기 위한 SoS의 연구가 점차 활발해지고 있으나, SoS의 복잡성과 미래 지향적인 특징으로 인해 연구대상으로서 활용할 수 있는 사례 시스템이 부족한 실정이다. 때문에 SoS의 특성을 잘 표현할 수 있는 사례를 구축하고자 하는 연구가 진행되고 있으며[1,2] 특히 교통 분야에서 SoS의 여제를 끌어내고자 한 시도들이 있었다[3,4].

한편, 다수의 시스템으로 구성, 시스템 간의 상호 협력 등 SoS와 여러 공통 점을 공유하는 다중 에이전트 시스템(Multi-Agent System, MAS) 분야에서는 교통 대상 MAS의 시뮬레이션과 검증을 위해 Simulation of Urban MObility(SUMO)[5]라는 시뮬레이터를 활용해왔다[6,7]. SUMO는 교통 상황의 세부적인 특성을 모델링 할 수 있을 뿐 아니라 시뮬레이션 결과를 시각적인 결과로써 도출한다. 그러므로 SUMO를 SoS의 사례 검증을 위한 시스템으로 활용할 경우 다양한 상황 하에서 SoS 레벨 목표의 달성 여부를 시뮬레이션을 통해 파악할 수 있을 것으로 기대된다. 본 연구에서는 SUMO가 SoS의 특성을 모델링 할 수 있는가를 기준으로 삼고, SUMO가 SoS의 사례를 서술하기 위한 시스템으로 적합하게 사용될 수 있는지 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 SoS의 개념 및 특성을 설명하고

SUMO에 대한 기본적인 배경지식을 설명한다. 3장에서는 예제 시나리오를 기반으로 SoS의 8가지 특성을 SUMO 코드를 통해 구현한 결과를 기술하고, 각 SoS 특성에 대한 SUMO의 모델링 적합성을 분석한다. 4장에서는 결론 및 향후 연구 방향에 대해 논의한다.

2. 배경지식

2.1. 시스템 오브 시스템즈(SoS)의 특성

SoS의 정의에 대해서는 다양한 견해가 존재하나, 대다수의 연구에서 Maier의 정의를 공유하고 있다[8]. Maier에 따르면 SoS란 기능적 독립성(operational independence)과 관리적 독립성(managerial independence)을 가지는 시스템들의 집합이며, SoS 레벨 관리자의 지시 혹은 스스로의 판단에 따라 행동함으로써 개별 시스템의 목표보다 더 높은 레벨의 목표를 달성하기 위해 조직된 하나의 거대 시스템이다[8]. Nielson에 따르면[9], SoS는 자율성(Autonomy), 독립성(Independence), 분산(Distribution), 진화(Evolution), 동적 구조 변경(Dynamic Reconfiguration), 창발성(Emergence), 상호의존성(Interdependence), 상호운용성(Interoperability)의 8가지 특성을 가지며 각 특성에 대한 설명은 [표 1]에 정리된 바와 같다. 이 중 진화와 동적 구조 변경의 경우, 두 특성 모두 SoS에 변화를 일으킨다는 점은 서로 유사하나 변화의 시점과 계획의 유무 측면에서는 차이점을 갖는다. 즉, 진화는 SoS 레벨 관리자의 개입을 통해 이루어지며, 수행 결과의 분석을 통해 SoS가 다음에 수행하는 기능에 계획적인 변경이 생기는 것을 의미한다. 반면 동적 구조 변경의 경우 기능을 수행하는 도중에 효율적인 기능의 수행을 위하여, 계획과는 상관 없이 일어나는 SoS의 구성의 변화를 말한다.

한편 독립성과 상호의존성의 관계에서 보듯이 특성 간에 서로 상충될 수 있는 가능성이 있으므로, 특정 SoS가 [표 1]에 기술된 모든 특성을 갖지 않을 수 있다. 본 연구에서는 SUMO가 SoS 사례를 모델링 할 수 있는 시스템으로서

적합한가를 평가하기 위하여, SoS의 각 특성을 SUMO 모델링을 통해 얼마나 표현할 수 있는지를 기준으로 삼았다.

SoS 특성	설명
자율성 (Autonomy)	CS는 SoS에 참여함과 동시에 CS 각자의 규칙에 의해 통제/지배를 받는다.
독립성 (Independence)	CS가 SoS로부터 분리될 경우에도 자신의 기능을 충분히 수행할 수 있다.
분산 (Distribution)	CS는 다른 CS 및 SoS와 연결이 되어, 통신과 정보 공유를 할 수 있도록 흩어져 있다.
진화 (Evolution)	SoS는 계획적으로 변화하며, 이러한 변화는 느린 수준의 개입을 통해 지속적으로 일어난다.
동적 구조 변경 (Dynamic reconfiguration)	계획된 개입 없이, SoS의 구조와 구성이 변경될 수 있다.
창발성 (Emergence)	CS들의 협력으로 인한 시너지적인 효과가 행동으로써 나타난다.
상호의존성 (Interdependence)	CS는 SoS 수준의 목표를 달성하기 위해 다른 CS에게 의존한다.
상호운용성 (Interoperability)	SoS는 이종(Heterogeneous)의 CS들을 포함한다.

[표 1] SoS의 8가지 특성과 정의

2.2. Simulation of Urban Mobility (SUMO)

SUMO는 교통 상황의 미시적 시뮬레이션을 위해 Berlin Institute of Transportation Systems 에서 개발한 오픈 소스 도구이다. SUMO를 활용하여 자동차의 가속도, 크기뿐 아니라 각 신호등의 신호 주기에 이르기까지 도로 교통의 세부적인 특성을 디자인하고 이에 따른 결과를 시뮬레이션을 통해 시각적으로 확인할 수 있다. 시뮬레이션을 위해서는 기본적으로 길(edge)과 길이 교차하는 지점(node)에 대한 정보를 포함하는 지도(net) 파일과, 차량(vehicle)에 대한 모델과 차량의 인스턴스(instance), 각 인스턴스에 맞는 운행 경로(route)에 대한 정보를 포함하는 파일 등이 필요하다. 위의 모든 파일들은 XML 형태로 기술되며, 이 파일들을 시뮬레이터에 입력하면 지정한 시작 시간과 종료 시간 내에서 변화하는 교통 상황의 시뮬레이션 결과가 시각적인 애니메이션 또는 XML 파일로써 도출된다.

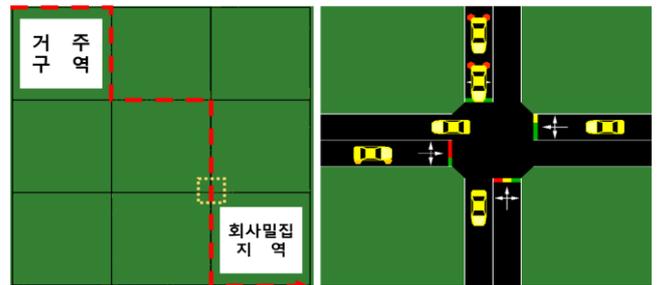
3. SUMO의 SoS 사례 시스템 모델링 적합성 분석

3.1. SoS 예제 시나리오

본 연구에서 기반으로 하는 시나리오는 출근 시간에 발생하는 도시의 교통 혼잡을 기본으로 하며, 아래와 같은 제약을 가진다.

- 1) 도시는 [그림 1]과 같이 거주 구역과 회사 밀집 지역으로 나뉘어져 있으며, 출근 시간에는 거주 구역에서 회사 밀집 지역으로 이동하는 차량의 수가 다른 경로로 이동하는 차량의 수보다 2배 이상 많다.

- 2) 교차로의 각 방향으로 배치된 교통 통제 장치가 CS가 되며, 하나의 CS는 해당 구역 도로의 교통량을 감지하는 카메라와 일정한 주기를 가지고 신호를 반복하는 신호등으로 구성되어 있다.
- 3) 거주 구역에서 회사 밀집 지역으로 가는 경로 상의 차량이 미리 지정한 제한 숫자 이상이 될 경우 이를 교통 혼잡 상황이라 칭하며, 제한 숫자는 시뮬레이션 이전에 주어진다. 교통 혼잡 상황이 발생하면 SoS 레벨 관리자는 혼잡 경로 상에 위치한 CS를 SoS로 구성하여 교통 통제를 시작하며, 이 때 CS는 자신의 신호 주기와 상관 없이 SoS 레벨 관리자의 지시에 따른 신호 주기를 반복한다. SoS 레벨 관리자는 교통 혼잡 시에 SoS에 포함된 CS에게 회사 밀집 지역 방향의 신호를 모두 초록불로 바꾸도록 명령하며, 혼잡 구간의 차량 수가 줄어들어 혼잡이 해소되어도, 혼잡 해소 후 바로 재발하는 것을 막기 위해 최소 15초간 통제를 지속한다.



[그림 1] SoS 예제 도로 구조도(좌)와 일부 교차로 확대도(우)

3.2. SoS 특성 기반 SUMO 적합성 분석 및 모델링 사례

이 장에서는 3.1에 기술한 시나리오를 SUMO를 이용해 모델링하고, [표 1]에 명시된 SoS 특성들이 어떻게 모델링 될 수 있는지 분석한다. SUMO를 이용한 예제 시나리오의 구현은 실시간 상호 작용을 위해 확장된 SUMO API 인 TraaS(TraCI as a Service) 를 함께 활용하였다[10].

먼저 자율성은 [그림 2]와 같이 교통 혼잡 상황일 때는 SoS 레벨 관리자의 지시에 따라 통제될 수 있는 동시에, CS가 적절한 신호 주기에 따라 해당 구역의 교통을 원활하게 하는 것을 통해 모델링 될 수 있다. 또한 독립성은 SoS에 속해 관리되지 않을 때에도 해당 구역의 교통 원활과 교통량 감시하는 기능을 스스로 수행해낼 수 있다는 점을 통해 구현된다. 즉, SoS 레벨 관리자의 통제 하에서 수행하게 되는 일과 통제되지 않는 상황에서 수행하는 일이 분리되어 있어, CS 각자의 규칙을 가지고 기능을 수행하기 때문에 자율성과 독립성 모두 SUMO에서 모델링 될 수 있다. [그림 3]의 10번째 줄부터 16번째 줄의 코드를 통해 SoS 상태(혼잡 상태)와 비 SoS 상태(혼잡하지 않은 상태)로 나누어져 CS의 신호등이 다르게 동작하는 것을 볼 수 있다.

분산 특성의 경우, [그림 1]과 같이 각 CS가 혼잡 경로상의 교차로에 지리적으로 분포되어 있으며 SoS 레벨 관리자를 중심으로 소통할 수 있으므로 구현 될 수 있다. 이 때 분산 되어있는 각 CS의 위치는 SUMO 파일 상에서 x, y 좌표로써 정의된다.



[그림 2] 자율통제 하의 교차로(좌)와 혼잡으로 인한 통제시의 교차로(우)

```

Procedure SoSonSUMO ( Sumofiles files )
1: t := simulation end time
2: max := # of cars to delegate the control authority to SoS manager
3: initiate traffic lights and cameras with files
4: run the simulator
5: while (current time ctime does not reach to t)
6:   generate a car and assign its route with the specific probability
7:   n := count the number of the cars on congestion area at ctime
8:   if (n > max)
9:     set the SoS phase
10:  if (SoS phase)
11:    update the traffic lights with the direction from residence
    area to business area on congestion distinct to green
12:    update the other lights to red
13:    if (SoS phase continues for 15 ticks)
14:      set the non-SoS phase
15:  else if (non-SoS phase)
16:    update each traffic lights as their cycles
    
```

[그림 3] SUMO API Traas를 이용한 시나리오 모델링 슈도 코드

다음으로 진화의 경우 2.1에 기술된 바에 따라 SoS가 수행하는 기능 자체에 변화가 생겨야 하므로, SoS가 CS의 신호 주기 변경만을 지시하는 현재의 시나리오에서는 변화 특성의 모델링이 불가능하다. 즉, 진화를 모델링 하기 위해서는 시뮬레이션이 종료된 후 나타난 결과를 바탕으로 SoS 레벨 관리자의 분석적인 판단과 계획에 따라 기능이 변경되어야 하므로, 사람의 개입이 필수적으로 동반되어야 한다. 반면 동적 구조 변경의 경우, 기능 수행 도중 동적으로 회사 밀집 지역이 변경될 때 참여하는 CS가 다르게 구성됨으로써 모델링 될 수 있다. 예를 들어, 회사 밀집 지역이 추가되어 혼잡 경로가 2개 이상 존재하는 시나리오에서는 어떤 경로에서 교통 혼잡이 발생했는지 판단하는 과정을 거쳐 SoS를 다르게 구성할 수 있으며 이 때 다른 경로 상의 CS가 SoS에 추가, 삭제됨에 따라 동적 구조 변경이 발생하게 된다.

창발성은 SoS를 구성하지 않고 개별 CS가 각각 행동했을 때의 결과와, SoS의 구성 시에 나타나는 결과를 비교함으로써 나타난다. 그러므로 코드 상에 구현될 수 없으며 시뮬레이션의 결과로서 확인된다. 교통 혼잡 상황을 통제할 경우와 전혀 통제하지 않은 경우를 비교하는 실험을 수행 시, 혼잡 상황을 통제했을 시에 차량의 목적지 도달률이 통제를 하지 않았을 때보다 최대 2배 이상 높아지는 것을 확인할 수 있었다.

상호의존성의 경우는 혼잡 경로에 포함되지 않는 CS는 SoS 레벨 관리자의 지시에 따라 자신의 신호를 빨간 불로 바꾸는 행동에서 나타날 수 있다. 해당 CS는 신호를 빨간 불로 바꾸므로 구역을 지나는 자동차들의 운행을 통제하게 되며, 담당 구역을 원활하게 하려는 CS 수준의 목표와 배치되어 희생을 하게 된다. [그림 2]에서 교통 혼잡시의 SoS 레벨 관리자의 통제가 이루어짐으로써 다른 방향을 담당하는 CS의 신호등이 빨간 불로 바뀌어 차량이 통제되었음을 볼 수 있으며 이와 같이 상호의존성이 발현된다. [그림 3]의 11번째 줄의 코드에서 혼잡 구간에 위치한 회사 밀집 지역 방향의 신호등을 제외한 나머지 신호등을 모두 빨간 불로 변경하는 코드를 확인할 수 있다.

상호운용성의 경우 본 시나리오에서의 CS는 모두 같은 종류로 이루어져있어 모델링 되지 않았다. 하지만 혼잡 구간의 일부 CS에 배기가스 측정 시스템과 같은 추가적인 장치를 설치하여 배기가스를 주기적으로 측정하는 코드를 구현한다면 본 특성 또한 모델링이 가능할 것으로 예상된다.

따라서 [표 2]와 같이, SUMO를 통해 SoS의 8개의 특성 중 7개의 특성을 SUMO 및 API를 활용하여 모델링이 가능한 것으로 판단하였다. 이로써 SUMO가 SoS의 다양한 특성을 표현하는 시스템으로써 활용될 수 있는 가능성을 확인할 수 있었다.

4. 결론

본 연구에서는 출근 시간대에 발생하는 교통 혼잡 예제 SoS 시나리오를 대

상으로 SUMO 및 TraaS API를 활용함으로써 모델링 하였으며, 이를 바탕으로 SUMO가 8가지의 SoS 특성을 적합하게 모델링 할 수 있는지 분석하였다. 분석 결과, [표 2]와 같이 SUMO 파일, TraaS API 및 결과 수준에서 SoS의 8가지 특성 중 1가지의 특성을 제외한 7가지의 특성을 모델링 할 수 있음을 알 수 있었다.

향후 연구로서 SUMO를 활용하여 추가적인 시나리오를 모델링 할 계획이며, 이를 통해 SoS 수준의 목표가 이루어질 수 있는지에 대한 여부뿐 아니라 목표를 달성할 수 있는 SoS 구성 전략의 평가 등의 다양한 목적의 검증이 수 행할 것이다.

특성	가능 여부	모델링 레벨
자율성	O	API
독립성	O	API
분산	O	SUMO 파일
진화	X	X
동적 구조 변경	O	API
창발성	O	결과
상호의존성	O	API
상호운용성	O	API

[표 2] SoS의 특성별 모델링 가능 여부 및 모델링의 레벨

Acknowledgement

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. R0126-16-1101, (SW 스타트업) 모델 기반의 초대형 복잡 시스템 분석 및 검증 SW 개발)

Reference

[1] D.Seo et al., "Case study on Simulation for System of Systems Goal Verification", Korea Conference on Software Engineering (2016)

[2] D.B.Agusdinata et al., "Specification of System-of-Systems for Policymaking in the Energy Sector", Integrated Assessment Journal 8, 2(2008)

[3] D.A.DeLaurentis, "Understanding Transportation as System-of-Systems Design Problem", In 43rd AIAA Aerospace Science Meeting and Exhibit (2005)

[4] M.Mansouri et al., "Maritime Transportation System of Systems Management Framework: A System of Systems Engineering Approach", International Journal of Ocean Systems Management 1, 2(2009)

[5] D.Krajzewicz et al. "SUMO (Simulation of Urban MObility)-an open-source traffic simulation." Proceedings of the 4th Middle East Symposium on Simulation and Modelling (2002)

[6] T.Azevedo et al., "JADE, TraSMAP and SUMO: a tool-chain for simulating traffic light control", Proceedings of the 13th International Joint Conference on Autonomous Agent and Multiagent Systems (2014)

[7] J.T.Baumfalk et al., "Norm-based Distributed Controllers in a Multi-Agent System", Master's thesis (2015)

[8] M.W.Maier, "Architecting Principles for System-of-Systems", INCOSE International Symposium (1996)

[9] C.B.Nielsen et al., "System of Systems Engineering: Basic Concepts, Model-based Techniques, and Research Directions", ACM Computing Surveys, 48, 2 (2015)

[10] "TraaS - TraCl as a Service", <http://traas.sourceforge.net/cms/>