

# 사이버-피지컬 시스템의 상호운용성 상충관계 분석에 관한 연구

김의섭<sup>○</sup>, 유준범

건국대학교 컴퓨터공학과

atang34@konkuk.ac.kr, jbyoo@konkuk.ac.kr

## A Study on Interoperability Conflict Analysis for Cyber-Physical System

Eui-Sub Kim<sup>○</sup>, Junbeom Yoo

Division of Computer Science and Engineering

### 요 약

사이버-피지컬 시스템은 다양한 이종의 어플리케이션 및 시스템이 동적으로 결합되어 동작하는 개방형(open-ended) 시스템이다. 하지만 상충되는 목적을 가진 어플리케이션 및 시스템이 결합된다면 상호운용성을 저해할 수 있다. 상호운용성의 실패는 시스템의 예측할 수 없는 동작을 발생시키고, 이는 시스템의 기능적 실패뿐만 아니라 사고로 이어지기 때문에 상호운용성의 상충여부를 개발 초기에 분석할 필요가 있다. 본 논문에서는 요구사항 단계에서 상호운용성의 상충을 분석할 수 있는 방안과 이를 명세할 수 있는 SysML의 확장 형태를 제시한다. 이를 통해 상호운용성의 상충 여부를 미연에 식별하여 대처할 수 있을 것으로 기대 할 수 있다.

### 1. 서 론

사이버-피지컬 시스템(CPS: Cyber-Physical System)은 피지컬 컴포넌트와 사이버 컴포넌트가 네트워크로 밀접하게 연결된 시스템으로서, 사이버 컴포넌트가 피지컬 컴포넌트를 모니터링하고, 연산하여 피드백 루프를 통해 피지컬 컴포넌트를 제어하는 시스템이다. 현재 안전 필수 도메인을 비롯한 다양한 도메인(지능적 교통량 관리 및 스마트 공장, 헬스 케어, 스마트 그리드 등)에 적용되고 있으며, 이런 사이버-피지컬 시스템은 기존 시스템과 다른 다양한 지능적 기능을 제공함으로써 삶의 질을 높이거나 기술적 발전을 이루는데 앞장서고 있다[1].

사이버-피지컬 시스템은 다양하게 변화하는 실세계에서 동작하는 개방형(open-ended) 시스템이며[2] 이종(heterogeneous)의 시스템이 커뮤니케이션하는 시스템이다. 따라서 동일 또는 이기종의 시스템 및 제품, 서비스가 특별한 제약이 없이 서로 호환되어 사용될 수 있도록 상호운용성(Interoperability)이 보장되어야 하며[3], 새로운 시스템 및 제품, 서비스가 추가되더라도 기능적 저하가 발생하지 않아야 한다. 또한, 동적이고 불확실한 현실 세계의 다양한 상황에 대해서도 충분히 인식하여 대처 및 대응할 수 있는 능력이 요구되고 있다[4].

하지만 사이버-피지컬 시스템의 이종성과 동적인

환경, 다양한 시스템 및 제품, 서비스의 결합, 새로운 기능의 추가 등은 사이버-피지컬 시스템의 상호운용성 보장을 저해하는 요소이다. 특히 상충되는 목적을 가진 시스템 및 제품, 서비스가 결합되면, 사이버-피지컬 시스템이 어떤 동작을 할지 예측할 수 없게 되는 문제가 발생한다. 이런 문제는 단순히 시스템의 성능 저하뿐만 아니라 안전을 위협하는 사고로 이어지게 되기 때문에 반드시 개발 및 설계 단계에서 충분히 분석하고 해결해야 할 문제이다.

본 논문에서는 사이버-피지컬 시스템의 상호운용성의 내재된 상충 문제를 해결하고자 상호운용성의 충돌을 유발하는 상충관계 분석 방안을 제시한다. 제시하는 방안은 요구사항 분석 단계에서 기능적 요구사항과 함께 상충관계를 분석해 개발 초기에 상충관계를 식별하여 상호운용성을 확보할 수 있도록 지원한다. 이를 통해 전체 개발 비용을 절감하고 보다 안전한 시스템의 개발이 가능할 것이다. 이를 위해 상충관계에 대한 분류 및 분석 방법을 제시하고 이를 명세할 수 있도록 SysML[5]의 확장된 프로파일을 제시한다.

상호운용성의 상충관계를 요구사항 분석 단계에서 수행함으로써 상호운용성의 상충 여부를 미연에 식별하여 대처할 수 있도록 시스템 설계가 가능할 것으로 예상되며, 또한 새로운 시스템과의 결합을 가능하도록 유연하고 확장성 있는 시스템의 개발에 도움이 될 것으로 기대할 수 있다.

## 2. 사이버-피지컬 시스템의 상호운용성 상충관계

상호운용성이란 시스템 및 제품, 서비스 가 추가적인 노력 없이도 타 시스템 및 제품, 서비스와 함께 동작할 수 있는 능력이다[4]. 사이버-피지컬 시스템은 다양한 이종의 시스템 및 제품, 서비스가 커뮤니케이션 하는 시스템으로서 상호운용성의 보장이 중요한 시스템이다. 시스템 내부적으로는 새로운 서비스 및 어플리케이션이 추가되어 상호운용 될 수 있으며 시스템 외부적으로는 다른 사이버-피지컬 시스템과 상호운용 될 수 있다. 그림 1은 사이버-피지컬 시스템의 아키텍처와 상호운용되는 요소를 나타낸 그림이다.

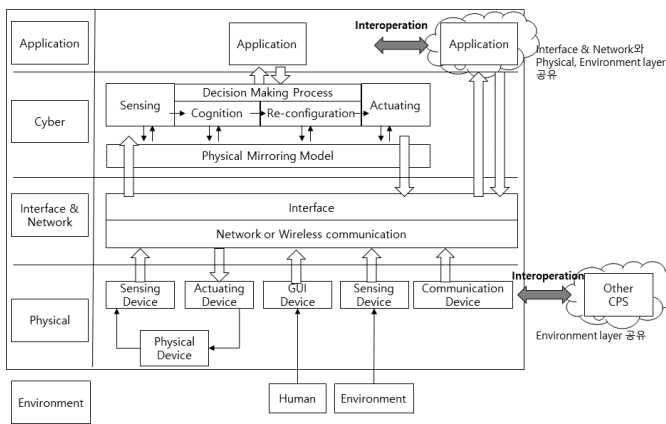


그림 1 사이버-피지컬 시스템의 아키텍처 및 상호운용성

상호운용성의 상충이란 상호 운용되어야 하는 어플리케이션 및 시스템이 제대로 커뮤니케이션을 하지 못해 상호 운용되지 못하는 정적인 상충(구조적 연결성 상충)과 서로 다른 어플리케이션 및 시스템이 각자의 연산 결과가 불일치하여 어떤 것을 수행해야 할지 판단하기 어려운 상황인 동적인 상충(기능적 상충)을 의미한다.

정적인 상충은 사이버 컴포넌트와 피지컬 컴포넌트의 상호운용 뿐만 아니라 이종의 서비스 및 어플리케이션, 이종의 네트워크 프로토콜, 제조업체가 다른 디바이스 및 어셋, 다른 사이버-피지컬 시스템과의 상호운용 등 다양한 곳에서 존재한다. 뿐만 아니라, 운용 중 다른 시스템 및 제품, 서비스와 동적인 조합, 유지보수를 위한 제품의 교체, 제품 및 어플리케이션의 업데이트 등에 의해서 정적 상충 상황이 발생한다. 해당 상충을 해결하기 위해서는 RAMI 4.0 표준[6]이나 AUTOSAR 표준[7]과 같은 산업 표준을 준수하여 개발하여 커뮤니케이션 측면에서 상충 상황이 발생하지 않도록 대비해야 한다.

반면, 동적인 상충은 기능적 상호운용성 상충을 의미한다. 정적인 상충은 참조 모델 및 표준 준수,

인터페이스 확인 등으로 보장 가능한 반면 동적인 상충은 분석 및 식별하기가 보다 어렵다. 예를 들어 고속도로의 사고를 감지하여 전광판 출력을 통해 사고를 대처하는 시스템에 새롭게 교통량을 확인하여 전광판을 통해 차로를 제어하는 서비스를 추가할 경우 두 서비스간 기능적 상호운용성 상충이 발생하여 사고 감지 및 대처 시스템은 전광판에 진입 통제 명령을 내리지만, 교통량 제어 시스템은 전광판에 진입 허용 명령을 내리는 상충상황이 발생할 수 있다. 문제는 이런 상충상황이 동적이고 무수히 많은 불확실성이 존재하는 실세계에서 모든 상황이 통제되도록 설계하는 것이 어렵다는 것이다. 따라서 이를 개발 단계 초기에 분석하여 대처할 수 있는 적절한 분석 방안이 필요하다.

- 정적 상충: 상호 운용되어야 하는 서로 다른 시스템 및 어플리케이션, 서비스가 커뮤니케이션에 실패하는 경우
  - 원인: 데이터 포맷의 불일치, 통신 프로토콜의 불일치, 커뮤니케이션 지연 등
  - 해결 방안: 데이터 포맷의 표준 확립과 준수, 커뮤니케이션을 위한 미들웨어 및 랩퍼 활용, RAMI 4.0 또는 AUTOSAR와 같은 참조 모델 및 표준 활용 등
- 동적 상충: 상호 운용되어야 하는 서로 다른 시스템 및 어플리케이션, 서비스의 판단 불일치로 어떤 판단을 따라 하는지 결정하지 못하는 경우
  - 원인: 커뮤니케이션 지연, 동일한 환경에 대한 불일치 인지, 기능적 상충 등
  - 해결 방안: 시나리오 기반 기능 분석을 통한 충돌 여부 평가

## 3. 사이버-피지컬 시스템의 상호운용성 상충관계 분석 방안

본 논문에서는 상호운용성의 상충관계를 식별하기 위해 요구사항 분석 단계에서 기능적 요구사항과 함께 상호운용성의 상충관계를 함께 분석하는 방법을 제시한다. 또한, 요구사항 명세와 상호운용성의 상충관계를 명세 및 분석할 수 있도록 확장된 형태의 SysML 프로파일을 제시한다. 확장된 SysML은 그림 2와 같다.

- requirement: requirement에는 어플리케이션 및 시스템에 요구되는 기능적 요구사항을 작성한다. requirement의 ID는 requirement를 식별할 수 있는 유일한 속성의 식별자이고, 이를 통해 추적성을 확보할 수 있다. Text는 requirement의 설명을 자연어로 기술하는 부분이다. Source는 requirement가 요구되는 요소의 이름을 기술하고, Source Layer는 그림 1의 아키텍처에 대응되는 Source의 Layer를 기술한다. Target은 명령을 내리고자 하는 요소의 이름을

기술하고, Target Layer는 그림 1에 대응하는 Target의 Layer를 기술한다. Command는 Target에 요구되는 명령을 기술하는 곳으로써 간략하고 명료하게 작성한다. Decision resource는 Source가 Command를 생성하는데 사용한 resource를 기술한다.

- conflict requirement: conflict requirement는 두 개의 어플리케이션 및 시스템이 동일한 Target 시스템에 다른 명령을 내리는 경우 Target의 대처 및 대응 방안에 대한 요구사항이다. Target이 동일하고 Source와 Command가 동일하지 않은지를 분석하여 체계적으로 도출이 가능하다. 또한, Source가 동일하더라도 Decision resource가 다를 경우 역시 충돌 가능성이 있으므로 해당 requirement를 통해 대처 방안을 분석해야 한다. 대처 방안으로는 요구사항 우선순위를 통해 대처할 수 있고, 상황에 따라 우선순위를 동적으로 할당해 대처할 수 있다.

- failsafe requirement: Conflict requirement가 수행되기 어려울 경우 대안으로 만족시킬 추가적인 요구사항을 의미한다. 상호운용성의 상충이란 시스템이 설계 동안에 예상하지 못한 상황에 도달했다는 것을 의미한다. 실세계에는 다양한 상황이 발생하기 때문에 분석가가 모든 것을 식별하는데 한계가 있다. 경우에 따라서 상호운용성의 상충 상황은 Conflict requirement도 온전히 수행되기 어려운 상황일 가능성이 크다. 따라서 시스템 롤백 또는 초기화 등의 기능을 통해 시스템이 지속적으로 동작 가능하도록 기능을 분석할 필요가 있고, failsafe requirement는 이런 상황에 대한 시스템의 기능을 명세 한다.

그림 3은 제시한 방법 및 표기법을 바탕으로 실시간 돌발상황 감지 시스템 (AIDS, Automatic Incident Detection System)과 차로 제어 시스템 (LCS, Lane Control System)에 적용한 간략한 예제를 보여준다. 그림에서 AIDS 시스템의 요구사항 중 "사고 발생시 전광판에 진입 금지 출력" 요구사항과 LCS의 "고속도로 교통량에 따른 적절한 차선 유도" 요구사항이 전광판이라는 동일한 요소를 Target으로 삼는 것을 확인할 수 있다. 이 경우 Target에 상충된 Command가 발생할 수 있기 때문에, 이런 상충된 Command에 대해 어떻게 대처할 것인지 전략을 세울 필요가 있다. 예제에서는 우선순위를 설정하여 이를 해결하고자 하는 것을 확인할 수 있다. 이를 통해 기능적 요구사항뿐만 아니라 상충된 관계를 개발 초기인 요구사항 분석 단계에서 수행할 수 있고, 상충 가능한 부분을 체계적으로 식별해 적절한 대처 방안을 분석하는 것이 가능하다.

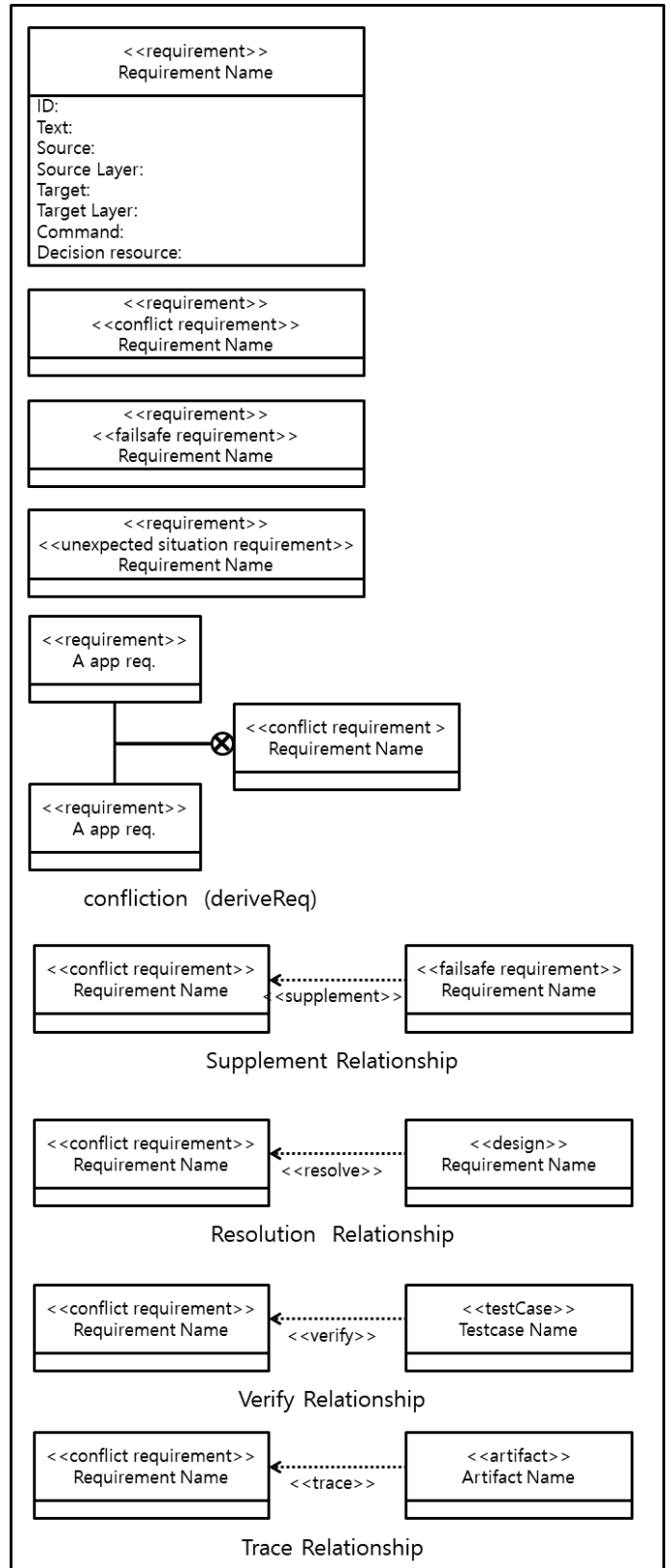


그림 2 상호운용성의 상충관계를 위한 확장된 SysML

제시한 방안의 문제점은 해당 분석을 요구사항 단계에서 수행하는 것이기 때문에 Target 또는 Command에 대한 명확한 식별 및 정의가 부족할 수 있다. 이 경우 개념적으로 정의를 하고 추후 명확한

요소가 정의되었을 때 적절하게 치환될 수 있도록, 사용한 용어 및 개념을 일관되게 유지하는 것이 중요하다. 또한, 모든 동일한 Target 마다 Conflict requirement를 작성하는 것은 지루하고 번거로운 일이 될 것이다. 이를 위해 앞으로 해당 분석을 자동/반자동으로 진행해 다이어그램을 생성해 주는 도구를 개발할 계획을 가지고 있다.

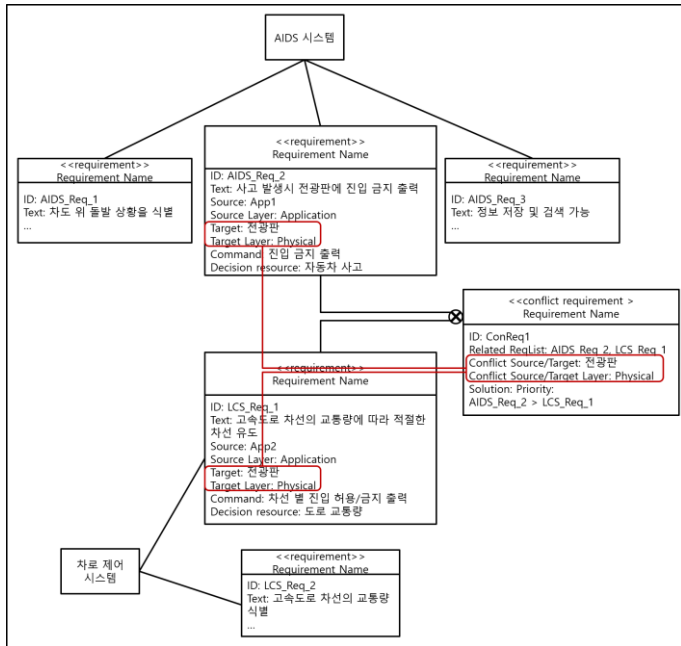


그림 3 상호운용성 상충관계를 기술한 요구사항 모델

#### 4. 결론

본 논문은 사이버-피지컬 시스템의 상호운용성의 상충관계를 요구사항 단계에서 분석할 수 있는 방안에 대해 제시한다. 요구사항 분석 단계에서 기능적 요구사항과 함께 상충관계에 대한 분석 및 명세를 수행할 수 있도록 SysML의 확장 형태를 제시하고, conflict requirement를 체계적으로 도출할 수 있는 방안에 대해서 제시했다. 설계의 이른 단계에서 해당 사항을 분석함으로써 개발 비용 절감 및 보다 안전하고 정확한 시스템의 개발이 가능할 것이라 기대할 수 있을 것이다. 추후, 해당 분석 모델을 바탕으로 자동으로 conflict requirement를 생성해주는 연구 및 도구 개발, 개발 사이클 전체를 통해서 해당 요구사항을 추적할 수 있는 방안에 대해서도 연구를 진행할 계획이다.

#### 사 사

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업(NRF-2017M3C4A7066479)의 지원을 받아 수행한 연구임.

#### 참고문헌

- [1] Griffor, Edward R., et al. "Framework for Cyber-Physical Systems: Volume 1, Overview," Special Publication (NIST SP)-1500-201. 2017.
- [2] Kit, Michal, et al. "An architecture framework for experimentations with self-adaptive cyber-physical systems." Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. IEEE Press, 2015.
- [3] IEEE, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, 1990.
- [4] Muccini, Henry, Mohammad Sharaf, and Danny Weyns. "Self-adaptation for cyber-physical systems: a systematic literature review." Proceedings of the 11th international symposium on software engineering for adaptive and self-managing systems. ACM, 2016.
- [5] SysML, OMG Systems Engineering Domain Special Interest Group, <http://syseng.omg.org/SysML.htm>
- [6] Status Report: Reference Architecture Model Industrie 4.0 (RAMI4.0), VDI/VDE Society Measurement and Automatic Control, July 2015
- [7] Official website of the AUTOSAR Partnership: [www.autosar.org](http://www.autosar.org)