



# COMMUNICATIONS

## OF SOFTWARE ENGINEERING SOCIETY

03/2025 VOL.5 NO.1

소프트웨어공학 소사이어티 소식



# Contents

## COMMUNICATIONS OF SOFTWARE ENGINEERING SOCIETY

### 소프트웨어공학 소사이어티 소식

- 03 게스트 편집자 인사말/이선아 (경상국립대학교)
- 04 기고문
  - 04 • 소프트웨어 제품라인 표준화 개관 : 이단형 (한국소프트웨어기술진흥협회)
  - 06 • 우리나라의 소프트웨어 제품라인 기술 연구 현황 : 강성원 (KAIST), 이지현 (전북대학교), 정필수 (경상국립대학교), 한영훈 (현대자동차)
- 14 신진연구자 소개
  - 이재권 (강원대학교)
  - 손정주 (경북대학교)
- 18 따끈한 논문
  - 18 • TSE 게재 논문, 다양한 언어모델을 활용한 오류 재현 : 강성민 (KAIST)
  - ASE 게재 논문, CodeLLM 을 위한 데이터 레시피 : 김기섭 (화웨이 홍콩 연구소)
- 22 STAARBugs 소개/최윤자 (경북대학교)
- 23 국내외 학술행사 소개
  - KCSE 2025 개최 후기 : 유준범 (건국대학교)
  - KCSE 2025 참가 후기 : 정호연 (서울과학기술대학교)
- 26 기관탐방
  - 산업 혁신과 사회 변화를 선도하는 디지털 엔지니어링 : 장정훈 (모아소프트)
  - University of Notre Dame : 이선아 (경상국립대학교)
- 30 소사이어티 광장
- 31 학회일정
- 32 기고문 및 소식모집
- 33 소사이어티 조직도
- 33 발행정보

01/2025  
VOL.5 NO.1



SOFTWARE  
ENGINEERING  
SOCIETY

◆ 게스트 편집자 인사말



**이 선 아**

경상국립대 교수

**안녕하세요. 2025년 3월 소사이어티공학 소사이어티 소식지의 게스트 편집을 맡게 된 이선아입니다.**

소프트웨어공학 분야에 관심을 갖게 된 것은 학부 당시 UML 수업을 들으면서부터입니다. 체계적인 개발을 가능하게 하는 소프트웨어공학 분야에 많은 흥미를 느꼈습니다. 그 이후 이화여자대학교 최병주 교수님의 연구실 석사, 삼성전자 재직 중 이단형 교수님이 주도하여 만드신 CMU 연계 재직자 석사, KAIST 강성원 교수님 연구실 박사로 졸업하였습니다. 또한 HKUST의 김성훈 교수님과 S. C. Cheung 교수님의 도움도 받았습니다. 많은 분들의 도움으로 소프트웨어공학 분야의 연구자가 될 수 있었습니다.

또한 지난 세월동안 소프트웨어공학 소사이어티가 한국의 소프트웨어공학의 연구와 교육의 중심에 있음을 보았고 감사드립니다. 소프트웨어공학 소사이어티 행사에 기존 학생으로 참여해서, 회사 재직 중 참여, 또한 현재 교수로서 참여하고 있습니다.

소프트웨어공학 분야는 소프트웨어의 체계적인 개발에 많은 공헌을 해 왔다고 생각합니다. 그런데 2016년부터 대중화되기 시작한 딥러닝과 2022년에 출시된 ChatGPT 등의 AI 시대에 소프트웨어공학은 어떤 모습이 되어야 할지에 대해서 아직은 잘 보이지 않습니다.

이러한 관점에서 이번 호는 오랫동안 연구가 진행된 소프트웨어 제품 라인과 새롭게 활발히 연구되는 LLM 관련 연구로 꾸며보았습니다.

소프트웨어 제품 라인으로는, KAIST 이단형 교수님의 표준 소개 원고와 KAIST 강성원 교수님, 전북대학교 이지현 교수님, 경상국립대 정필수 교수님, 현대자동차의 한영훈 박사님의 공동 집필 원고를 받을 수 있었습니다.

또한, 최신의 LLM 연구의 동향을 위해 KAIST 강성민 박사님, 화웨이 홍콩 연구소의 김기섭 박사님의 논문 소개 원고도 받을 수 있었습니다. 새로 임용된 강원대학교의 이재권 교수님과 경북대학교 손정주 교수님의 소개글도 감사 드립니다. 마지막으로 경북대학교의 최윤자 교수님이 보내주신 STAAR Bugs에 대한 소개글도 감사드립니다.

소프트웨어공학의 전통적인 기법의 발전과 최신 기술 트렌드의 검토와 모색을 통해 소프트웨어연구, 교육, 산업체 적용에 대한 소프트웨어공학 분야의 길을 찾아나갈 수 있기를 기대해 봅니다.

추가로, 2025년 2월에 KCSE 조직위원장으로 수고하신 유준범 교수님의 개최 후기와, 서울과기대 정호연 학생의 참가 후기로 준비하였습니다.

산업체로는 모아소프트의 장정훈 상무님으로부터 산업 혁신과 사회 변화를 선도하는 디지털 엔지니어링 원고를 받을 수 있었습니다.

마지막으로, 제가 2024년 연구년으로 다녀온 University of Notre Dame의 풍경과 Jane Cleland-Huang 교수님 연구실에 대한 소개도 간략히 넣었습니다.

2025년 3월 소사이어티공학 소사이어티 소식지가 소프트웨어공학을 사랑하시는 많은 분들께 도움이 되기를 희망합니다. 그리고 해당 소식지 발간을 위해 힘써주신 이주용 교수님, 지은경 교수님, 손정주 교수님 그리고 정우성 교수님 감사드립니다.



## 소프트웨어 제품라인 표준화 개관

한국소프트웨어기술진흥협회 이단형 회장

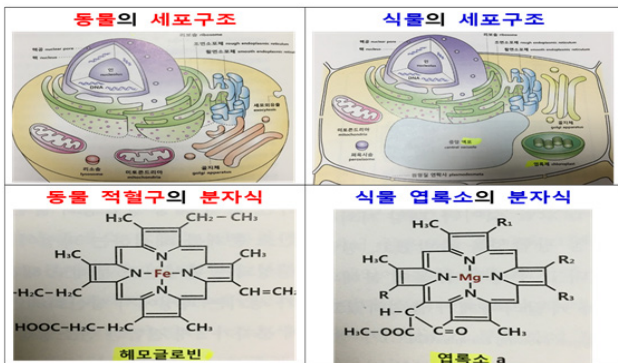
미국에서는 '제품라인'(Product Line)으로, EU에서는 '제품패밀리'(Product Family)로 명명되어 발전하여 왔으나, 본 고에서는 '제품군플랫폼'(Product Line Platform & Product Family Platform)으로 명명한다.

미국은 1975년부터 1995년까지 약 20년간 냉전시대에 러시아보다 무기체계에서 앞서기 위해 방위산업 업체인 록히드마틴(Lockheed Martin)과 보잉(Boeing)의 기술 주도하에 미국 육군, 공군, 해군의 무기체계 관련 조직이 협력해 STARS(Software Technology for Adaptable, Reliable Systems) 프로그램을 성공시키면서, '제품라인'의 토대를 마련했다.

유럽연합(EU)은 미국의 방위산업용 STARS 프로그램의 '제품라인'을 모방, 1995년부터 지금까지 자동차산업, 항공기산업, 의료기기산업, 통신기기산업, 가전산업 등 산업분야에 적용하고 있다. 이를 위해 EU는 회원국가 모두를 아우르는 ITEA(Information Technology for European Advancement) 프로그램에 14조원의 연구비와 연인원 5만 5000명 연구인력을 투입해오고 있다. 특히, EU는 자동차산업의 '제품패밀리'인 '오토사'(AUTOSAR: AUTomotive Open System Architecture)를 개발했다. '오토사'는 사양서(Spec)가 A4 용지로 5만 페이지에 이르며, 현재 세계 자동차 제조업체는 모두 오토사를 준수해서 자동차를 제조하는 수준으로 '제품패밀리'를 발전시켰다.

'제품군플랫폼'은 창조주(조물주)가 애용하는 매우 중요한 기반 기술이다. 그 이유는 다음과 같다: 지구 상에는 차별화되는 생명체의 종(동물과 식물) 약 1,000만 가지가 살아가고 있다. 그러나 이 모든 생명체들이 작동하는 주요 공통메커니즘(플랫폼)인 세포, 호흡기능(식물은 광합성) 및 에너지원인 포도당의 공급 메커니즘들이 유사하거나 동일하다. 아래의 그림은 동물과 식물의 세포구조, 호흡과 광합성, 적혈구와 엽록소의 공통성을 요약해서 보여준다.

- 지구상 1,000 만 가지 종류의 동물과 식물 세포구조는 모두 동일함
  - 동물의 호흡기능: 음식물 소화 → 포도당 + 산소 → 탄산가스 + 물 + 동물의 에너지
  - 식물의 광합성기능: 태양광 + 탄산가스 + 물 → 포도당 + 산소 + 식물의 성장
- (Source: "생명은 어떻게 작동하는가" 박문호, 2019)



'제품군'은 다수의 멤버제품(예: 현대자동차가 생산·판매하는 모든 차종)으로 구성된다. '제품군플랫폼'은 '제품군'에 속하는 모든 멤버제품이 공통으로 이용하기 위해서 재사용이 가능한 자산(예: 제조사양서, 제조 방법 및 절차, 제품을 구성하는 필수적 또는 선택적 항목, 등)을 플랫폼으로 체계화한 것을 의미한다. 즉, 개별 제품 각각의 요구사항에 맞추어, '공통부품'은 '제품군플랫폼'에 속해 있는 자산을 그대로 재사용하고, '가변부품'은 일부분을 수정해 재사용하거나, 멤버제품에 특화된 기능은 추가로 개발해서 사용한다. '제품군플랫폼'의 목적은 시장별·고객별 맞춤 제품을, 경쟁력 있는 가격과 품질로, 짧은 개발기간에 생산·판매하는 것이다. 시장과 기술, 그리고 고객취향 변화에 따라, '제품군플랫폼'은 지속적으로 진화하는 특징을 갖는다. '제품군플랫폼'은 보통 다수의 기술과 관련되고, 수습에서 수백 개 응용영역에서 재사용된다. 여기에서 다수의 기술은 가상화기술(Virtualization), 클라우드컴퓨팅(Cloud Computing), 사물인터넷(IoT), 빅데이터분석(Big Data Analytics), 인공지능(Artificial Intelligence), 다중코어(Multi Core), 보안기술(Security), 양자컴퓨팅(Quantum Computing) 등을 포함한다. 현재 대부분 선진국에서 화두가 되고 있는 '인더스트리(Industry) 4.0' 과 '제4차 산업혁명' 성공여부는 견고하면서도 유연성이 있는, '제품군플랫폼'을 활용할 수 있는 역량에 좌우(성공 기여도 90% 이상)된다고 하겠다. '제품군플랫폼'은 제품의 제조원가 중 소프트웨어 비중이 높은 자동차(52%), 항공기(51%), 정보기전(53.7%), 통신장비(52.7%), 의료장비(45.4%) 분야에서 제조업의 전장화, 제품의 지능화, 첨단화를 기하며 기업과 국가의 글로벌 경쟁력(미국 Cummins 수소자동차의 경우 '제품군플랫폼'을 토대로 품질향상: 50%, 생산성향상: 200%를 달성했음)을 높이는데 쓰이고 있다. 2003년부터 현재까지 모든 산업분야에 적용되는 '제품군플랫폼' 세계표준을 미국·EU와 협력해서 한국이 주도해오고 있다. '제품군플랫폼'의 참조모델(IS 26550)을 시발점으로 해서 한국이 주관해서 제정해오고 있는 '제품군플랫폼' 관련 세계표준 24개는 다음과 같다.

- IS 26550: Reference model for product line engineering and management
- IS 26551: Tools and methods for product line requirements engineering
- IS 26552: Tools and methods for product line architecture design
- IS 26553: Tools and methods for product line realization
- IS 26554: Tools and methods for product line testing
- IS 26555: Tools and methods for product line technical



- management
- IS 26556: Tools and methods for product line organizational management
  - IS 26557: Tools and methods for variability mechanisms in SSPL
  - IS 26558: Tools and methods for variability modeling in SSPL
  - IS 26559: Tools and methods for variability traceability in SSPL
  - IS 26560: Tools and methods for product line product management
  - IS 26561: Tools and methods for product line technical probe
  - IS 26562: Tools and methods for product line transition management
  - IS 26563: Tools and methods for product line configuration management
  - IS 26564: Tools and methods for product line measurement
  - IS 26565: Tools and methods for product line maturity framework
  - IS 26566: Tools and methods for product line texture

- IS 26567: Tools and methods for product line verification and validation
- IS 26568: Tools and methods for product line asset management
- IS 26569: Tools and methods for the dev. env. of sw and sys. product line
- IS 26570: Guide for the application of sw and sys. product line standards
- IS 26580: Tools and methods for the feature-based approach to SSPL
- IS 26581: Tools and methods for config. mgt. of feature-based PLE
- IS 26582: Tools and methods for change mgt. of feature-based PLE

'제품군플랫폼'은 지난 50년간 미국과 EU에서 괄목할 수준으로 진화해 왔다. 또한, 현재 진행 중인 '인더스트리(Industry) 4.0' 과 '제4차 산업혁명' 시대 및 향후 20-30년 내에 확산될 양자컴퓨팅 시대에는 더욱 획기적으로 진화해 나갈 것이다. 향후, 어떤 국가가 글로벌 시장경쟁에서 우위를 점하고, 경제성장과 새로운 일자리 창출을 모색하기 위해서는, '제품군플랫폼' 표준을 활용해서 제품·서비스의 고객맞춤화·지능화·차별화를 기하면서, 여러 산업영역에 특화된 '제품군플랫폼'을 선도할 수 있는 역량을 높여야 할 것이다.



## 우리나라의 소프트웨어 제품라인 기술 연구 현황



KAIST 강성원 교수



전북대학교 이지현 교수



경상국립대학교 정필수 교수



현대자동차 한영훈 박사

### 1. 서론

소프트웨어 제품라인(Software Product Line, SPL)은 소프트웨어 제품군(群)의 각 제품을 독립적으로 개발하지 않고 사전에 파악한 공통부분에 제품들의 가변부분과 제품 특정한 부분들을 결합하여 제품군을 개발하는 방식으로, 중복 개발을 회피하여 최소의 비용으로 제품군을 개발할 수 있게 한다. Paul Clements는 SPL 기술이 "시스템의 개발비용이 1/10로 줄어든다, 결합의 수가 1/10로 줄어든다, 개발기간이 1/10로 줄이는 기술"이라고 말하며,<sup>1)</sup>

이 기술을 통해 제품군 개발에 극적인 비용절감효과를 얻을 수 있다는 것을 강조하였다. 특히 이 기술은 제품군의 유지보수를 각 제품별로 하지 않고, 제품군 전체에 대하여 일회적으로 하는 것으로 충분히 함으로써 유지보수 단계에 발생할 수 있는 "유지보수 악몽(maintenance nightmare)"을 막아줄 수 있는 기술이다. 이런 효과로 인하여, 제품라인 기술을 확보한 회사와 그렇지 못한 회사의 소프트웨어 제품 개발 경쟁력은 크게 다른 수준일 수 밖에 없고, 따라서 SPL은 회사의 소프트웨어 시장에서의 성패를 가를 수 있는 결정적인 소프트웨어 기술이라고 볼 수 있다.

기술의 이러한 중요성에도 불구하고 SPL 기술의 국내 보급은 미미한 수준이다.<sup>2)</sup> 이 논문에서는 이런 상황에서도 국내에서 활발히 연구가 진행되어 국제적 경쟁력을 갖는 세 개의 SPL 기술분야에 대하여 국제적인 연구 동향과 우리나라의 연구 동향을 비교 검토한다. 이 세 분야는

- SPL로의 마이그레이션
- SPL을 위한 가변성 모델링과 분석
- SPL 시험

으로 논문의 제2절~제4절에서 한 분야씩 소개한다.

### 2. SPL로의 마이그레이션

SPL로의 마이그레이션은 SPL 개발 방식을 사용하지 않던 개발 조직이 SPL 개발 방식으로 옮겨가는 프로세스를 말한다. SPL로의 마이그레이션을 위해서는

\*저자 순서는 가나다 순이며 저자들의 기여도는 동일함.

1) [Linden 07] page v.

2) 이 논문의 부록 A에는 지난 10년동안 국내의 학회와 저널에 우리말로 발표된 저술과 논문의 목록이 나온다.

리거시(legacy) 제품들로부터의 피쳐(feature) 식별은 물론, 소스코드를 포함한 제품 개발 산출물을 기반으로 한 제품들 간의 차이 분석을 통한 가변성 식별, 가변성 모델링, 재공학, 리팩토링 등이 필요하다.

클론-앤-오운(Clone-And-Own, CAO) 방식은 각 제품을 완전히 독립적으로 개발하지는 않지만 (공통성을 부분적으로 이용하여) 제품군을 손쉽게 개발하는 방법이다. 기존 소프트웨어 시스템과 유사한 새로운 제품을 개발할 때, 기능적으로 동일하거나 유사한 리거시 코드를 복제(copy)한 결과인 클론을 적절히 수정하여 새로운 제품을 만든다[Lapeña 16]. 여기서 클론은 기존 제품이나 하위 시스템을 복제한 후 새로운 제품을 개발하는 종류인 제품-간-클론(inter-products clone)을 말한다. CAO 방식은 다양한 제품 변종(variant)을 빠르게 개발할 수 있어 제품 개발에 많이 사용된다. 그러나 CAO 방식은 임시방편적 재사용이기 때문에 재사용한 부분에 대한 변경이 요구될 때 변경의 전파가 어렵고[Krueger 09, Dubinsky 13], 유지보수 및 재사용한 부분에 대한 추적이 어렵다[Dubinsky 13]. 이러한 문제점들을 해결하기 위해서는 CAO 방식을 사용하는 조직은 SPL로 마이그레이션 해야 한다.

이 장에서는 소스코드를 기반으로 SPL로의 마이그레이션을 수행한 최근의 국내외 연구를 소개한다.

#### 2.1 국외 연구 동향

CAO 기반 개발 방식의 문제를 해결하고자 한 많은 초기 연구들은 제품-간-클론 검출에 치중했다. SPL로 마이그레이션하기 위해서는 CAO 방식으로 개발되어 온 제품들의 소스코드로부터 효율적이고 정확한 제품-간-클론 검출이 중요한 과정이지만, 제품 개발 방식을 SPL로 전환하려면 SPL를 실현하는 가변성 추출, 가변성 모델링, 제품라인 플랫폼 구축, 자동화 지원까지 이어져야 한다. SPL로의 마이그레이션 관련 최근의 국외 연구로는 공통성 및 가변성 추출, 제품라인 아키텍처 구축, 제품라인 코드베이스 구축에 대한 연구가 있다.

SPL로의 마이그레이션을 위한 공통성 및 가변성 추출 연구로 [Schlie 20]은 클래스, 메소드, 문장과 같은 제품의 코드 요소들을 트리 모델(Pure::variants

국내 연구자에 의해 국외 저널과 국외 학회에서 발표되는 논문도 있지만 그 수는 많지 않고, 부록 A에 나오는 논문들을 볼 때에 SPL 기술의 국내 보급 수준은 미미하다고 볼 수 있을 것이다

도구의 variant model)로 변환한 후, 한 개의 제품으로 시작하여 제품군의 제품들을 하나씩 추가해 가며 클론 관계에 있는 한 쌍의 코드 요소들 간의 유사도를 계산하고 그 결과값에 따라 공통성인지 가변성인지를 결정한다. 이 방법은 유사도 범위에 따라 각 코드 요소들의 공통성, 선택 가변성(optional), 또는 택일 가변성(alternative) 여부를 결정하기 때문에 동일한 코드 요소가 아니어도 공통성으로 식별될 수 있고, 가변성에 대한 판단도 정확도가 낮다.

소스코드로부터 제품라인 아키텍처를 구축한 연구로는 [Lee 20][Lima 19]가 있다. [Lee 20]은 제품들의 소스코드를 클래스와 메소드의 코드 요소로 분해한 후 각 코드 요소들의 공통성 지수를 산출하여 공통 코드 요소인지 가변 코드 요소인지 결정한다. 반면, [Lima 19]는 Stan4)<sup>3)</sup>와 PlantUML Dependency<sup>4)</sup>를 이용하여 소스코드를 모델로 변경하고 제안한 측정식에 따라 공통 모델 요소인지 가변 모델 요소인지 결정하여 제품라인 아키텍처를 구축한다. 그러나 이 두 연구에서 구축하는 제품라인 아키텍처는 공통 컴포넌트와 가변점과 가변값을 구성하는 가변 컴포넌트들로 이루어진 아키텍처가 아니라, 아키텍처 구성 요소들을 가변점과 가변값으로 정의하지 않은 느슨하게 기술된 아키텍처여서 이로부터 제품 아키텍처를 자동 생성할 수 없다.

소스코드로부터 제품라인 코드베이스를 구축하는 최근 연구로는 [Fenske 17][Ghallab 23][Krieter 23]가 있다. [Fenske 17]은 Copy Paste Detector 도구를 이용하여 클론을 검출한 후 공통 및 가변 코드 요소를 리팩토링한다. 피쳐 기반 SPL 개발 지원 도구인 FeatureIDE<sup>5)</sup>는 이 연구의 클론 검출과 리팩토링 방법을 사용하여 이 작업을 지원하지만 코드베이스 구축을 위한 리팩토링 과정의 대부분은 도구 내에서 수작업으로 수행되어야 한다. [Ghallab 23]은 Mobioos Forge 도구를 소개하는데, 이 도구는 제품의 소스코드를 입력으로 하여 가변성 모델링, 피쳐와 코드 매핑을 지원한다. 이 도구는 피쳐 선택에 따라 제품을 자동 생성하는 단계까지 지원하지만 사용자가 가변성 모델링부터 코드 매핑까지 전부 수작업으로 진행해야 한다. [Krieter 23]는 리거시 제품 개발 과정의 변경 이력인 포크(fork)와 브랜치(branch)를 통합한 커밋 그래프(commit graph)를 사용하여 SPL 코드베이스를 구축한다. 이 방법은 제품들을 이미 버전 관리 시스템에서 관리해온 경우에 적용할 수 있다.

[Schlie 20][Lee 20][Lima 19]는 제품들 간에 이루어진 클론 과정에 대한 정보 없이 먼저 추출식으로 마이그레이션을 위해 클론 코드를 식별한다. 반면, [Ghallab 23]과 버전 관리 시스템을 기반으로 하는 [Krieter 23]는 피쳐 정보를 기반으로 하향식으로 진행하는 클론 검출 과정이 불필요하다. 제품 간 클론을 검출하고 이를 토대로 가변성을 결정하는 것이 정확하지 않을 수 있고 복잡도가 높은 과정일 뿐 아니라 이후 과정을 자동화하기 어렵기 때문에, 국외 연구는 [Krieter 23]과 같이 버전 관리 시스템을 이용하여 SPL을 지원하는 방향으로 연구를 진행하고 있다.

3) <http://stan4j.com/>  
4) <https://plantuml-depend.sourceforge.net/>  
5) <http://fosd.net/fide>

## 2.2 국내 연구 동향

SPL로의 마이그레이션에 대한 국내 연구로 CAO 방식으로 개발된 제품군을 대상으로 한 연구인 [Lee 17][Kim 23a][Kim 23b]가 있다.

[Lee 17]은 공통성 및 가변성 추출 후 피쳐 모델을 구축하고 피쳐 모델과 일관되도록 리거시 시스템의 아키텍처를 수정하고 컴포넌트를 리팩토링한다. 이 연구는 소스코드로부터 클론을 검출하고 공통성 및 가변성을 추출하기 위해서 제품 쌍을 구성한 후 코드 차이를 분석하고, 이후 제품을 하나씩 추가해 가며 제품-간-클론을 검출한다. 이 과정에 정의된 규칙에 따라 가변점과 가변값을 결정하고 피쳐 모델을 구축한다. 이후 리팩토링 과정에서 아키텍처와 컴포넌트에 가변점을 넣는다. 이 연구는 구축한 제품라인 아키텍처로부터 원하는 제품의 아키텍처를 자동생성할 수 있지만, 마이그레이션 과정의 대부분이 수동으로 이루어지고 가변성 추출 시 제품 쌍으로 비교하기 때문에 제품 쌍 구성 순서에 따라 최종 결과가 달라질 수 있다는 문제점이 있다.

[Kim 23a][Kim 23b]의 연구는 [Lee 17]과 마찬가지로 추상 구문 트리(abstract syntax tree)를 기반으로 소스코드를 비교하지만, 제품 쌍의 비교가 아니라 전체 제품들을 한번에 비교하여 비교 순서가 최종 클론 검출 결과에 영향을 미치지 않도록 했다. 이 연구는 공통 코드와 가변 코드를 분리한 후 가변점과 가변값을 삽입하면서 전체 코드를 통합하는 방식으로 가변성 모델링 과정 없이 제품라인의 코드베이스를 구축한다. 특히, 이 연구는 소스코드를 입력으로 하여 제품라인 코드베이스를 구축하는 전체 과정을 자동화했으며 제품라인 코드베이스로부터 원하는 제품을 오류없이 생성할 수 있다.

국내에서는 임시방편적 재사용으로 제품군 개발을 하여 유지보수 악몽을 겪는 기업들이 이 문제에 대한 해결 방법을 모색하고 있다. 이러한 요구는 특히 자동차 제어 소프트웨어 분야에서 제기되고 있다. 일 예로 연료전지시스템 제어 소프트웨어는 승용차, 상용차 및 특수차량 뿐만 아니라 기차, 선박, 항공 등 모빌리티 분야 전반에 걸쳐 요구된다. 승용차 내에서도 옵션에 따라 다양한 변종들이 존재하는 것을 감안한다면, 연료전지시스템 제어 소프트웨어는 수많은 변종들을 동시에 대응해야 한다. 현재는 이를 단일 소프트웨어로 대응하고 있지만 변종의 수가 사용자의 요구에 따라 지속적으로 증가하고 있기 때문에 이를 해결하기 위한 대안으로 SPL개발 기술이 필요한 상황이다. SPL개발 기술을 도입하기 위해 단일 소프트웨어 개발 기술력을 고도화시키고 있다. V-model 기반 하에 소프트웨어 아키텍처 중심의 개발을 진행하고 있으며, 아키텍처 분석 도구인 Lattix도 활용하고 있다. 이러한 아키텍처 기술들은 SPL기술들을 도입하는데 기반기술로 활용될 예정이다.

## 2.3 국외 국내 연구의 비교

국외 연구는 리거시 제품들로부터의 가변성 추출, 가변성 모델링, 아키텍처 재구축, 코드베이스 구축을 위한 기반 기술 연구를 포함한 다양한 요소 기술들이 다양한 연구자에 의해 연구되었다. 그러나 국외 연구는 기업이 획득한 입력으로부터 SPL 플랫폼으로 제품을 생성하는 전과정을 위한 기술로 개발되지 않고, 특정 단계의 중요한 요소 기술에 집중하고 있다는 점에서



한계가 있다. 이 요소 기술들은 주어진 가정과 제약에 맞는 특정한 상황에서는 잘 동작하지만, SPL로의 마이그레이션에 실제로 적용되기 위해서는 먼저 요소 기술들이 일반화되고 상호 결합되어야 한다. 국외 연구에 비해, 국내의 SPL로의 마이그레이션 연구는 소수의 연구자 그룹에 의해 수행되었고 요소 기술들의 세밀함이 국외 연구에 비해 낮으나 SPL 플랫폼 구축부터 제품 생성 단계에 이르는 이음새 없는 기술을 개발하였다.

SPL로의 마이그레이션 향후 연구로 국내의 연구를 상호 보완해 적용하면서 제약조건과 가정을 일반적 상황으로 완화시킬 필요가 있다. 국내에서는 임시방편적 재사용으로 제품군 개발을 하여 유지보수 악몽을 겪는 기업들이 이 문제에 대한 해결 방법을 모색하고 있고 이러한 요구는 특히 자동차 제어 소프트웨어 분야에서 제기되고 있다. 따라서 소스코드 수준뿐만 아니라 모델 수준에서의 클론 검출 및 제품라인 플랫폼 구축을 위한 연구가 진행될 수 있을 것으로 전망한다. 더욱이 이들 모델이 버전 관리 시스템에 관리되고 있어 국외 연구들의 방향과 유사한 방향을 택할 수 있다.

### 3. SPL을 위한 가변성 모델링과 분석

이 절에서는 SPL 개발을 위한 가변성 모델링과 가변성 모델의 분석에 대한 국내외 연구 동향을 소개한다.

#### 3.1 국외 연구 동향

가변성 모델링은 SPL을 개발하기 위하여 제품들 간의 공통성과 가변성을 체계적으로 분석하는 활동이다. 가변성을 모델링 하기 위하여, 지금까지 피처 모델[Kang 90][Kang 02][Kang 13]과 Orthogonal Variability Model(OVM)[Pohl 05]을 포함하여 다양한 모델링 방법들이 제안되었다. 이러한 방법들 중에서 공통성과 가변성을 모두 다루는 피처 모델이 가장 활발하게 연구되고 있다. 따라서, 이 절에서는 가변성 모델링 연구 동향의 논의 범위를 피처 모델로 한정한다.

FODA 피처 모델[Kang 90]이 1990년에 최초로 제안된 이후, 이를 개선한 다양한 피처 모델이 제안되었다. 대표적인 피처모델로 FORM[Kang 02]과 Extended-FORM[Kang 13]이 있다. 피처 모델을 기반으로 지금까지 여러 도구들이 개발되었는데, 대표적인 도구의 하나로 Pure::variants<sup>6)</sup>는 피처 모델 기반의 소프트웨어 자산(asset)을 재사용 관리하기 위한 PTC사의 상용 도구이다. 제품군 모델(family model)과 제품 모델(variant model)을 활용하여 제품들 간의 공통성과 가변성 모델링을 지원한다. 또 다른 도구로 FeatureIDE<sup>7)</sup>는 Eclipse기반의 피처지향 소프트웨어 개발(feature-oriented software development)을 지원하는 오픈소스 프레임워크로 피처 모델링을 지원한다. 그 외에 웹기반의 피처 모델링 도구인 SPLOT<sup>8)</sup>이 있다.

피처 모델은 이와 같이 많은 연구자들에 의하여 연구되어 다양하게 진화하였음에도 불구하고, SPL 개발 전 단계에 걸쳐 활용하기에는 다양한 문제점과 한계점을 가지고 있다. [Kang 17]에 따르면, 기존의 전통적인 피처

모델은 (1) 과학적 원리의 미흡, (2) 추적성 문제, (3) 확장성 문제, (4) 복잡성 문제, (5) 가변성 해소 정의의 메커니즘의 부재 및 (6) 피처 그룹핑 메커니즘의 부재라는 문제점을 지적하였다. 피처 수준의 가변성 모델링뿐만 아니라 소프트웨어 각 개발 단계에서 발생하는 가변성을 체계적으로 다루기 위해서는 위의 문제점과 한계점들이 해결된 새로운 가변성 모델링 방법이 필요하다.

가변성 모델의 분석은 가변성 모델링을 통해 만들어진 가변성 모델이 잘 만들어졌는지 다양한 분석관점으로 검증하는 활동이다. 지금까지 제안된 가변성 모델에 대한 분석 연구들의 대부분은 피처 모델 분석에 관한 연구들이다. 그 이유는 가변성 모델에 대한 연구들 중에서 피처 모델이 차지하지 연구들이 지배적으로 많을 뿐만 아니라 OVM과 비교할 때 피처 모델에는 공통적인 부분이 포함되어 있어 모델이 더 복잡하기 때문이다.

이런 이유로 지금까지 다양한 분석 관점들이 개발되었는데, [Benavides 10]의 연구에 따르면 1990년부터 2009년까지 약 30개의 분석 관점들이 제시되었다. 이러한 분석 관점들은 크게 두 분류로 구분된다. (1) 올바른 모델이 만들어졌는지 검증하는 분석관점들과 (2) 올바른 모델로부터 산출되는 통계적 분석 관점(예, 피처모델로부터 구성할 수 있는 잠재적인 제품의 수)들이다. 이 연구에 따르면, 올바른 모델이 만들어졌는지 검증하는 분석 관점에 대한 연구들이 통계적 분석 관점에 대한 연구들보다 더 활발하게 연구되고 있다. 그 이유는 통계적 분석들은 올바른 모델이 만들어졌을 때 수행할 수 있기 때문이다.

[Maßen 04] 연구에서는 이러한 분석 관점들의 중요도를 분류한다. [Maßen 04] 연구에 따르면, 이러한 다양한 분석 관점들 중에 모순(inconsistency) 분석이 가장 중요한 분석으로 분류한다. 그 이유는 피처 모델에 모순이 있으면, 이 모델로부터 제품 구성이 불가능하기 때문에 반드시 해결해야 하는 문제이기 때문이다. 이에 반하여 Dead feature 분석과 False optional 분석 등과 같은 변칙들(anomalies)은 제품 구성이 가능한 모델을 대상으로 분석을 수행하기 때문에 모순 분석 보다 낮은 중요도로 분류하였다.

피처 모델에서 모순 분석을 수행하기 위해서는 먼저 피처 모델을 분석하기 쉬운 형태로 변환해야 한다. 대표적인 변환 방법으로 (1) 명제 논리(Propositional Logic) 기반의 분석, (2) Constraint Satisfaction Problem (CSP) 기반의 분석, (3) Description Logic 기반의 분석, (4) First-Order Logic 기반의 분석 및 (5) Dynamic-priority 기반의 분석들이 제안되었다. 이 중에서 명제 논리 기반의 분석 방법에 대한 연구들이 가장 활발하다 그 이유는 명제 논리 기반의 분석들의 실용적인 측면을 입증하였기 때문이다. 명제 논리 기반의 분석은 피처 모델을 명제 논리식으로 변환한 뒤 SAT Solver를 통해 분석한다. SAT Solver의 평가결과가 참이면, 피처 모델로부터 1개의 이상의 제품을 구성할 수 있다는 의미로 피처 모델에 모순이 없다는 뜻이다. 반면에 평가결과가 거짓이면, 피처 모델로부터 제품을 구성할 수 없다는 의미로 피처 모델에 모순이 있다는 뜻이다.

피처 모델을 실세계(real-world)에서 실용적으로 활용하게 위해서는 피처 모델의 규모가 증가하여도 효율적인 분석이 가능해야 한다. [Mendonca 09]과 [Liang 15]의 연구들은 명제 논리 기반의 분석의 실용적인 측면을 다양한 실험을 통하여 입증하였다. [Mendonca 09]과 [Liang 15]의 연구에 따르면,

6) <https://www.pure-systems.com/purevariants>.

7) <https://featureide.github.io/>.

8) <http://www.splot-research.org/>

피쳐 모델을 명제 논리식으로 변환할 때 주로 2개의 literal을 가진 clause들이 만들어지기 때문에 명제 논리식의 규모가 커지더라도 SAT Solver로 분석하기에 매우 효율적이라는 것을 다양한 실험을 통해 입증하였다.

지금까지 제안된 연구들을 통해서 피쳐 모델에 모순의 발생 여부는 분석 가능하지만, 모순이 발생한 원인을 정확하게 찾는 연구는 거의 없다. 일부 연구들은 피쳐 모델의 모순 분석 결과에 대하여 “설명(explanation)” 개념을 만들어서 원인으로써 제공하지만, “설명”은 (1) 피쳐 모델의 규모가 크거나 혹은 (2) 모순의 발생 지점이 여러 곳인 경우에는 원인의 범위를 정확하게 찾지 못하는 한계점이 있다. 현재 수준보다 피쳐 모델을 더 실용적으로 활용하여 위하여, 모순의 발생 여부뿐 아니라 그 원인을 정확하게 검출하여 연구가 필요하다.

### 3.2 국내 연구

가변성 모델링에 대한 국내 연구들은 국외 연구와 비교하여 그 수는 현저히 적지만 지속적으로 수행되어 왔다. [Kang 17]에서 지적인 피쳐 모델의 문제점과 한계점을 해결하기 위하여, [강성원 18]과 [한영훈 18]의 연구들이 나왔다. [강성원 18]는 기존의 피쳐모델들이 여러 차례 진화하였음에도 불구하고 여전히 해결되지 않은 여러 문제점들을 해결하거나 개선한 Compositional Feature Model(CFM) 방법을 소개했다. CFM은 [강성원 17]에서 소개한 가변성 모델링 메커니즘에 기초하여 산출물의 가변성을 피쳐수준에서 도형적으로 표현하는 것이 가능하다. 또한, CFM은 가변성 모델을 정의하는 방법이며 이는 그 자체가 하나의 모델이 아니라 모델을 인스턴스로 가질 수 있도록 해준다[강성원 18]. [한영훈 18]의 연구에서는 CFM 방법을 기반으로 Systematic Variability Modeler(SVM) 도구를 개발하였다.

가변성 모델 분석에 대한 국내 연구로 [한영훈 19]에서는 [강성원 18]에서 개발된 SVM 가변성 모델을 대상으로 국내 최초로 모순 분석 기술을 개발하여 적용하였다. [한영훈 22]는 SVM 가변성 모델에 모순이 있는지 분석하는 것과 동시에 모순의 원인을 정확하게 검출하는 방법과 도구를 제안하였다. 특히, 모순의 원인을 정확하게 검출하는 방법은 기존의 피쳐 모델의 문제점과 한계점을 개선한 SVM 가변성 모델에서 적용 가능하다는 점과 모순이 발생한 원인의 위치를 정확하게 찾는 점에서 국외 연구보다 앞섰다고 볼 수 있다.

### 3.3 국외 국내 연구의 비교

가변성 모델링 방법에 대한 국외의 연구는 대부분 피쳐 모델에 관하여 수행되었다. [Kang 17]에서 피쳐 모델의 문제점과 한계점이 발표된 이후에도 후에도 국외 연구에서 새로운 피쳐 모델링 방법을 제시되지 않았다. 반면에 국내 연구에서는 피쳐 모델의 문제점과 한계점을 해결하는 연구로 [강성원 18]에서 CFM 모델링 방법이 개발되었고, 이 방법을 기반으로 [한영훈 18]에서 SVM도구가 개발되었다.

가변성 모델의 분석에 대한 국외 연구들은 주로 피쳐 모델에 대한 분석관점들을 정의하는 연구, 피쳐 모델 분석하기 쉬운 형태로 변환하는 연구 등을 포함하는 피쳐 모델에 대한 분석이었다. 이러한 연구들을 기반으로 피쳐 모델에서 다양한 분석 관점들을 효율적으로 분석하는 것이 가능하였다. 그러나,

분석된 결과에 대한 원인을 자동으로 찾는 연구들은 매우 드물다. 특히, 가장 중요하다고 분류된 모순 분석으로 피쳐 모델에 모순이 있는지 여부는 분석하는 것이 가능하였지만, 그 원인을 찾는 연구들은 거의 없었다. [한영훈 22]에서는 모순 분석 수준을 고도화하였는데, 구체적으로 SVM 가변성 모델에 모순이 있는지 분석하고, 모순이 있는 경우 그 원인을 정확하게 찾아준다.

## 4. SPL 시험

SPL 시험의 목표는 소프트웨어 제품군의 시험 자산을 체계적으로 재사용하여 반복적인 시험을 최소화하는 것이다. 예를 들어, 세 제품에 공통으로 포함된 컴포넌트를 각 제품에서 반복 시험하는 것은 비효율적이다. 대신, 해당 컴포넌트를 한 번만 시험하고 그 결과를 세 제품에 재사용하면 시험 비용을 크게 절감할 수 있다. 그러나 소프트웨어 제품군은 단일 소프트웨어보다 규모가 크고 복잡도가 높기 때문에 보다 체계적이고 정교한 시험 기술이 필요하다. 본 절에서는 SPL 시험 기술의 국내외 연구 동향을 살펴본다.

### 4.1 국외 연구 동향

SPL은 잠재적으로 매우 많은 수의 제품이 생성 가능하다. 이러한 특성으로, 최소한의 시험 노력으로 최대의 시험 효과를 달성하기 위한 접근 방법들이 제안되었다. 이와 관련하여, 이 절에서는 SPL 시험 연구를 점진적 시험, 조합 상호작용 시험, 메타 휴리스틱 시험, 시험 자동화로 나누어 소개한다.

과거에는 시험 자산의 체계적인 재사용을 위해 공통 시험 자산(common test assets)을 개발하고 이를 관련 제품들에 재사용하는 기술들이 연구되었다. 그러나 이런 접근에서는 공통 시험 자산의 개발과 유지 비용이 점차 증가하는 단점을 보였다. 이에 따라 최근에 제품군 내 각 제품을 순차적으로 고려하고, 이전 제품들과의 차이(delta)만을 시험하는 델타 기반 점진적 접근법(delta-oriented incremental approach)이 제안되었다[Lachmann 15][Al-Hajjaji 17]. 제품군 내 두 제품 간의 델타를 분석하기 위해 단일 제품의 두 버전 간 델타를 분석하는 방법을 적용할 수 있다는 점에 착안하여, 델타 기반 점진적 접근법의 최신 기술들은 단일 제품에 대한 회귀 시험 기술을 적용한다[Lity 19].

SPL을 통해 수십 또는 수백 개의 제품을 생산할 경우 이를 모두 시험하는 데는 막대한 비용이 들 수 있다. 시험 비용을 최소화하기 위해, 적은 수의 제품을 샘플링하면서 시험 효과를 극대화할 수 있는 기술로 조합 상호작용 시험(combinatorial interaction testing)이 있다[Lopez-Herrejon 15]. 이는 n개의 피쳐 조합을 시험하는 방식이며, 두 피쳐 간 상호작용(n=2)을 대상으로 하는 쌍대 시험(pairwise testing)이 대표적이다. 예를 들어, Windows 또는 Linux 운영체제, IE 또는 Chrome 브라우저, Oracle 또는 JDBC 데이터베이스를 지원하는 제품군이 있다고 가정할 때, 총 8가지 운영체제-브라우저-데이터베이스 조합이 가능하지만, 아래 그림 1과 같이 6개의 제품만으로도 모든 두 피쳐 간 상호작용을 시험할 수 있다.제한된 비용으로 시험 효과를 극대화하기 위해 제품군 내에서 최적의 제품 집합을 선택하는 메타 휴리스틱 알고리즘들이 제안되었다[Wang 14][Ferreira 17]. 이 알고리즘들은 시험 비용, 선택된 제품 수, 제품 간 유사도 등을 최소화하고, 결함 탐지율, 커버리지, 뮤테이션 스코어

OS	Browser	DB
Windows	IE	Oracle
Windows	IE	JDBC
Windows	Chrome	Oracle
Windows	Chrome	JDBC
Linux	IE	Oracle
Linux	IE	JDBC
Linux	Chrome	Oracle
Linux	Chrome	JDBC



OS	Browser	DB
Windows	IE	Oracle
Windows	IE	JDBC
Windows	Chrome	Oracle
Windows	Chrome	JDBC
Linux	IE	JDBC
Linux	Chrome	Oracle

그림 1. 쌍대 시험을 위한 제품 샘플링

등을 최대화하는 다목적 최적화 알고리즘(multi-objective optimization algorithms)을 활용한다. 또한, 신속한 결함 탐지를 위해 선택된 제품들의 시험 우선순위를 결정하는 연구도 진행되었다. 이 과정에서는 시험 시간, 커버리지, 제품 간 유사도, 결함 발생 빈도 등을 최적화하는 다목적 최적화 알고리즘이 사용되었다.

마지막으로, SPL 시험의 높은 비용을 절감하기 위해서는 자동화 도구의 개발이 필수적이다. 하지만 최근 조사[Agh 24]에 따르면, 2003년부터 2023년까지 게재된 주요 SPL 시험 논문 118편 중 49편에서만 제안된 방법에 대한 자동화 도구를 개발했다고 언급하였고, 그 중 공개된 도구는 19개(16.1%)에 불과하다. 앞으로 SPL 시험 기술이 실무에 정착하기 위해서는 자동화 도구 개발이 더욱 활발히 이루어져야 하며, 이러한 도구들을 공개하여 현장 전문가들이 쉽게 접근할 수 있게 해야 할 것이다.

## 4.2 국내 연구

현재까지 국내외에서 개발된 SPL 시험 기술들은 요구사항, 아키텍처, 가변성 모델, 추적성 정보 등 다양한 개발 산출물을 필요로 한다. 그러나 이러한 산출물을 철저히 관리하지 않는 조직에서는 해당 기술을 활용하기 어렵다. 또한, 전문가의 개입을 요구하여 자동화가 어렵다는 한계가 있다. 이러한 문제를 해결하기 위해, 국내에서 코드 기반 SPL 시험 기술이 연구되었다[Jung 19][Jung 20][Jung 22][Jung 24]. 그 밖에도, SPL의 불필요한 시험 비용을 줄이는 경로 기반(path-based) 통합 시험 기술이 연구되었다[Lee 19][Lee 23]. 이 절에서는 코드 기반 SPL 시험 연구[Jung 19][Jung 20][Jung 22][Jung 24]에 대해 소개한다.

제안된 기술은 여러 제품에서 동일한 커버리지와 시험 결과를 생성하는 동치 실행(equivalent execution)을 미리 검출하여 시험 항목의 중복 실행을 피하는 방법이다. 비결정적 실행(non-deterministic execution)이 발생하지 않는 가정 하에, 제품 1에 적용된 시험 항목이 실행하는 모든 코드가 제품 2에 포함하면, 마치 하나의 시험 항목을 동일한 제품에서 두 번 실행하듯 제품 2에서 동치 실행된다. 이러한 원리를 기반으로 제안된 기술은 동치 실행 여부를 판단한다. 코드 간 비교에는 해시값을 사용한다.

그림 2는 코드 기반 SPL 시험 기술의 핵심 아이디어를 보여준다. 클래스 A-G로 구성된 세 제품 P1={A, B, C, D, E}, P2={A, B, C, D, E, F}, P3={B, G}에 대해,

시험 항목 t1, t2, t3가 각각 <A, B, C, D>, <G, B>, <E>의 실행 트레이스를 갖는다고 하자. 이 경우 t1은 클래스 A-D를 포함하는 제품들(P1, P2)에서, t2는 클래스 G, B를 갖는 제품(P3)에서, t3는 클래스 E를 갖는 제품들(P1, P2, P3)에서 동치 실행을 발생시킨다. 따라서 t1, t2, t3는 각 해당 제품들에서 한 번만 실행하면 되므로, 중복 실행을 피할 수 있다. 제안된 기술은 제품 코드의 해시값 추출, 시험 항목의 실행 트레이스 추출, 그리고 동치 실행 여부 판단으로 구성되며, 이 모든 과정은 자동화가 가능하다. 따라서 이 기술을 활용하면 SPL 시험을 코드 기반으로 자동화할 수 있어, 시험 비용을 획기적으로 절감할 수 있다.

## 4.3 국외 국내 연구의 비교

Boeing, GmbH, General Motors, Lockheed Martin 등의 국외 선도 기업들은 SPL에 대한 깊은 이해와 높은 기술 수요를 바탕으로 학계와 협력하여 실증적인 연구 결과를 도출하였다[Fischer 23][Flores 12][Gregg 19]. 이들은 나아가 제안된 기술을 실제 산업체 프로젝트에 적용하여 그 효용성을 성공적으로 입증하였다.

한편, 국내에서는 SPL 연구·개발에 적극 참여하는 산업체가 드물고 대부분의 기술 검증이 모의 실험에 의존하고 있다. 그러나 실제 산업 환경에서의 검증이 이루어지지 않으면 연구 결과의 실용성과 신뢰성을 확보하기 어려우며, 이러한 취약한 실험 환경은 국내에서의 SPL 기술 발전을 저해하는 주된 요인이다. 이러한 문제를 해결하기 위해서는 적극적인 산학 협력을 통해 연구자들이 산업체의 실질적인 요구를 반영하고 이를 바탕으로 실용적이고 실증적인 연구를 수행해야 한다.

## 5. 결론

SPL의 핵심 기술인 제품군의 공통성과 가변성의 식별과 그에 기반한 제품라인 플랫폼 모델의 구축은 제품군을 위한 아키텍처 설계에 해당되는 활동이다. 그리고 SPL을 적용한 제품군 개발을 위해서는 기존의 방식에서 SPL 개발방식으로 효과적으로 옮겨가야 하며 또한 재사용이 극대화된 SPL 시험 기술은 제품의 품질 보장에 필요하기 때문에, 이 기술들은 SPL의 중요한 기본 기술이다. 이 논문에서 볼 수 있듯이 SPL의 중요한 이 세 측면에서 우수한 기술이 국내에서 연구 개발되었다. 그러나 아직까지는 산학 협력이 활발하지 않아 실증적인 연구가 부족한 상황이며, 이 문제를 해결하기 위해서는 국내 SPL



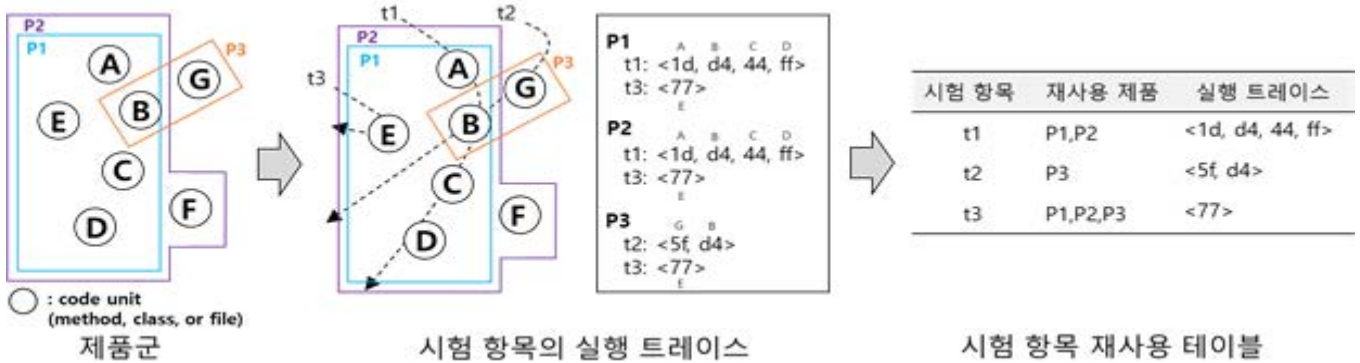


그림 2. 시험 항목의 동치 실행 검출

수요 기업들과 연구자들이 긴밀히 협력하여 연구하고 기술 개발을 해야 한다.

국내의 여러 글로벌 IT 기업이 체계적인 소프트웨어 아키텍트 양성 프로그램을 가지고 있을 정도로, 소프트웨어 아키텍처 기술은 국내에서 그 중요성이 널리 인식되어 있으며 우리나라가 세계적으로도 앞선 기술 수준과 인력 공급 기반을 갖고 있다. 그러나 이 논문에서 소개된 SPL 기술들은 제품군 개발에 반드시 필요한 아키텍처 기술이고 제품군 개발을 책임진 소프트웨어 아키텍트가 반드시 갖추어야 할 기술임에도, 국내에 이 기술을 갖춘 소프트웨어 아키텍트는 드물다. SPL 기술 없이 제품군을 생산하는 회사는 사업이 잘 되어 많은 고객에게 제품이 인도될 때 오히려 "유지보수 악몽"으로 인해 큰 어려움을 겪게 된다. 급격히 변화 발전하는 소프트웨어 시장에서 성공하고자 하는 기업은 SPL 기술을 조기에 확보해야 하고 이를 위해 산학 협력에도 적극적으로 노력해야 할 것이다.

### 참고문헌

[Agh 24] Agh, H., Azamnouri, A., & Wagner, S., "Software product line testing: a systematic literature review," *Empirical Software Engineering*, 29(6), 146, 2024.

[Al-Hajjaji 17] Al-Hajjaji, M., Lity, S., Lachmann, R., Thüm, T., Schaefer, I., & Saake, G., "Delta-oriented product prioritization for similarity-based product-line testing," *IEEE/ACM 2nd International Workshop on Variability and Complexity in Software Design (VACE)*, pp. 34-40, 2017.

[Benavides 10] D. Benavides, S. Segura, and A. Ruiz-Cortés, "Automated analysis of feature models 20 years later: a literature review," *Information Systems*, 35(6), pp.615-636, 2010.

[Dubinsky 13] Y. Dubinsky, J. Rubin, T. Berger, S. Duszynski, M. Becker, and K. Czarnecki, "An exploratory study of cloning in industrial software product lines," *Proc. of the 17th European Conference on Software Maintenance and Reengineering (CSMR)*, pp. 25-34, 2013.

[Fenske 17] W., J. Meinicke, Sandro Schulze, Steffen Schulze, and G. Saake, "Variant-preserving refactorings for migrating cloned products

to a product line," *Proc. of the IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 316-326, 2017.

[Ferreira 17] Ferreira, T. N., Lima, J. A. P., Strickler, A., Kuk, J. N., Vergilio, S. R., & Pozo, A., "Hyper-heuristic based product selection for software product line testing," *IEEE Computational Intelligence Magazine*, 12(2), pp. 34-45, 2017.

[Fischer 23] Fischer, S., Ramler, R., Assunção, W. K., Egyed, A., Gradl, C., & Auberger, S., "Model-based Testing for a Family of Mobile Applications: Industrial Experiences," *Proc. of the 27th ACM International Systems and Software Product Line Conference-Vol. A*, pp. 242-253, 2023.

[Flores 12] Flores, R., Krueger, C., & Clements, P., "Mega-scale product line engineering at general motors," *Proc. of the 16th International Software Product Line Conference-Vol. 1*, pp. 259-268, 2012.

[Ghallab 23] K. Ghallab, T. Ziadi and Z. Chalal, "Migrating individual applications into software product lines using the Mobioos Forge Platform," *Proc. of the 30th Asia-Pacific Software Engineering Conference (APSEC)*, Seoul, Korea, pp. 483-492, 2023.

[Gregg 19] Gregg, S. P., Scharadin, R., & Clements, P. C., "The best of both worlds: Agile development meets product line engineering at Lockheed Martin," *INSIGHT*, 22(2), 25-31, 2019.

[Jung 19] Jung, P., Kang, S., Lee, J., "Automated code-based test selection for software product line regression testing," *Journal of Systems and Software*, 158, 2019.

[Jung 20] Jung, P., Kang, S., Lee, J., "Efficient Regression Testing of Software Product Lines by Reducing Redundant Test Executions," *Applied Sciences*, 10(23), 8686, 2020.

[Jung 22] Jung, P., Kang, S., Lee, J., "Reducing Redundant Test Executions in Software Product Line Testing—A Case Study," *Electronics*, 11(7), 1165, 2022.

[Jung 24] Jung, P., Lee, S., & Lee, U., "Automated code-based test case reuse for software product line testing," *Information and Software Technology*, 166, 107372. 2024.

[Kang 90] Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., Peterson, A. S., "Feature-Oriented Domain Analysis (FODA) Feasibility Study," SEI Technical Report, 1990.

[Kang 02] Kang, K., Lee, J., Kim, Donohoe, P., "Feature oriented product line engineering," *IEEE Software*, Vo. 19, No. 4, pp. 58-65, 2002.

[Kang 13] Kang, K. C. and Lee, H., "Variability Modeling," *Systems and Software Variability Management*, Ed. R. Capilla et al., Springer-Verlag Berlin Heidelberg, 2013.

[Kang 17] S. Kang, Y. Han, H. Ahn, and J. Lee, "Critical analysis of feature model evolution," *The 8th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, November 24-26, 2017.

[Kim 23a] T. Kim and J. Lee, "A code clustering technique for unifying method full path of reusable cloned code sets of a product family", *KIPS Transactions on Software and Data Engineering*, Vol.12, No.1, pp.1~18, 2023.

[Kim 23b] T. Kim, "Product line code base construction for migration from the clone-and-own approach to software product line engineering", Master's thesis, Jeonbuk National University, 2023.

[Krieter 23] S. Krieter, J. Krüger, T. Leich, and G. Saake. "VariantInc: Automatically pruning and integrating versioned software variants," *Proc. of the 27th ACM International Systems and Software Product Line Conference - Vol. A (SPLC '23)*, pp. 129-140, 2023.

[Krueger 09] C.W. Krueger and K. Jackson, "Requirements engineering for systems and software product lines," IBM Corp., Somers, NY, USA, 2009.

[Lachmann 15] Lachmann, R., Lity, S., Lischke, S., Beddig, S., Schulze, S., & Schaefer, I., "Delta-oriented test case prioritization for integration testing of software product lines," *Proc. of the 19th International Conference on Software Product Line* pp. 81-90, 2015.

[Lapeña 16] R. Lapeña, M. Ballarin, and C. Cetina, "Towards clone-and-own support: Locating relevant methods in legacy products," *Proc. of the 20th International Systems and Software Product Line Conference*, pp. 194-203, 2016.

[Lee 17] H. Lee and K. B. Lee, "A feature-oriented method for extracting a product line asset from a family of legacy applications", *KIPS Tr. Software and Data Eng.* Vol.6, No.7, pp.337~352, 2017.

[Lee 19] Lee, J., & Hwang, S., "Path-based integration testing of a software

product line," *Big Data, Cloud Computing, Data Science & Engineering* 3, pp. 93-102, 2019.

[Lee 20] J. Lee, T. Kim, and S. Kang, "Recovering software product line architecture of product variants developed with the clone-and-own approach," *Proc. of the IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 985-990, 2020.

[Lee 23] Lee, J., "An Experimental Evaluation of Path-Based Product Line Integration Testing and Test Coverage Metrics. *Applied Sciences*," 13(23), 12670, 2023.

[Liang 15] J. Liang, V. Ganesh, K. Czarnecki, and V. Raman, "SAT-based Analysis of Large Real-world Feature Models is Easy," *Proc. of the 19th International Conference on Software Product Line*, pp.91-100, 2015.

[Lima 19] C. Lima, W. KG Assunção, J. Martinez, W. Mendonça, I. C Machado, and C. FG Chavez, "Product line architecture recovery with outlier filtering in software families: the Apo-Games case study," *Journal of the Brazilian Computer Society*, 25(7), pp. 1-17, 2019.

[Linden 07] van der Linden, F., Schmid, K., Rommes, E., *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*, Springer-Verlag, 2007.

[Lity 19] Lity, S., Nieke, M., Thüm, T., & Schaefer, I., "Retest test selection for product-line regression testing of variants and versions of variants," *Journal of Systems and Software*, 147, 46-63, 2019.

[Lopez-Herrejon 15] Lopez-Herrejon, R. E., Fischer, S., Ramler, R., and Egyed, A., "A first systematic mapping study on combinatorial interaction testing for software product lines," *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 1-10). IEEE, April 2015.

[Maßen 04] T. Maßen, and H. Lichter, "Deficiencies in feature models," *Workshop on software variability management for product derivation-towards tool support*, Vol. 44, 2004.

[Mendonca 09] M. Mendonca, A. Wąsowski, and K. Czarnecki, "SAT-based Analysis of Feature Models is Easy," *Proc. of the 13th International Software Product Line Conference*, pp.231-240, 2009.

[Pohl 05] K. Pohl, G. Böckle and F. Linden, *Software Product Line engineering: foundations, Principles and Techniques*, Heidelberg: Springer, 2005.

[Schlie 20] A. Schlie, S. Schulze, and I. Shaefer, "Recovering variability information from source code of clone-and-own software systems," *Proc. of the 14th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS)*, pp. 1-9, 2020.

[Wang 14] Wang, S., Buchmann, D., Ali, S., Gotlieb, A., Pradhan, D., & Liaaen, M., "Multi-objective test prioritization in software product line testing: An industrial case study," Proc. of the 18th International Software Product Line Conference-Volume 1, pp. 32-41, 2014.

#### 부록 A. 국내 출간 SPL 논문 및 서적(2014년 ~ 2024년)

[조경록 23] 조경록, 김재경, 이은서, "IoT 기능 시험에서 공용성·가변성 시험항목 추출을 위한 프레임워크 설계에 관한 연구," 제25회 한국 소프트웨어공학 학술대회(KCSE), 2023. (안동대학교)

[김태영 23] 김태영, 이지현, "클론앤아웃으로 개발된 제품군의 클론 코드 통합 도구," 제25회 한국 소프트웨어공학 학술대회(KCSE), 2023. (전북대학교)

[한수빈 23] 한수빈, 이지현, "소프트웨어 제품라인 시험 커버리지 측정 도구," 제25회 한국 소프트웨어공학 학술대회(KCSE), 2023. (전북대학교)

[구태완 23] 구태완, 김백준, 성병준, 박홍기, "소프트웨어 재공학 목표 기반 차량 제어 소프트웨어 재공학 사례 연구," 제25회 한국 소프트웨어공학 학술대회(KCSE), 2023. (현대자동차).

[한영훈 22] 한영훈, "소프트웨어 제품라인 개발을 위한 가변성 모델에서 모순의 원인 검출," 한국과학기술원 박사학위논문, 2022.

[한수빈 22] 한수빈, 이지현, 고서연, "구성가능한 소프트웨어 제품라인 시험에서 코드 커버리지 측정," 정보처리학회논문지, vol.11, no.7, 통권 118호 pp. 273-282, 2022. (전북대학교)

[강성원 21] 강성원, 이지현, 한영훈, 정필수, 소프트웨어 제품 라인 개발 입문, 도서출판홍릉, 2021. (한국과학기술원, 전북대학교)

[박삼준 21a] 박삼준, 노성규, 김도형, 이순주, 박병수, 이인섭, "항법소프트웨어 Software Product Line 프레임워크 운영개념," 정보처리학회논문지/소프트웨어 및 데이터 공학 제10권 제6호(2021. 6), 2021. (국방과학연구소)

[박삼준 21b] 박삼준, 노성규, 이관우, 박병수, 남성호, "관성항법소프트웨어 SPL(Software Product Line) 구현을 위한 플랫폼 범위결정 기법," 정보처리학회 논문지 (KTSDE) 10권 7호 251-256, 2021. (국방과학연구소, 한성대학교)

[정필수 20] 정필수, 강성원, "효율적인 소프트웨어 제품라인 회귀시험을 위한 자동화된 코드 기반 시험 방법," 한국정보과학회 소프트웨어공학 소사이어티 소프트웨어공학회지, vol.29 no.2, pp.1-6, 2020. (한국과학기술원)

[박병수 20] 박병수, 백승준, 이인섭, 서강선, 노성규, 박삼준, "무기체계 항법소프트웨어 플랫폼 피쳐모델링," 제22회 한국 소프트웨어공학 학술대회(KCSE), 2020. (국방과학연구소)

[김태영 20] 김태영, 이지현, "자동화된 소프트웨어 가변성 추출을 위한 정의/사용 정보 기반 제품군 공유 코드 정렬," 제22회 한국 소프트웨어공학 학술대회(KCSE), 2020. (전북대학교)

[남승우 19] 남승우, 혼다네스, 홍장의, "휘처모델과 태스크 유사도 기반 추천 시스템을 이용한 R&D문서 재사용 지원," 제21회 한국 소프트웨어공학 학술대회(KCSE), 2019. (충북대학교)

[한영훈 19] 한영훈, 강성원, 이동민, 박병훈, "소프트웨어 제품 라인 개발을 위한 가변성 모델의 무모순성 검증," 제21회 한국 소프트웨어공학 학술대회(KCSE), 2019. (한국과학기술원, T3Q)

[강성원 18] 강성원, 한영훈, 안휘, 이지현, "소프트웨어 제품라인 개발을 위한 합성적 피쳐 모델 CFM과 CFM을 지원하는 가변성 모델링 도구 SVM," 정보과학회지, 36(4), 9-18, 2018. (한국과학기술원, 전북대학교)

[한영훈 18] 한영훈, 강성원, 박연하, 박병훈, "SVM: 소프트웨어 제품 라인 개발을 위한 가변성 모델링 도구," 제20회 한국 소프트웨어공학 학술대회(KCSE), 2018. (한국과학기술원, T3Q)

[김진우 18] 김진우, 이우신, 김학준, 진소연, 조세현, "데이터링크 소프트웨어에 대한 소프트웨어 제품라인 공학 적용 연구," 한국컴퓨터정보학회논문지, vol.23, no.12, 통권 177호 pp. 65-72, 2018. (한화 시스템)

[강성원 17] 강성원, 체계적인 소프트웨어 제품라인 개발, 홍릉과학출판사, 2017. (한국과학기술원)

[황병한 17] 황병한, 진영택, "웹 시스템 군의 개발을 위한 소프트웨어 제품라인 공학의 적용," 한국전자거래학회지, vol.22, no.2, pp. 39-60, 2017. (푸투라스소프트, 한밭대학교)

[엄석환 15] 엄석환, 강성원, 김진규, 이선아, "소프트웨어 제품 라인의 요구사항과 아키텍처 간 추적성 모델링," 정보처리학회 논문지 (KTSDE) 4권 11호 487-498, 2015. (한국과학기술원)

[김정아 14] 김정아, 김세훈, "소프트웨어 제품라인의 휘처모델과 구성요소간 가변성에 대한 일관성 검증 규칙," 정보처리학회 논문지, vol.3, no.1, 통권 016호 pp.1-6, 2014. (관동대학교, 에듀크래프트)

[백하은 14] 백하은, 이지현, 강성원, "조합 시험 설계를 이용한 소프트웨어 제품라인 시험항목 생성 방법," 정보과학회논문지: 소프트웨어 및 응용, vol.41, no.7, pp. 469-480, 2014. (한국과학기술원, 전북대학교)



## 신진연구자 소개

강원대학교 이재권 교수



• 이재권 박사 (Dr. Jaekwon Lee)

조교수

소프트웨어공학 연구실

강원대학교 컴퓨터공학과

jaekwon.lee@kangwon.ac.kr

### 주요 약력

- 2024.09~현재 강원대학교 컴퓨터공학과 조교수
- 2023.06~2024.08 University of Ottawa, 박사후연구원
- 2022.09~2023.06 University of Luxembourg, 박사후연구원
- 2022.09 University of Luxembourg 컴퓨터공학과 박사 (지도교수: Lionel C. Briand)
- 2015.02 충북대학교 컴퓨터공학과 석사 (지도교수: 정우성)
- 2012.02 충북대학교 컴퓨터공학과 학사

### 주요 연구분야

- Automated test generation
- Fuzzing
- Mutation testing
- Search-based software engineering

### 대표논문

- [1] Jaekwon Lee, Seung Yeob Shin, Lionel C. Briand, and Shiva Nejati, "Probabilistic Safe WCET Estimation for Weakly Hard Real-time Systems at Design Stages," in ACM Transactions on Software Engineering and Methodology (TOSEM), Vol.33, No.2, pp. 32:1-34, Feb 2024.
- [2] Jaekwon Lee, Enrico Viganò, Oscar Cornejo, Fabrizio Pastore, and Lionel C. Briand, "Fuzzing for CPS Mutation Testing," in proceedings of the 38th IEEE/ACM International Conference on Automated Software (ASE2023), Esch-sur-Alzette, Luxembourg, Sep 11-15, 2023, pp. 1377-1389.
- [3] Jaekwon Lee, Seung Yeob Shin, Shiva Nejati, Lionel C. Briand, and Yago Isasi Parache, "Estimating Probabilistic Safe WCET Ranges of Real-Time Systems at Design Stages," in ACM Transactions on Software Engineering and Methodology (TOSEM), Vol.32, No.2, 37:1-33, Mar 2023.
- [4] Jaekwon Lee, Seung Yeob Shin, Shiva Nejati, and Lionel C. Briand, "Optimal Priority Assignment for Real-Time Systems: A Coevolution-Based Approach," in Empirical Software Engineering (EMSE), Vol.27, pp. 142:1-49, Aug 2022.
- [5] Jaekwon Lee, Dongsun Kim, Tegawendé F Bissyandé, Woosung Jung, Yves Le Traon, "Bench4BL: Reproducibility Study on the Performance of IR-based Bug Localization," in proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2018), Amsterdam, Netherlands, Jul 15-21, 2018, pp. 61-72.

### 소개글

안녕하세요, 강원대학교 컴퓨터공학과에 조교수로 재직중인 이재권입니다. 소프트웨어공학 소사이어티에 인사를 드릴 수 있게 되어 영광입니다. 임용 전에는, Lionel Briand 교수님의 지도로 룩셈부르크 대학에서 박사과정을 마쳤고, 동 대학과 오타와 대학교에서 박사후 연구원으로 근무 했습니다. 주 연구분야는 AI 기법들을 이용한 소프트웨어 테스트 자동화이고, 소프트웨어 설계 검증 등의 연구도 해왔습니다. 대부분의 연구는 협력업체들인 ESA, Huld, GomSpace, LuxSpace, Blackberry와 같은 회사들과 함께 진행하여서, 가상물리시스템을 대상으로 소프트웨어 설계 및 구현에 대한 테스트 및 검증을 자동화하는 방법들에 대해 연구해왔습니다.

### Q. 강원대학교 컴퓨터공학과는 어떤 곳인가요?

A. 강원대학교는 강원도 춘천시에 위치한 국립 종합 대학교 입니다. 현재 1도 1국립대학교를 목표로 도내 여타 국립대학들과 통합을 추진하면서 글로벌30에 선정되었습니다. 이를 기반으로, 우수연구지원 및 외국인 학생 유치에 많은 노력을 하고 있습니다. 그 중 컴퓨터공학과는 23명의 전임교수들과 600명 이상의 학부 재학생과 함께 운영되는 큰 규모의 학과입니다. 우리 학과는 여러 분야의 교수님들이 함께 협업하여, 다양한 연구프로젝트와 산학협력을 통해 학생들이 실질적인 문제해결 능력을 갖춘 인재를 양성하고 있습니다. 개인적으로는 캠퍼스 안에 언덕이 조금 많은 편이라 이동할 때 마다 운동하는 기분이 들지만, 체력 증진과 함께 춘천의 아름다운 풍광을 즐길 수 있는 점이 좋습니다.



〈강원대학교 한 카페에서의 전경〉



## Q. 춘천에서의 생활은 어떤가요?

A. 춘천은 호반의 도시, 낭만의 도시 등등 많은 별명을 가진 도시입니다. 춘천이란 이름 또한 봄이 빨리 오는 고을/시내란 의미를 가졌습니다. 그래서 많은 연인들이 찾아와 추억을 쌓아가는 도시이지요. 저도 이러한 지명이 마음에 들어 춘천으로 오는 걸음이 흥겨웠던 기억이 납니다. 이러한 배경은 춘천이 인구 30만 내외의 작은 도시지만, 우수한 자연경관을 기반으로 관광업 중심의 성장을 하도록 했습니다. 덕분에 강원도 도시 중 유일하게 서울광역시하철과 연결되어 서울과의 왕래도 매우 용이합니다. 그래서, 개인적으로는 여유로운 일상을 즐기고, 또 필요하면 서울의 문화도 간간히 즐길 수 있는 최적의 장소가 아닌가 생각합니다. 아직 춘천에 온지 얼마 되지 않아, 춘천의 봄을 즐겨 보지 못해서, 곧 다가올 봄이 기대됩니다.



〈소양강변의 전경〉

Q. 협력 업체들과의 연구 과정에서 기억에 남는 경험이 있나요?

A. 제가 박사과정 동안 몸담았던 연구실은 주로 협력 업체들과 함께 연구를 수행했습니다. 기업과의 협업은 실무에서 발생하는 문제를 이해하고, 이를 기반으로 연구 아이디어를 도출하는 데 매우 효과적이고, 실무 데이터를 활용해 실험할 수 있다는 점도 큰 장점이었습니다. 무엇보다, 연구를 통해 제안한 접근법을 실무에 적용해 그 효과를 직접 확인할 수 있다는 점은 매우 가치 있는 경험이었습니다. 한 번은 LuxSpace에 초대받아 회사를 방문했던 적이 있습니다. 그곳에서 실제로 개발 중인 인공지능을 직접 볼 수 있었던 것도 인상적이었지만, 특히 제가 제안한 접근법을 기반으로 문제를 해결할 수 있었다며 한 엔지니어가 저를 보며 미소 지었던 순간이 아직도 기억에 남습니다. 이처럼 연구 결과가 현장에서 활용되고, 실질적인 어려움을 해결하며 프로젝트에 기여하는 모습을 보는 순간들은 연구자로서 가장 큰 보람을 느낀 경험이었습니다.

또 다른 기억에 남는 경험은 저와는 전혀 다른 생각과 생활방식을 가진 사람들을 만난 일입니다. 오타와 대학에서 근무하던 시절, Blackberry의 한 엔지니어와 함께 연구를 진행한 적이 있습니다. 그분은 오타와 비행장에서 비행기 조종사 강사로 활동하며 개인 경비행기를 타는 독특한 취미를 가지고 있었습니다. 경비행기 조종은 저에게는 매우 낯선 취미였지만, 같은 일을 하는 분이 그런 활동을 즐기는 모습을 보며 생각보다 친근하게 다가왔고, 저도 관심을 갖게 되었습니다. 강원대에 임용되기 전, 그분이 자신의 경비행기에 저를 태워주겠다고 제안하셨지만, 서둘러

귀국하는 바람에 그 경험을 놓친 것이 무척 아쉽습니다.

Q. 어떤 연구를 수행중이신가요?

A. 현재 저는 가상물리 시스템의 테스트 케이스 자동 생성에 관한 연구를 진행하고 있습니다. 가상물리 시스템은 실생활과 밀접하게 연결되어 있거나 개발 비용이 상대적으로 높기 때문에, 품질 보증에 실패할 경우 큰 인명 피해나 재산적 손실을 초래할 위험이 있습니다. 따라서 고품질의 테스트 스위트가 요구됩니다. 제 연구는 퍼징(Fuzzing)과 뮤테이션 테스팅(Mutation Testing)을 활용하여 기존 테스트 스위트에서 탐지하지 못한 결함을 발견하고, 이를 기반으로 테스트 케이스들을 추가하여 더 안전하고 신뢰할 수 있는 테스트 스위트를 생성하는 것을 목표로 하고 있습니다. 최근에는 테스트 케이스 생성 시 의 적용을 방해하는 이슈들을 해결하기 위해 대규모 언어 모델(LLM)을 활용한 테스트 드라이버 생성 방법을 연구하고 있습니다. 또한, 이렇게 생성된 테스트 케이스는 소프트웨어가 진화하면서 무용지물이 되거나 컴파일 오류를 발생시킬 가능성이 있습니다. 이를 방지하기 위해, 테스트 스위트와 소스 코드의 공진화 방법에도 관심을 가지고 연구를 진행 중입니다.

## Q. 끝으로 소식지를 읽는 분들께 하고 싶은 말은?

A. 소프트웨어공학 소사이어티에 이렇게 소개할 수 있는 기회를 주셔서 감사합니다. 여러분과 함께 만들어갈 소프트웨어 공학의 미래가 매우 기대됩니다. 저는 앞으로도 소프트웨어공학 분야에서 의미 있는 연구를 통해 소프트웨어의 신뢰성과 효율성을 높이는 데 기여하고자 합니다. 학회나 기타 행사에서 직접 뵙고 함께 교류하며 많은 이야기를 나눌 수 있기를 바랍니다. 감사합니다.



## 신진연구자 소개

경북대학교 손정주 교수



• 손정주 박사 (Dr. Jeongju Sohn)  
 조교수  
 소프트웨어 진화 및 유지보수 연구실  
 경북대학교 IT대학 컴퓨터학부  
 jeongju.sohn@knu.ac.kr  
 https://jsohn92.github.io/

### 주요 약력

2024.03~현재 경북대학교 IT대학 컴퓨터학부 조교수  
 2021.10~2024.02 SnT, University of Luxembourg, 박사후 연구원  
 2021.08 한국과학기술원 전산학부 박사 (지도교수: 유신)  
 2017.02 한국과학기술원 전산학부 석사 (지도교수: 유신)  
 2015.02 이화여자대학교 컴퓨터공학 학사

### 주요 연구분야

- Automated Software Testing  
 - Software Evolution  
 - Software debugging  
 - Search-based Software Engineering

### 대표논문

- [1] Yu Pei, Jeongju Sohn\*, Sarra Habchi, and Mike Papadakis. 2024. Non-Flaky and Nearly-Optimal Time-based Treatment of Asynchronous Wait Web Tests. ACM Trans. Softw. Eng. Methodol. Just Accepted (September 2024). <https://doi.org/10.1145/3695989>
- [1] J. Sohn and S. Yoo, "Empirical Evaluation of Fault Localisation Using Code and Change Metrics," in IEEE Transactions on Software Engineering, vol. 47, no. 8, pp. 1605-1625, 1 Aug. 2021, doi: 10.1109/TSE.2019.2930977.
- [3] J. Sohn, Y. Kamei, S. McIntosh and S. Yoo, "Leveraging Fault Localisation to Enhance Defect Prediction," 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 2021, pp. 284-294, doi: 10.1109/SANER50967.2021.00034.
- [4] Jeongju Sohn, Sungmin Kang, and Shin Yoo. 2023. Arachne: Search-Based Repair of Deep Neural Networks. ACM Trans. Softw. Eng. Methodol. 32, 4, Article 85 (July 2023), 26 pages. <https://doi.org/10.1145/3563210>
- [5] J. Sohn and M. Papadakis, "CEMENT: On the Use of Evolutionary Coupling Between Tests and Code Units. A Case Study on Fault Localization," 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE), Charlotte, NC, USA, 2022, pp. 133-144, doi: 10.1109/ISSRE55969.2022.00023.

### 소개글

안녕하세요, 경북대학교 IT대학 컴퓨터학부에 조교수로 재직중인 손정주입니다. 소프트웨어공학 소사이어티 소식지에 제 소개를 할 수 있는 기회를 갖게 되어 영광입니다. 저는 석사 과정부터 주로 소프트웨어 테스팅 및 디버깅 자동화, 특히 소프트웨어의 변화를 분석하여 이를 활용하는 연구를 진행해왔습니다. 현재는 소프트웨어 진화 및 유지보수 연구실을 운영중입니다. 경북대학교 임용전에는 한국과학기술원에서 유신 교수님의 지도 아래에 석사와 박사학위를 취득하였고, 그 이후 룩셈부르크 대학의 SnT 센터에서 Mike Papadakis 교수님 아래에서 박사 후 연구원으로 근무하였습니다.

### Q. 현재 연구분야에 대해서 설명해 주세요. 왜 해당 분야를 연구하시는지, 향후 해당 연구분야에서 이루시고 싶은 방향에 대해서 말씀해 주시면 좋겠습니다.

A. 저는 소프트웨어 진화를 중심으로 다양한 유지보수 활동을 연구해왔으며, 현재도 이 주제에 집중하고 있습니다. 소프트웨어 진화란 환경과 요구사항 변화에 따라 소프트웨어를 지속적으로 수정, 확장, 최적화하는 과정을 의미합니다. 최근에는 자동화 기법이 도입되었지만, 소프트웨어 진화는 여전히 개발자 주도의 코드 변화를 중심으로 이루어지고 있습니다.

제 연구는 이러한 코드 변화를 분석하여 유지보수 활동의 효율성과 상호작용을 높이는 데 초점을 두고 있습니다. 특히, 소프트웨어 진화가 결함 발생에 미치는 영향과, 그로 인해 소프트웨어와 유지보수 활동이 받는 영향을 분석하고 있습니다. 저는 결함이 시간적, 공간적으로 균일하게 발생하지 않고, 특정 영역과 시점에 집중적으로 발생한다는 결함의 지역성(Fault Locality)에 주목하여, 이를 기반으로 코드 변화의 특성과 문맥을 분석하여 과거와 현재의 결함 식별 활동을 연계하고 유지보수 효율성을 높이는 방안을 제시하고자 합니다.

박사과정 동안에는 결함 식별(Fault Localization)과 결함 예측(Defect Prediction) 활동의 상호보완성을 연구했습니다. 현재는 이를 통합하여 소프트웨어 실패의 원인을 신속히 파악하고 잠재적 결함을 예측하는 연구로 확장하고 있습니다. 잠재적 결함은 테스트 부재나 실행 결과의 전파 실패로 아직 발견되지 못한 채 남아 있는 결함을 의미하며, 이는 유지보수 비용 상승의 주요 원인 중 하나입니다. 저는 이러한 잠재적 결함을 사전에 예측하고 해결하는 방법을 모색하고 있습니다. 또한, 최근에는 연구 범위를 결함 패치 생성으로 확장하였으며, 테스트 Flakiness, 특히 웹 테스팅에서 발생하는 Flaky 테스트 문제를 과거 코드 변화와 현재 코드베이스를 분석하여 해결하는 연구를 진행 중입니다.

궁극적으로, 저는 소프트웨어 변화를 중심으로 독립적으로 수행되던 유지보수 활동들을 유기적으로 연결하는 연구를 지향합니다. 빠르게 변화하고 복잡성이



증가하는 현대 소프트웨어 환경에서, 이러한 통합적 접근은 소프트웨어 품질과 개발 생산성을 크게 향상시킬 것으로 기대하고 있습니다 :)

**Q. 교수님께서 룩셈부르크 대학에 계셨는데요, 한국에서도 퍼뜨리고 싶은 문화가 있으면 소개해 주세요.**

A. 룩셈부르크 대학에서 가장 인상 깊었던 문화 중 하나는, 연구환경을 제외하면, 학생들 간의 사교활동을 뽑을 수 있습니다. 연구실 구성원들은 함께 페스티벌에 참여한다든지, 아니면 근교로 여행을 다녀오거나, 퇴근 후 보드게임을 같이 즐기는 등 각자의 취향에 따라 다양한 활동을 함께 했습니다. 이러한 사교활동은 구성원 간의 친밀도를 높였고, 결과적으로 서로 더 편안하게 의견을 나누고 협력할 수 있는 환경을 조성하였습니다.



〈룩셈부르크 시티 전경. 연구실 사람들의 주말 산책 코스 중 하나였습니다〉

물론 이러한 활동이 모든 연구실에 적합하지 않을 수도 있지만, 재밌고 창의적인 연구는 다양한 관점을 가진 사람들이 적극적으로 참여할 때 이루어진다는 점을 고려하면, 이런 사교활동은 구성원의 소속감과 연구 효율성을 동시에 높일 수 있는 장점이 있다고 생각합니다.

**Q. 국내 가장 대표적인 국립대학 중 하나인 경북대학교에서 근무 중이신데요, 학교 자랑 부탁드립니다.**

A. 경북대학교는 국내에서 가장 오래된 국립대학 중 하나로, 넓은 캠퍼스와 다양한 시설을 자랑합니다. 특히, 일청담 주변의 산책로와 중앙에 위치한 테니스장은 학생, 교직원 그리고 지역 주민들에게 사랑받는 공간입니다. 현재 재학생은 약 25,000명, 대학원생은 6,250명 이상으로 매년 약 5,000명의 신입생이 입학하며 활발한 학문적 활동이 이루어지고 있습니다. 경북 지역의 학생들이 높은 비율로 지원 및 입학하며, 지역 거점대학으로서의 이상적인 역할을 충실히 수행하고 있습니다 :)



〈경북대학교 본관. 이곳에서 최종면접을 봅니다〉

경북대학교 컴퓨터학부는 35명의 전임 교수와 4명의 겸임 교수님을 중심으로 인공지능, 데이터 사이언스, 네트워크 컴퓨팅, 시스템 소프트웨어, 컴퓨팅 이론 및 시스템, 그리고 비주얼 컴퓨팅, 총 6개 주요 분야에서 활발히 연구가 진행되고 있으며 학문적 협력과 지역 기업과의 협업을 통해 지역 발전에 기여하고 있습니다. 또한, 오픈랩을 통한 학부생 연구 프로그램, 학부-대학원 연계 프로그램 등을 통해 연구 중심 대학으로의 발전을 도모하고 있으며, 다양한 장학금과 연구 장려금을 통해 우수한 인재 유치에 지속적으로 힘쓰고 있습니다.

**Q. 학문 발전을 위해서는 성별로 인한 차별이 없어야 할 겁니다. 그간 양성 평등을 위한 여러가지 사회적 노력에도 불구하고 아직도 컴퓨터공학과에서 여학생들의 비중이 높지는 않습니다. 어떻게 해야 여학생들이 컴퓨터공학에 좀 더 관심을 많이 가질 수 있을지 의견 부탁드립니다.**

A. 어려운 질문입니다... 사실 이 문제는 뚜렷한 해결 방안이 존재한다고 보기는 어렵습니다. 이 문제는 국제적인 이슈로 여러 나라에서 여학생들의 참여를 촉진하기 위해 다양한 노력이 이루어지고 있는데, 예를 들어, 부가적인 지원 혜택을 제공해 여성 인재의 풀을 구성하려는 시도가 있습니다. 물론 이러한 시도가 모두 성공적인 것은 아니며, 때로는 역차별이라는 논란도 존재하기는 합니다. 그러나 지난 10년간 여학생, 그리고 여성 연구진들의 컴퓨터공학 분야 참여와 기여도를 분석하면, 극적인 변화는 아니지만 점진적으로 성장하고 있음을 확인할 수 있습니다. 결국, 저는 지금처럼 이 문제에 대해 지속적으로 관심을 갖고 다양한 노력을 이어간다면, 장기적으로 긍정적인 변화가 이루어질 것으로 기대하고 있습니다.



TSE 게재 논문, 다양한 언어모델을 활용한 오류 재현

## 다양한 언어모델을 활용한 오류 재현



• 강성민 박사후연구원  
한국과학기술원  
sungmin.kang@kaist.ac.kr

자동 오류 재현이란 어떤 오류에 대한 설명 또는 정보를 토대로 오류를 원하는 때에 재현할 수 있도록 테스트를 만드는 과정이다. 예컨대 유저가 '저장 버튼을 눌렀더니 앱이 튕겼어요'라고 말했다면, 그 설명을 바탕으로 어플리케이션에서 저장 버튼을 눌러서 갑자기 꺼지는 현상을 자동으로 일으킬 수 있도록 스크립트를 짜는 일이다. 오류를 고치기 위해, 또 실제로 고치고 나서 오류가 없어졌는지를 확인하기 위해서는 일반적으로 오류를 일으켜봐야 하기 때문에 오류 재현은 중요한 문제다. 그러나 그 중요성에도 불구하고 자연어 설명, 특히 개발자 간의 오류에 대한 의사소통을 분석하는 것은 지금까지 매우 어려운 일이었기 때문에 관련된 연구는 제한적이었다.

지난 24년 10월에 저널 TSE에 게재된 논문 "Evaluating Diverse Large Language Models for Automatic and General Bug Reproduction"은 많은 양의 자연어 정보를 학습한 거대언어모델 (Large Language Model)을 오류 재현에 적용하는 것이 주제다. 언어모델은 자연어와 코드 정보를 모두 학습하여 자연어를 기반으로 연관된 코드를 생성할 수 있기 때문에, 오류 재현과 같은 작업에 제격이라 할 수 있다. 특히, 이번 TSE 논문에서는 잘 알려진 OpenAI의 언어모델들 외에도 소위 open-weight LLM, 즉 일반인이 다운로드 받아서 사용할 수 있는 언어모델들이 버그 재현을 얼마나 잘하는지를 평가하여, 보안상의 이유로 OpenAI 언어모델을 사용할 수 없는 경우, 또 가지고 있는 gpu 자원이 많지 않은 경우 오류 재현에 어떤 모델을 쓰는 것이 가장 좋은지에 대한 지표를 제공한다.

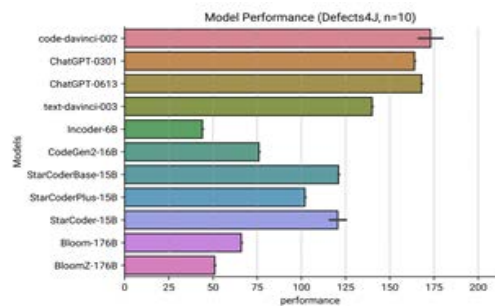
### 1. 오류 재현과 언어모델

앞서 이야기했듯이, 오류 재현은 많은 경우 자연어 설명을 동반하는데, 자연어와 코드를 동시에 다루는 기법은 소프트웨어공학 학계에서 오랫동안 고안되어왔지만, 변화무쌍한 자연어를 휴리스틱으로 분석하는 것에 상당한 어려움이 따랐다. 반면 언어모델의 상당한 강점 중 하나는, 자연어와 코드를 똑같은 벡터 형태로 처리하기 때문에, 모델 자체가 자연어와 코드가 섞여 있는 입력을 잘 처리하기도 하고, 자연어와 코드에 대한 각각의 처리 방법을 연구자나 사용자가 구현하지 않아도 돼서 편리하다. 그러한 이유에서 언어모델은 자동 오류 재현에 알맞은 도구이고, 본 TSE 논문의 전신인 ICSE'23 논문에서는 가장 먼저 쓸 수 있는 형태로 공개되었던 OpenAI의 Codex 언어모델 (code-davinci-002)을 활용하여 오류 재현 성능을 평가하였다.

그러나 OpenAI의 언어모델들을 연구에 사용하기에는 몇 가지 문제가 있다. 우선 OpenAI는 사업상의 기밀이라는 이유로, GPT-4 이후로는 자사의 언어모델에 대해서 거의 어떤 것도 공개하고 있지 않다. 따라서 OpenAI의 모델들은 블랙박스만 이용할 수 있을 뿐, 그 구조에 대해서 더 분석해서 유용한 결과를 얻는다면, 언어모델의 학습 데이터가 벤치마크의 평가 데이터를 포함하고 있는지를 아는 것이 불가능하다. 또 한편으로는, 이후 보일 실험 결과에서도 드러나듯, 언어모델의 행동 양상이 수시로 바뀌기 때문에 현재 버전의 언어모델에서 잘 되고 있다고 해서 앞으로도 잘 되리라는 보장을 하기가 어렵다. 더욱 난감한 것은 OpenAI는 특정 모델에 대한 사용을 예고 없이 막을 때도

있다는 것인데, 초창기 언어모델 기반 방법론을 평가할 때 많이 사용되던 Codex 모델이 2023년 3월 갑자기 서비스 종료된 것이 대표적인 예다. 이 모든 특징은 연구의 재현성을 위협하고, 언어모델에 대한 심도 있는 연구를 어렵게 한다.

다행인 것은 OpenAI 등의 대기업이 언어모델을 독식하는 것이 연구 생태계에 좋지 않다는 것을 인지한 사람이 많아서, 많은 단체에서 대중이 쓸 수 있는 라이선스로 언어모델을 학습하고 배포하고 있다는 점이다. 다운로드 받아서 개인 컴퓨터에서 사용할 수 있는 소위 open-weight 언어모델은 연구의 재현성에 더 친화적일 뿐만 아니라, 언어모델을 사용하는 회사의 입장에서도 사내 망에서 언어모델을 사용할 수 있으므로 보안 문제에서도 더 자유롭다. 2024년 들어서는 언어모델 종류도 무척 다양해졌지만, 본 논문에서는 실험을 했던 2023년 당시에 대표적으로 연구자가 사용할 수 있었던 몇 개의 언어모델을 사용했다. 구체적으로, open-weight 모델로는 전세계 개발자들이 함께 개발한 Bloom 및 StarCoder, Salesforce에서 개발한 CodeGen, Meta에서 개발한 InCoder 등을, closed-weight 모델로는 ICSE'23 논문에서 사용한 code-davinci-002와 그 이후로 새로이 공개된 ChatGPT API, 코드에 대해 별도의 학습을 거치지 않은 text-davinci-003을 활용하여 여러 언어모델의 오류 재현 성능을 비교하였다.



(a) LLM comparison on the Defects4J dataset.

Figure 1. 11개 언어모델의 버그 재현 성능 비교.

Figure 1은 OpenAI의 closed-weight 언어모델과 open-weight 언어모델의 성능을 비교한 결과이다. 그래프에서 볼 수 있는 것처럼 원래 실험을 했던 code-davinci-002의 성능이 가장 높았지만, ChatGPT 모델들의 성능도 크게 뒤처지지 않았다. 코드를 별도로 학습하지 않은 text-davinci-003의 성능은 code-davinci-002 모델에 비해 떨어져서, 높은 오류 재현 성능을 내기 위해서는 코드를 따로 학습하는 것이 유리하다는 점을 알 수 있다. 한편 open-weight 모델들의 성능은 모두 OpenAI의 언어모델보다는 떨어졌지만, 코드에 특화된 StarCoder 계열의 언어모델은 OpenAI의 언어모델이 재현한 오류의 75%를 재현할 수 있었다. 따라서 보안이 특히 중요하다면, StarCoder 등의 open-weight 모델을 직접 구동하여 보안을 챙기고 성능도 크게 잃지 않을 수 있다.

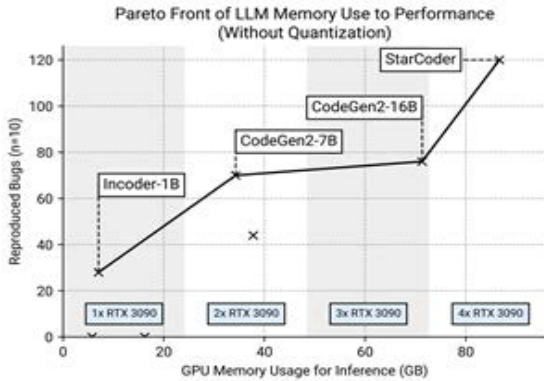


Figure 2. GPU 메모리 사용량 대비 언어모델의 버그 재현 성능.

Figure 2는 GPU 메모리 사용량별 오류 재현 성능을 나타낸다. 대형언어모델을 직접 구동할 경우 GPU 메모리가 특히 많이 필요한데, 많은 GPU를 구매하거나 구동하는 것은 비용이 상당하기 때문에 사용할 수 있는 GPU 메모리에 따른 성능을 평가하는 것이 연구자나 실제 사용자의 입장에서 유용한 정보라고 봤다. GPU 사용량으로 평가를 하니 StarCoder는 메모리를 가장 많이 사용하면서 성능이 가장 좋음을 확인할 수 있었고, 사용할 수 있는 메모리가 적은 경우에는 다른 모델을 사용하는 것이 현명한 선택일 수 있음을 볼 수 있었다. 가령 CodeGen2-7B 모델의 경우, 메모리도 StarCoder의 약 절반 정도를 쓰고, 버그 재현 성능도 약 절반이었다. 최근에는 DeepSeek, Qwen 등 성능이 좋은 작은 모델이 많이 나와서 사용을 고려해보자. 또한 Ollama 등의 플랫폼에서는 언어모델의 '압축된' 버전 역시 배포하고 있어서, 메모리가 적은 경우에 사용할 수 있다.

Model Prompt	GPT-0301 Prompt 1	GPT-0613 Prompt 1	GPT-0613 Prompt 2
Performance	164	72	168

Figure 3. OpenAI 언어모델의 프롬프트별 성능

다음으로 closed-weight 언어모델의 위험을 하나 짚고자 한다. Figure 3는 세 달 간격으로 발표된 GPT-3.5-turbo-0301과 GPT-3.5-turbo-0613의 행동 변화를 나타낸다. 실험 초기에는 GPT-0301에 실험했던 프롬프트, 즉 'Prompt 1'을 그대로 이용하여 GPT-0613에 실험하자 오류 재현 성능이 현저하게 떨어졌다. 이렇게나 현저하게 성능이 떨어지는 것이 이상해서 생성 결과를 살펴보니, GPT-0301은 테스트 함수를 만들었다면, GPT-0613은 테스트 파일을 통째로 만드는 경우가 잦아서, 언어모델이 테스트 함수를 만들 것이라는 가정을 한 이후의 평가 파이프라인이 제대로 처리하지 못하고 있음을 확인했다. 이를 보고 프롬프트에 'test method만 만들어라'라고 추가 명령을 하니 ('Prompt 2') GPT-0613의 성능이 확연히 개선되어 원래 GPT-0301 성능과 비슷해졌다. 이는 closed-weight 모델을 서비스에 활용하는 것의 위험성을 드러내는데, 모델이 버전이 바뀌며 예고 없이 이전에 되던 프롬프트가 오작동을 일으키거나, 언어모델의 생성 양상이 바뀌어서 이후 처리 파이프라인을 새로 만들어야 할 수 있기 때문이다.

추가로 언어모델에서 테스트를 생성할 때 '온도' (temperature) 파라미터의

영향, 다양한 언어모델의 '자기 일관성' (self-consistency) 에 대한 실험 등 더 많은 정보와 실험 결과를 본 논문에서 찾을 수 있으니, 궁금한 독자는 논문을 참고하기 바란다.

### 3. 앞으로의 연구

맺으며, 본 논문의 한계와 그에 따른 앞으로의 중요한 연구 방향을 논하고자 한다. 먼저, 본 논문에서는 오류 재현에 대한 언어모델의 성능만 비교했지만, 실제로 소프트웨어공학에서 언어모델이 활용될 여지는 오류 재현 이외에도 많기 때문에, 다방면에서 언어모델의 성능 및 메모리 등에 대한 상충 관계가 밝혀져야 할 것이다. 또한 언어모델의 '압축' (quantization)에 따른 성능 변화를 본 논문에서는 평가하지 않았지만, GPU 자원에 한계가 있는 현 상황에서는 압축의 정도에 따른 성능 변화 역시 평가가 되어야 실제로 언어모델을 사용할 때 다양한 제반 상황을 고려하여 최적의 선택을 할 수 있을 것이다.

한편, 최근 여러 언어모델이 나오고 있고, 그 특성 및 학습 과정이 다르기 때문에, 어떤 문제에서는 언어모델 A를 사용하고 다른 문제에서는 B를 사용하는 것이 나은 상황이 있을 수 있다. 또한 크고 비싼 언어모델 C과 작고 싼 언어모델 D가 있다면, D를 쓸 수 있는 상황에서는 최대한 쓰되 문제가 어렵다면 C를 활용하여 컴퓨팅 자원의 최소화를 꾀하는 방법을 개발할 수 있을 것이다. 이처럼 언어모델이 조합될 수 있는 경우의 수가 많기 때문에, 소프트웨어공학 문제에 맞는 조합법 또한 흥미로운 주제다. 본 논문에서는 이러한 조합을 고려하지 않았지만, 필자가 속해있는 COINSE연구실의 DroidAgent 등 다른 방법론은 언어모델의 성능에 따라 맡기는 일을 분리하고 있으므로 참고할 수 있다.

오류 재현 역시 아직 발전할 길이 멀다. 본 논문에서 사용하는 Defects4J 벤치마크의 버그들의 경우, 아직 약 33%의 버그만 자동으로 재현할 수 있는 실정이다. 실패한 경우를 분석해보면, 언어모델이 오류에 대한 설명을 오해하는 경우도 있고, 오류 재현을 위해서는 코드에 대한 상당한 전문지식이 필요한데 언어모델이 이를 갖추고 있지 못해서 실패하는 경우도 많다. 언어모델의 오류, 언어모델의 비전문성 등을 극복하여 오류 재현 성능을 높이는 것 역시 중요한 연구라 하겠다.

연구를 진행하면서 오류를 설명하는 오류 리포트 자체가 정보가 적은 경우가 종종 있음을 확인했다. 실제로 디버깅하다보면 몰랐던 정보가 밝혀져서 개발자가 필요한 내용을 더 구체적으로 말해주는 경우도 있을 것이다. 따라서 장기적으로 봤을 때, 개발자와 대화형으로 상호작용하며 오류를 재현하는 '행위자' (agent)의 개발이 중요한 다음 단계이리라 본다.

### 4. 감사인사

본 논문에 대해서 필자가 대표로 글을 써서 소개했지만, 많은 분들이 논문에 기여해주셨기 때문에 세상에 나올 수 있었다. 이 자리를 빌려 감사의 인사를 드리고자 한다. 동고동락하며 어렵고 귀찮은 일도 싫은 내색 없이 끝까지 책임감 있게 마무리한 공동일저자 윤주연 연구원, 학부생이라 어려웠을텐데 함께 실험에 참가하여 실험 결과 도출에 기여한 Nargiz Askarbekkyz 학생, 그리고 연구를 지도하고 가능하게 한 유신 교수님께 감사 인사를 드린다.





# ASE 게재 논문, CodeLLM 을 위한 데이터 레시피

## CodeLLM 을 위한 데이터 레시피



• 김기섭 박사  
화웨이 홍콩 연구센터  
falcnk00@gmail.com

여러가지 언어 모델의 등장 및 확산에도 불구하고, 사용된 학습 데이터에 대한 투명성 부족은 지속되고 있다. 보안 문제가 종종 언급되지만, 대부분은 품질 좋은 모델은 돈이 되기 때문에 기업들은 데이터 공개를 꺼린다. '그렇다면 직접 해보는 수밖에.'라는 생각으로 시작하게 된 연구가 지난 2024년 11월 미국에서 열린 ASE2024에서 발표한 논문 "DataRecipe --- How to Cook the Data for CodeLLM?"이다. 이는 경험적 실험을 통해 CodeLLM 학습에 어떤 데이터를 어떻게 사용해야 하는지를 파헤쳐보고자 하였다. 이 글을 통해 해당 논문의 기본 아이디어 및 논문을 작성하기까지의 소소한 이야기들을 공유하고자 한다.

### 1. 코드 생성의 과거, 현재, 그리고 미래

인간은 게으른 동물이기에 기술은 발전할 수밖에 없다. 소스코드는 개발자가 직접 '바닥부터 작성'하던 영역에서 검색을 통해 '참고 및 가져오기'를 지나 지금은 '자동적인 생성'을 원하고, 실제로 많은 코드들이 LLM을 통해 자동 생성되고 있다. 이는 생산성을 극대화하고 개발 프로세스의 속도를 혁신적으로 향상시키는 결과를 가져왔다.

자동 코드 생성의 장점은 단순히 빠른 작업 속도에만 그치지 않는다. 패턴 인식과 재사용을 통해 반복적인 코딩 작업을 최소화하고, 기존에 작성된 방대한 양의 코드 데이터베이스를 기반으로 한 최적의 코드 제안이 가능하다. 또한, 정확성과 일관성을 보장하여 사람이 실수하기 쉬운 영역에서 오류를 줄이고, 코드 품질을 높이는 데 기여하고 있다.

그러나 이와 같은 기술 발전이 모든 문제를 해결하는 것은 아니다. 자동 생성된 코드는 종종 보안, 성능, 가독성 측면에서 추가적인 검토와 수정이 필요하며, 특히 복잡한 도메인에 특화된 문제를 다룰 때는 여전히 인간 개발자의 전문성이 필요하다. 도구를 이해하고 활용하는 능력이 중요해진 지금, 개발자는 코드를 작성하는 기술뿐 아니라 도구를 효과적으로 사용하는 방법에 대한 학습도

필수가 되고 있다. 미래의 코드 생성 기술은 단순히 개발자의 요구를 수동적으로 따르는 것을 넘어, 더 스마트하고 주도적인 방식으로 발전할 것이다. 예를 들어, 시스템 요구사항을 기반으로 자동으로 소프트웨어 설계를 제안하거나, 사용자의 피드백에 따라 실시간으로 최적화된 코드를 제시하는 형태로 진화할 가능성이 크다. 또한, 도메인 특화 모델의 등장으로 특정 산업이나 업무 환경에 맞춘 더욱 정교한 코드 생성이 가능해질 것이다.

결론적으로, 코드는 이제 단순한 결과물이 아니라 인간과 AI가 협업하는 과정의 산물이 되고 있다. 기술 발전이 거듭될수록 코드 생성의 역할은 단순 반복 업무에서 벗어나 창의적 문제 해결의 보조자로 자리 잡을 것이다. 인간 개발자는 AI와 함께 문제의 본질에 더욱 집중할 수 있는 시대를 맞이하게 될 것이다. 이에, 필자는 다양한 방법으로 CodeLLM을 정교하게 학습시키는 방법을 연구한다.

### 2. 학습 데이터

데이터는 LLM의 성능을 결정짓는 가장 핵심적인 요소 중 하나다. 아무리 뛰어난 아키텍처를 설계하거나 최적화된 학습 방식을 적용하더라도, 부적절하거나 편향된 데이터를 학습시킨 모델은 한계를 가질 수밖에 없다. 이는 아이들을 가르치는 것과 같다. 모델의 학습 과정에서 데이터는 단순한 입력값을 넘어, 모델이 세상을 이해하는 틀과 패턴을 형성한다. 따라서 어떤 데이터를 어떻게 처리하고, 활용하느냐는 모델의 품질을 좌우한다.

대부분의 빅테크에서 LLM을 학습하고, 많은 대학교들과 정부 기관들까지도 자신들만의 LLM을 공개하는 현재, 그 어떤 기관도 모델 학습에 대한 모든 정보를 공개하지는 않는다. 돈이 되기 때문이다. 이에, 필자는 직접 데이터 품질과 연관 될 수 있는 여러 요소들을 바꿔보고 모델들을 튜닝 해봄으로써 데이터를 어떤 방법으로 가공해야 더 나은 모델을 만들 수 있는가에 집중해왔다.

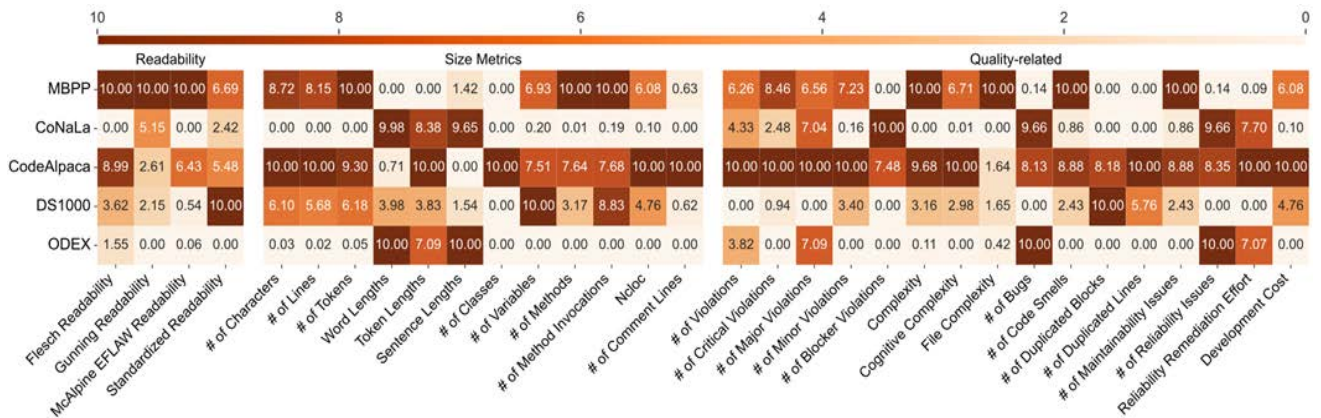


Figure 1 각 데이터 셋의 특성 파악: Readability, Size Metrics, Quality Metrics.

'고품질 학습 데이터'라는 단어는 분야별로 너무 다른 정의가 가능하기 때문에 통틀어 정의한 사람도, 명확하게 정의할 수 있는 사람도 아직은 존재하지 않는다. 따라서, 학습 데이터에 대한 품질 평가는 데이터 자체에 대한 평가를 선행하기보다는 LLM을 직접 학습시켜 특정 작업에 대한 결과를 바탕으로 어떤 특성을 가진 데이터가 얼마만큼의 성능을 내는지 평가해볼 수 있다.

### 3. 논문의 주요 아이디어 및 결과

본 논문은 이 논문은 CodeLLM의 학습을 위한 데이터셋 최적화 방법을 다루고 있으며, 코드 생성 작업에서 모델 성능에 영향을 미치는 다양한 데이터 품질 특성과 변형 전략을 조사한다. 먼저, <Figure 1>에서처럼, 코드 생성과 관련하여 가장 잘 알려져 있는 데이터셋들을 선정하고 수집했다. 그 후, 각 데이터셋의 특징들 (i.e., Readability, Size-metrics, Quality-metrics) 을 먼저 파악해보고, 여러 가지 LLM들을 순수한 데이터셋들을 사용해 fine-tuning 한 결과를 기준으로 사용하였다.

실제 데이터셋의 변형에 관련해서는 여러 논문들의 insight를 조합해 품질향상을 위한 코드 레벨 리팩토링 자동화, 텍스트와 코드 간의 의미적 상관관계, 그리고 데이터셋의 디자인 선택과 같은 요소들이 모델의 최종 성능에 영향을 미치는지에 대하여 식별하였다.

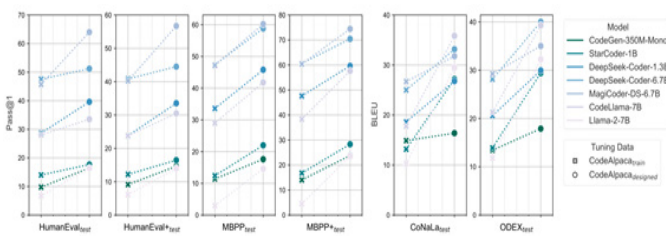


Figure 2 데이터셋 디자인 정렬을 통한 학습결과.

실험을 통해 데이터셋 디자인을 같은 형식으로 맞추어 주는 것 (<Figure 2>)이 모든 경우의 수에서 CodeLLM의 성능을 크게 향상시킨다는 것을 보여주었다, 이는 함수 레벨의 코드 제공과, 함수의 시그니처 및 설명이 포함된 디자인이 가장 효율적인 데이터셋 변형 방법을 알 수 있었다. 추가적으로 텍스트와 코드 쌍의 의미적 상관관계를 개선하는 것도 특정 벤치마크에서 성능을 향상시킬 수 있음을 확인할 수 있었다 (<Figure 3>).

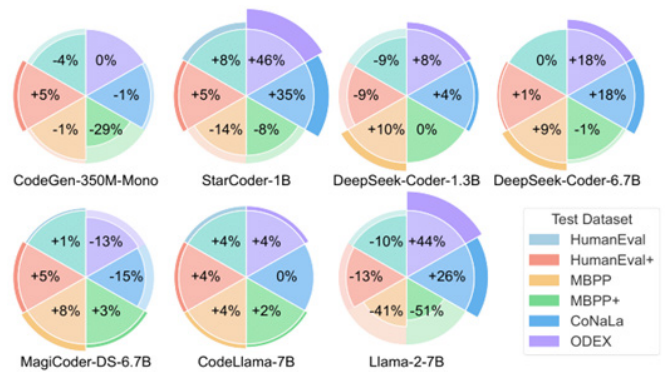


Figure 3 의미적 상관관계 향상을 통한 학습결과.

이 연구는 데이터 처리 선택이 자동 코드 생성 모델 학습을 최적화하는 방법에 대한 실험 가능한 통찰을 제공한다. 데이터 뿐 아니라 LLM도 architecture 및 pre-training에 사용된 데이터에 따라서 각자 다른 특징을 가지고 있다. 실험을 통해 LLM의 특성 또한 고려되어야 한다는 것을 알 수 있었고, 필자는 현존하는 LLM 들의 특성을 파악하기 위한 추가 연구를 진행 중이다.





## STAARBugs 를 소개합니다!

경북대학교 컴퓨터학부/소프트웨어재난연구센터  
최윤자 교수



• 최윤자, 교수  
경북대학교 컴퓨터학부/소프트웨어재난연구센터  
yuchoi76@knu.ac.kr

소프트웨어재난연구센터(<https://staar.knu.ac.kr>)에서는 C코드에서 반복적으로 발생하는 오류들을 분류하여 데이터베이스화하고 이들 오류정보들을 누구나 쉽게 접근하여 활용할 수 있도록 공개 인터페이스 STAARBugs (<https://staarbugs.org/>)를 개발하여 오픈하였습니다.

비슷한 SW오류는 계속 반복해서 나타나는 경향을 보입니다. 개발자들이 코드를 재사용하면서 오류까지 같이 전이되거나, 비슷한 알고리즘을 구현하다가 비슷한 실수를 하는 경우가 많기 때문입니다. Google의 보안팀인 Project Zero 팀에 따르면, 2020년 한 해 자체적으로 조사한 심각한 보안 오류 중 25%는 이전에 발견된 오류와 유사했다고 합니다. 더구나 최근에는 이러한 비슷한 오류를 학습한 시까지 등장하여 비슷한 오류를 생산하면서 이 문제는 더 가속되고 있습니다.

소프트웨어재난연구센터에서는 이러한 문제를 체계적으로 연구하고 해결하기 위해, 유사 오류를 모아서 정리한 STAARBugs를 공개했습니다. 연구센터는 2021년부터 유사 오류 재발 방지 기술을 연구하고 있으며, STAARBugs는 연구진이 지난 4년간 실제 C 코드에서 모은 13종류, 500여개 오류를 한데 모아 놓은 데이터베이스입니다. 각 오류마다 오류 리포트, 코드 저장소, 패치, 유사 오류, 오류 발현 경로 등 SW오류 연구에 필요한 여러 유용한 정보를 정리해 놓은 것이 특징이며, 연구가 진행됨에 따라 새로운 오류와 정보들이 꾸준히 추가될 예정입니다.

STAARBugs는 정적 분석, 피징, 오류 수정 연구용으로 최적의 정보를 제공하고 있으며, 이외에도 기계 학습용, 통계 분석용, 교육용 데이터로 활용될 수 있고, 누구나 활용할 수 있도록 모든 데이터가 공개되어 있습니다. 소프트웨어를 전문적으로 연구하고 교육하시는 소사이어티의 많은 분들에게 널리 사용될 수 있기를 희망합니다.

도구에 대한 제언이나 개선사항은 카이스트 허기홍 교수(kihong.heo@pr.osys.kaist.ac.kr) 또는 경북대학교 최윤자 교수(yuchoi76@knu.ac.kr)에게 연락주시기 바랍니다.

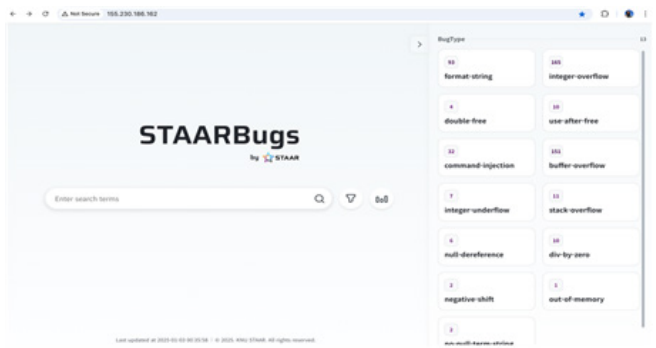
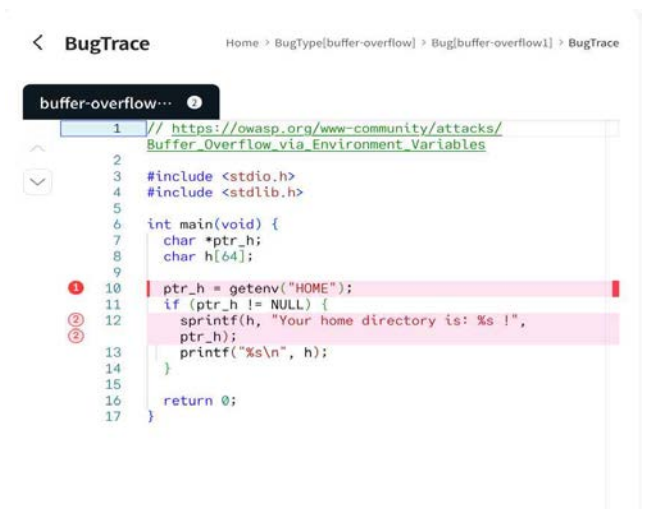
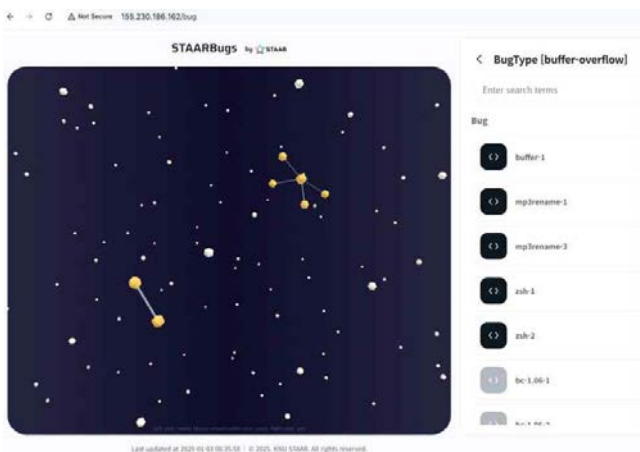


Figure 1. STAARBugs : main interface







KCSE 2025 개최 후기

## 제27회 한국소프트웨어공학 학술대회 (KCSE 2025)를 마치고



• 유준범 교수  
건국대학교 / 조직위원장

올해로 제27회를 맞는 한국소프트웨어공학 학술대회 (KCSE 2025)가 2025년 1월 20일부터 3일간 강원도 평창 한화리조트에서 개최되었습니다. KCSE 2025는 한국정보과학회와 한국정보처리학회가 주최하고, 한국정보과학회 소프트웨어공학 소사이어티와 한국정보처리학회 소프트웨어공학 연구회가 공동으로 주관하는 행사로서, 9개의 기관-기업이 후원해 주셨습니다. 이번 KCSE 2025는 2개의 기초강연, 4개의 튜토리얼, 3개의 신진연구자 세미나, 12편의 초청논문 발표, 63편의 논문 발표로 구성되었으며, 3일 동안 총 306명이 참가 등록하여 성황리에 진행되었습니다.

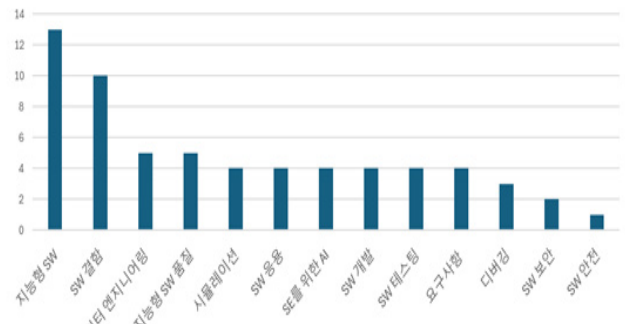
### 1. 행사를 준비하며

KCSE 2025 준비는 조직위원장과 학술위원장을 중심으로 조직위원과 학술위원을 구성하고, 학술대회의 주제를 결정하는 일부터 시작됩니다. 이번 학술대회는, AI 기술을 활용하여 소프트웨어를 보다 효율적으로 개발해 보자는 최근 연구 동향을 반영하여 “지능형 개발 기술과 소프트웨어 공학의 미래”가 주제로 선정되었습니다. KCSE 행사 준비는 크게 학술대회의 전반적인 행사 준비와 운영을 담당하는 조직위원장의 업무와 모든 기초연설, 초청논문 및 튜토리얼 연사를 섭외하고, 제출된 총 90여 편의 논문들을 총 42명의 학술위원(교수)의 다중 심사를 통해 최종 발표 논문을 선정하는 학술위원장의 업무로 구성됩니다. 우리 소프트웨어공학 소사이어티는 26년 동안 축적된 노하우를 잘 정리·보관하고 있어, 매년 전년도보다 조금 더 효율적이고 의미 있는 학술대회를 개최하기 위해서 노력하고 있습니다.

### 2. 분야별 논문 통계

학술대회가 성공적으로 개최되기 위해서 요구되는 가장 중요한 요소는 논문 발표일 것입니다. 이번 KCSE 2025는 빨라진 설날(구정) 연휴로 인해,

예년보다 2주 일찍 개최되었습니다. 이에 따라, 논문 모집 일정도 2주 일찍 시작되고 마감되어, 학부-대학원의 기말고사 기간과 겹치는 어려움이 있었음에도 불구하고, 작년보다 약 80% 증가한 91편의 논문이 접수되었습니다. 일반논문은 3명의 심사위원, 단편논문은 2명의 심사위원께서 심도 깊은 논문 심사를 진행해 주셨으며, 아쉽게도 3일의 학술대회 일정으로 소화 가능한 최대 논문인 51편이 최종 채택되었습니다. 채택되어 발표된 논문들의 주제는 아래의 [분야별 논문 통계]에서 보이는 것과 같이, AI를 활용한 소프트웨어공학 기술과 SW결함에 대한 연구가 많은 비중을 차지하였습니다. 이 중 총 2편의 최우수논문상(일반논문)과 9편의 우수논문상(일반, 단편, 산업체, 학부생논문) 논문이 선정되어 만찬에서 시상식이 진행되었습니다. 기관별로는 전북대와 충북대가 7편, 고려대 6편, KAIST 5편 등 총 24개 대학과 19개 연구소-기업이 논문을 제출하였습니다.



일차	세션	주제	강사
1일차	튜토리얼	언어모델을 이용한 소프트웨어 시험 및 오류 분석	강성민 박사 (KAIST)
		딥러닝 API 퍼징	김미정 교수 (UNIST)
		멀티코어 프로세서를 고려한 항공 SW 인증: DO-178B Level A 인증 10주년	김태호 교수 (서울여대)
	기초강연	LLM 기반 Agent에 대한 이해와 테스트 방법	백두산 박사 (KT)
신진연구자 초청세미나	AI for Software Engineering	정진민 부사장 (삼성전자)	
	강화 학습의 안전성과 안정성을 위한 반례 기반 클러스터링 및 MDP-PPO 하이브리드 접근법	권영구 박사 (경기대)	
	공용 공간의 인터랙티브 서비스 제공을 위한 인공지능 및 소프트웨어 공학 접근법	백경덕 박사 (KAIST)	
2일차	디버깅	시스템 오브 시스템즈와 소프트웨어 공학	송지영 교수 (한남대)
		지능형 SW (1)	시뮬레이션 SW
	SW 안전과 보안	SW 응용	SW 결함 (1)
	SW 결함 (2)	SE를 위한 AI	지능형 SW (2)
3일차	기초강연	재사용을 위한 소프트웨어 모델링	강성민 교수 (KAIST)
	요구사항 분석	SW 테스트	데이터 엔지니어링
		지능형 SW 품질	SW 결함 (3)

### 3. 프로그램 구성 및 운영

27년 전통의 KCSE 2025는 총 3일간 진행되며, 첫째 날에는 튜토리얼과 신진연구자 초청 세미나 및 기초연설(개회식)로 구성됩니다. 둘째 날과 마지막날은 논문 발표 세션만으로 진행되며, 폐회식으로 모든 일정이 종료됩니다. 기초강연은 산업체와 학계 각각 한 분을 초청하여, 첫째 날 개회식 직후와 둘째 날 만찬 전에 진행됩니다. 올해 KCSE 2025는 위의 표와 같이 진행되었습니다.

### 4. 행사 이모저모

이번 KCSE 프로그램의 특징을 살펴 보면, 튜토리얼과 기초강연, 발표논문들이 학술대회 주제인 “지능형 개발 기술과 소프트웨어 공학의 미래”에 잘 부합되었기 때문에, 각계의 많은 분들이 등록 및 참가 하셨습니다. 이번 학술대회의 사전등록자는 266명으로서 예년의 평균보다 약 20% 증가하였는데, 첫째날에 40명이 추가로 현장등록하여 총 306명이 등록하셨습니다. 조직에서는 최대 300명을 예상하고 식사, 명찰, 기념품 등을 준비하였기 때문에, 현장등록자가 30명을 넘어서 계속 증가하자, 등록데스크의 운영 학생들의 당황하는 모습도 볼 수 있었습니다. 등록자가 300명이 넘은 것은 이번 제27회 KCSE 2025가 처음입니다.

만찬 시간에는 최우수·우수논문상 시상자가 처음으로 공개되고 시상식이 진행되었습니다. 예상치 못했던 본인의 이름과 논문이 화면에 나오자 놀라서 당황한 학생들도 많았습니다.

올해는 전북대 류덕산 교수님 연구실이 7편으로 최다논문상을 수상하셨습니다. 저는 개인적으로 최다논문상이 가장 의미 있는 (부러운) 상인 것 같습니다.

KCSE 2025를 후원해 주신 9개의 기업체 대표분들께 감사장을 드림으로써 감사의 마음도 전달해 드렸습니다.

### 5. 행사를 마치며

이번 KCSE 2025의 조직위원장을 맡아 행사 전반을 준비하고 운영하면서, 학술대회의 성공적인 개최를 위해서는 많은 분들의 지지와 도움이 필요함을 다시 한번 느낄 수 있었습니다. 먼저 올해도 주저 없이 후원 요청을 흔쾌히 허락해 주신 기관과 기업에게 감사의 말씀을 전하고 싶습니다. 솔루션링크, 비트컴퓨터, 포털웍스, 이에스지, 다한테크, 브이플러스랩, 한국정보통신기술협회, 슈어소프트테크, 모아소프트게 지면을 통해 다시 한번 감사의 말씀을 드립니다. 마지막으로 대회장이신 아주대 이정원 교수님, 학술위원장 충북대 홍신 교수님, 10분의 조직위원, 42분의 학술위원, 좌장을 맡아 주신 10분의 교수님들과 3일간 최선을 다해 학술대회 운영을 책임져 준 11명의 건국대, 충북대, 아주대 학생들에게 감사의 인사를 드립니다.





KCSE 2025 참가 후기

# 제27회 한국소프트웨어공학 학술대회 (KCSE 2025) 참가 후기



• 정호연 학부생  
서울과학기술대학교

올해 2025년, KCSE가 1월 20일~22일에 강원도 평창에서 열렸습니다. 이번 행사는 300여명의 인원이 참가 등록하여 가장 많은 등록 인원과 함께 성황리에 마무리할 수 있었습니다.

이번 KCSE에 참여한 서울과학기술대학교 정호연 학부생(지도교수: 김진대 교수님)과 대화를 나누었습니다. 아래 문답에서 "편"은 편집자를 "정"은 정호연 학생을 지칭합니다.

## 편: 반갑습니다. 학부생으로서 참여하셨다니, 이번이 처음 참석하는 학술대회인가요?

정: 이제 막 두번째로 학술대회에 참가해보게 되었습니다. 또 제가 참여한 논문의 발표가 이뤄지는 첫번째 학술대회이기도 했습니다. 제2저자로 참여했기 때문에 발표를 제가 하지 않았지만 연구 및 논문 작성에 참여하여 완성시킨 결과물이 학술대회에서 발표되는 만큼 지도교수님과 제1저자인 학생과 함께 그 준비과정에서 많은 시간과 노력을 들였던 기억이 납니다.

## 편: 이번에 참여하신 논문이 최우수 논문상을 수상하셨죠? 축하드립니다! 간단한 논문 소개 부탁드립니다.

정: 감사합니다! 저희 논문의 주제는 "현실적인 소프트웨어 디버깅 시나리오에서 대규모 언어 모델의 성능 평가"입니다. 기존의 LLM을 사용하여 연구된 디버깅 방식이 그 시나리오에 있어서 현실적이지 않은 가정을 하여 이뤄진다고 보고 그렇다면 현실적으로 개발자가 LLM을 사용하여 디버깅하는 시나리오에서 LLM의 디버깅 성능이 어떨지, ChatGPT-4o와 Defect4J 데이터셋 중 NPE 버그를 사용하여 평가한 연구라고 할 수 있습니다. 그 결과 ChatGPT 모델이 대략 70% 정도의 버그에서 결함을 찾아내고 설명하였고 55% 정도의 버그에 대해 사용 가능한 패치를 제공하였습니다. 이 실험을 단독화를 거친 데이터셋에 대해 진행한 결과에서는 수치상으로 거의 동일한 결과가 나왔지만 ChatGPT의 답변이 피상적으로 바뀌는 경향을 관찰할 수 있었습니다.

## 편: 흥미로운 주제인데요! 본인이 참여한 발표 외에도 여러 발표들을 들어봤을 텐데 다른 발표들은 어떠셨나요?

정: 네, 이번 학술대회에서 여러 신진연구자, 석박사 과정생분들과 교수님 등 많은분들의 발표를 들을 수 있었습니다. 첫날 신진연구자 세미나 시간 때는 시스템 오브 시스템즈(SoS)를 주제로 한 강연을 들어봤었는데, 간단히 개념 정도만 들어봤던 분야의 연구자께서 시스템 간의 독립성 및 관계성을 바탕으로 한 여러 단계의 SoS 분류 및 주변 개념들과의 연관성에 대해 강연해주셔서 보다 흥미로운 주제에 대해 더 자세히 알 수 있던 좋은 기회였습니다. 또한 각 세션에서도 생소한 방식이나 새로운 방식으로 여러 문제에 대해 접근한 연구결과들을 들을 수 있는 기회들이 많았습니다. 저는 디버깅과 소프트웨어 결함 쪽에 집중해서 들었는데, 이번 주제가 "지능형 개발 기술과 소프트웨어

공학의 미래"였던 만큼 해당 분야에서도 앞으로의 디버깅이나 결함과 관련된 새로운 방법론들에 대한 연구결과들을 들을 수 있었고 특히 언어모델을 주제로 한 연구들이 많았던 것 같습니다. 그리고 주제에 대한 발견뿐 아니라 학술대회에서의 발표 방법에 대하여도 많이 배울 수 있는 기회였던 것 같습니다. 자연스럽게 발표를 이어가시는 연구자분들의 발표를 듣다 보니 제스처부터 억양, 빠르기, 진행 방식 등 좋은 발표에 대해 생각해 보고 배울 수 있었습니다.

## 편: 이번 학술대회에서 가장 인상적이었던 것을 꼽자면 어떤 것이 있을까요?

정: 개회식부터 기조강연, 여러 세미나실에서 열리는 많은 발표자분들의 세션 시간이나 튜토리얼까지 모두 좋은 시간이었지만 역시 가장 기억에 남았던 건 둘째 날 저녁의 넓은 홀에서 열린 저녁 식사 및 시상식이었습니다. 넓은 홀에서 모든 참여자분들과 함께 즐겁게 저녁 연회를 즐기면서 조직위원회의 소개와 시상식, 이번 년도와 내년의 학술대회들에 대한 소개 시간 등이 이어졌는데, 감사하게도 저희 논문이 최우수상을 탈 수 있었기에 더 인상적이었던 것 같습니다. 중간에 깜짝 공연이 있었던 것도 기억이 나네요. 또 폐회식 때는 짧은 폐회 인사와 경품 추첨 시간이 이어졌었는데, 상품이 굉장히 풍성해서 제가 타지 못했던 것이 많이 아쉬웠던 것이 기억납니다.

## 편: 끝으로 이번 학술대회에 대한 간단한 소감 부탁드립니다.

정: 청정한 산속 공기에서 설경을 배경으로 하여 많은 분들의 노고가 들어간 학술 대회에 참여할 수 있어서 정말 감사한 기회였습니다. 이번 학술대회 참가로부터 좋은 자극을 받아 저의 학술적인 성취에도 더욱 정진해야겠다는 생각이 듭니다. 내년에는 울산과학기술원에서 열리는 것으로 알고 있는데요, 그때도 많은 것을 배우고 느낄 수 있는 기회가 될 수 있을 것 같아 기대가 됩니다.





## 산업 혁신과 사회 변화를 선도하는 디지털 엔지니어링

(주)모아소프트 장정훈 부문장/상무

**편집자:** 저는 모아소프트를 '디지털 엔지니어링 플랫폼 구축 전문기업'으로 소개받았습니다. 디지털 엔지니어링이 무엇인지, 그리고 모아소프트가 이를 어떻게 활용하고 있는지 소개 부탁드립니다.

장정훈 상무: 디지털 엔지니어링은 기존의 엔지니어링 방법론에 디지털 기술을 접목하여 제품 개발의 효율성과 정밀성을 극대화하는 접근 방식입니다. 이를 통해 제품 설계, 검증, 운영까지 전 과정에서 데이터 기반의 의사결정을 지원하고, 보다 신뢰성 높은 시스템을 구축할 수 있습니다.

모아소프트는 다양한 산업에서 요구하는 복잡한 시스템을 효과적으로 관리할 수 있도록 디지털 엔지니어링 기반의 플랫폼을 개발하고 있습니다. 우리의 솔루션은 방위산업, 우주항공, 자동차, 철도 등 다양한 분야에서 활용되며, 시스템 설계부터 검증까지의 프로세스를 자동화하여 비용과 시간을 절감할 수 있도록 돕고 있습니다.



**편집자:** 구체적으로 어떤 방식으로 플랫폼을 개발하고 있으며, 차별화된 강점은 무엇인가요?

장정훈 상무: 모아소프트의 디지털 엔지니어링 플랫폼은 단순한 소프트웨어 제공을 넘어, 산업 전반의 디지털 전환을 가속화하는 데 중점을 두고 있습니다. 우리는 최신 모델 기반 시스템 엔지니어링(MBSE) 기술을 적용하여 엔지니어링 데이터의 체계적인 관리를 지원하며, 시뮬레이션 및 디지털 트윈을 활용하여 실시간 성능 분석 및 최적화가 가능합니다.

또한, AI 및 빅데이터 기술을 활용한 스마트 검증 시스템을 통해 오류를 사전에 예측하고, 유지보수 비용을 절감하는 데 기여하고 있습니다. 이러한 접근 방식 덕분에 복잡한 시스템을 효과적으로 운영할 수 있으며, 다양한 산업에서 신뢰받는 엔지니어링 솔루션을 제공하고 있습니다.

**편집자:** 기존 엔지니어링 방식과 비교했을 때, 디지털 엔지니어링이 가지는 핵심적인 차별점은 무엇인가요?

장정훈 상무: 기존의 엔지니어링 방식은 물리적인 프로토타입을 기반으로 반복적인 검증 과정을 거치는 반면, 디지털 엔지니어링은 가상 환경에서

시뮬레이션과 디지털 트윈 기술을 활용하여 문제를 사전에 예측하고 최적화할 수 있습니다. 이를 통해 개발 단계에서 발생할 수 있는 오류를 사전에 방지하고, 실시간 데이터 분석을 통해 운영 및 유지보수의 효율성을 극대화할 수 있습니다.

또한, 전통적인 문서 기반의 개발 방식에서 벗어나 모델 기반 시스템 엔지니어링(MBSE, Model-Based Systems Engineering)을 활용하여 보다 직관적이고 체계적인 설계를 가능하게 합니다. 이를 통해 복잡한 시스템을 보다 쉽게 관리하고 협업할 수 있도록 지원합니다.

**편집자:** 디지털 엔지니어링에 대한 각 산업의 인식은 어떠한가요? 도입 과정에서의 도전 과제도 있을 것 같습니다.

장정훈 상무: 산업별로 디지털 엔지니어링에 대한 인식과 도입 속도는 차이가 있습니다. 방위산업과 우주항공 분야에서는 이미 디지털 엔지니어링이 필수적인 요소로 자리 잡고 있으며, 시뮬레이션과 데이터 분석을 통한 설계 및 검증 프로세스가 활발하게 활용되고 있습니다. 자동차 산업에서도 전기차와 자율주행 기술이 발전하면서 디지털 엔지니어링의 중요성이 더욱 부각되고 있습니다.

반면, 일부 전통적인 제조업이나 인프라 산업에서는 디지털 엔지니어링이 아직 생소한 개념일 수 있습니다. 디지털 전환이 필수적이라는 점을 인식하고 있지만, 기존 시스템과의 통합 문제, 초기 투자 비용, 디지털 기술에 대한 이해 부족 등이 도입 과정에서의 도전 과제로 작용할 수 있습니다.

또한, 디지털 엔지니어링은 단순히 새로운 소프트웨어를 도입하는 것이 아니라, 조직 문화와 프로세스 전반에 변화를 요구하는 경우가 많습니다. 따라서 기업들이 이를 성공적으로 정착시키기 위해서는 기술적인 부분뿐만 아니라 인력 교육, 데이터 관리 체계 구축, 장기적인 전략 마련이 필수적입니다.



**편집자:** 디지털 엔지니어링이 사회 전반에 미치는 영향은 어떤 것들이 있을까요?

장정훈 상무: 디지털 엔지니어링은 단순히 기술의 발전을 넘어서 사회 전반에 혁신적인 변화를 가져오고 있습니다.

첫째, 안전성과 신뢰성이 중요한 산업에서 필수적인 기술로 자리 잡고 있습니다. 예를 들어, 항공기나 자동차 같은 복잡한 시스템에서의 설계 오류는 큰 사고로 이어질 수 있습니다. 디지털 엔지니어링을 활용하면 설계 초기 단계에서부터 안전성을 확보하고, 이를 지속적으로 검증할 수 있기 때문에 보다

안전한 제품을 제공할 수 있습니다.

둘째, 지속 가능성과 환경 보호 측면에서도 큰 역할을 합니다. 가상 환경에서 시뮬레이션을 통해 제품의 성능을 최적화하면 불필요한 자원 낭비를 줄일 수 있으며, 효율적인 에너지 사용이 가능해집니다. 이는 탄소 배출량 저감과도 직결되는 중요한 요소입니다.

셋째, 인공지능(AI) 및 데이터 분석 기술과 결합하여 미래 산업의 패러다임을 바꾸고 있습니다. 기존의 엔지니어링 방식에서는 불가능했던 방대한 데이터 분석이 가능해지면서, 시스템의 유지보수와 운영 방식까지도 획기적으로 변화하고 있습니다. 예를 들어, 항공기 엔진의 상태를 실시간으로 모니터링하여 고장이 발생하기 전에 예방 조치를 취하는 것이 가능해지고 있습니다.



### 편집자: 그렇다면, 모아소프트가 추구하는 디지털 엔지니어링의 미래 방향성은 무엇인가요?

장정훈 상무: 모아소프트는 디지털 엔지니어링을 통해 산업별로 최적화된 맞춤형 솔루션을 제공하고자 합니다. 현재는 방위산업과 우주항공, 자동차 산업을 중심으로 솔루션을 개발하고 있지만, 앞으로는 더 다양한 산업으로 확장해 나갈 계획입니다.

특히, 인공지능과 빅데이터를 활용한 스마트 검증 시스템을 강화하여 시스템의 신뢰성을 더욱 높이고자 합니다. 또한, 디지털 트윈 기술을 발전시켜 가상 환경에서의 시뮬레이션 정확도를 향상시키고, 이를 통해 고객이 최적의 결정을 내릴 수 있도록 지원할 것입니다.

모아소프트는 디지털 엔지니어링을 통해 단순한 소프트웨어 제공을 넘어, 산업 전반의 패러다임을 변화시키는 역할을 수행하고자 합니다. 앞으로도 기술 혁신을 지속하여 다양한 산업의 디지털 전환을 지원하고, 보다 효율적이고 안전한 미래를 만들어 가겠습니다.

### 편집자: 모아소프트가 아닌 상무님을 대상으로 개인적인 질문 하나 하겠습니다. 모아소프트의 소프트웨어융합부문 부문장이자 경남지사 (항공전자소프트웨어센터)를 책임지고 있는 장정훈 상무님이 디지털 엔지니어링 분야에서 개인적으로 가장 중요하게 생각하는 점은 무엇이며, 이 기술이 앞으로 어떻게 발전할 것이라고 보시나요? 또한, 디지털 엔지니어링을 통해 이루고자 하는 개인적인 비전이 있을까요?

장정훈 상무: 저는 사실 어릴 때부터 '비행기를 만드는 사람이라는 꿈'을 가지고 있었습니다. 하늘을 나는 기술에 대한 매력은 그저 기계적이고 기술적인 것뿐만 아니라, 사람들에게 자유와 가능성을 선사하는 부분이 크기 때문입니다. 디지털 엔지니어링은 그런 꿈을 실현하는 데 중요한 역할을 한다고 믿습니다. 예를 들어, 비행기의 설계나 성능을 디지털 트윈을 통해 시뮬레이션하고, 시로 예측하여 더 안전하고 효율적인 항공기를 만드는 것이 바로 디지털 엔지니어링이 제공하는 가능성입니다.

저는 27년간 우주항공 산업에서 다양한 경험을 쌓아왔으며, 강의 및 세미나를 통해 항공기의 신뢰성과 안전성 확보방안, 디지털 엔지니어링 도입의 중요성을 알리는 데 힘써 왔습니다. 앞으로도 저에게 주어진 자리에서 디지털 엔지니어링을 통해 비행기를 설계하고 제작하는 과정에서 더욱 정밀하고 신뢰성 있는 시스템을 구축하는 데 도움이 되고 싶습니다.

그 과정에서 발생할 수 있는 잠재적인 오류를 미리 예측하고 해결할 수 있는 시스템을 개발하며, 결국에는 더 안전하고 혁신적인 항공기 개발을 선도하는 것이 제 꿈입니다.

### 편집자: 마지막으로, 디지털 엔지니어링을 연구하거나 활용하고자 하는 분들에게 조언 한 말씀 부탁드립니다

장정훈 상무: 디지털 엔지니어링은 단순한 기술이 아니라, 전 산업에 걸쳐 필수적인 패러다임이 되어가고 있습니다. 이를 효과적으로 활용하기 위해서는 개별 기술뿐만 아니라, 시스템적인 사고를 갖추는 것이 중요합니다.

또한, 변화하는 기술 트렌드를 빠르게 습득하고, 다양한 산업과의 융합을 고려하는 시각이 필요합니다. 디지털 엔지니어링을 활용한 혁신이 앞으로 더욱 가속화될 것이므로, 이에 대한 관심과 연구를 지속한다면 미래 산업에서 큰 기회를 찾을 수 있을 것입니다.

### 장정훈 부문장/상무 ((주)모아소프트)

이메일 : jhjang@moasoft.co.kr

- 경력 : - 1998년 모아소프트 입사
- 약 27년간 국방, 철도, 자동차, 원자력 분야 등의 소프트웨어 시험 및 검증 활동 수행
- 항공분야 소프트웨어 기술지원 100건 이상 수행 경험

### 모아소프트 소개

(주)모아소프트는 '디지털 엔지니어링 플랫폼 구축' 전문기업입니다. 국방, 우주항공, 자동차 등 다양한 산업 분야에서 27년간 쌓아온 소프트웨어 신뢰성 확보 방안에 대한 노하우를 바탕으로, 디지털 융합 기반의 M&S, 가상 개발 검증, IPS&RAM-C, AI/DATA, MRO 등 각 분야의 전문 기술을 플랫폼 형태로 제공합니다.

### 모아소프트의 소프트웨어 공학 기술

모아소프트는 다양한 산업 분야에서 소프트웨어의 신뢰성과 안전성을 보장하는 기술을 보유하고 있습니다. 주요 기술은 다음과 같습니다:

#### • 소프트웨어 검증 및 인증 컨설팅

- DO-178C: 항공 소프트웨어의 안전성을 보장하기 위한 국제 표준
- ISO 26262: 자동차 산업에서 기능 안전을 보장하는 표준
- IEC 61508: 전자 시스템의 기능 안전을 보장하는 국제 표준

#### • 모델 기반 시스템 엔지니어링(MBSE)

- 요구사항 관리 및 추적성 확보 방안
- 안전 필수 시스템의 모델 기반 설계 및 검증 방안

#### • 임베디드 소프트웨어 안전 분석

- 임베디드 소프트웨어의 코드 분석 및 검증
- 정적 분석을 통한 소프트웨어의 보안 취약점 식별

#### • AI/Data Science

- 인공지능과 데이터 과학을 활용한 분석 및 예측

#### • 시스템 컨설팅

- 수명주기 관리 및 신뢰성 분석을 통한 시스템 최적화(TLCSM/IPS/RAM)
- 전자 시스템의 PCB설계 및 부품 국산화 지원

#### • 디지털 융합 솔루션

- 디지털 트윈을 통한 시뮬레이션 및 M&S



## 기관탐방 University of Notre Dame

경상국립대 이선아 교수

2024년 연구년을 맞아 6개월 정도 노터데임(University of Notre Dame)에 다녀왔습니다. 방문 계기는 재직 중인 경상국립대학교가 항공-우주-국방의 글로벌 대학으로 특성화하고 있으며, 항공우주산업(KAI)과 우주항공청이 옆에 있는 지리적 요건으로 소프트웨어공학 역시 항공, 우주, 국방으로의 특성화가 요구되고 있기 때문입니다. 이를 먼저 선도적으로 리딩하고 있는 Jane Clelang-Huang 교수님의 Software and Requirements Engineering Center<sup>9)</sup>는 2018년 ICSE에 Dronology<sup>10)</sup> 페이퍼를 게재한 이후 지속적으로 드론 연구에서의 소프트웨어 공학 기법 적용을 리딩하고 있는 연구실입니다.



Jane Clelang-Huang 교수님

노터데임대학교는 미국 인디애나주 사우스벤드에 위치한 세계적으로 유명한 명문 사립 대학교입니다. 노터데임대학교는 미국에서 가장 유명한 가톨릭 대학교 중 하나로, 현재까지도 가톨릭의 강한 종교적 정체성을 가지고 있습니다. 또한 "파이팅 아이리쉬" 라는 문구를 흔히 볼 수 있으며, 아일랜드계 미국인의 자부심을 상징하는 학교로 중요한 의미를 가집니다.

이러한 특성을 나타내듯, 학교는 광활한 수도원같이 아름답고 조용하며, 평화로운 분위기였습니다. 교정에 있는 골든 돔(Golden Dome)은 관광 명소로 넓은 캠퍼스에서 길을 잃을 때 해당 돔을 찾아 위치가 어디쯤인지 가능하곤 했습니다. 또한 도서관에도 세계의 구세주라는 벽화가 그려져 있습니다. 교정에는 다람쥐들이 많이 다녀서 쉽게 볼 수 있습니다.

또한 학교 내에 스타디움이 있고, 시즌별로 미식 축구 게임이 있습니다. 1930년 개장한 대규모 풋볼 스타디움으로, 미국의 대학 스포츠의 전통과 열정을 느낄 수 있는 좋은 경험이었습니다.



골든 돔과 헤스버그 도서관



스타디움

도착한 시기는 2024년 6월로 여름방학을 맞아 캠퍼스는 한산했고, 학생들은 대부분 인턴을 떠난 상황이었습니다. 이 시기에 Computer Science and Engineering 학과에서는 학부 학생들의 연구 인턴을 진행하고 있었습니다. 시카고와 거리가 2시간 이내로 가까워서 시카고에 있는 거버너스 주립대학교(Governors State University) 학생들과 교수님들이 참여하는 연구 프로젝트가 진행되었습니다. 이러한 방문 연구 인턴제를 여러 학교가 연계하여 시행하는 것도 좋은 아이디어로 보입니다.

9) <http://sarec.nd.edu/index.html>

10) <https://ieeexplore.ieee.org/abstract/document/8444851>





연구 인턴 수행하는 학부 학생들과의 사진

또한 아쉽게도 연구 미팅하는 사진을 찍지는 못했지만, 아래 사진에서 보듯이 연구실 학생들은 친근해서 쉽게 어울리고 친구가 될 수 있었습니다. 아래는 연구실에 있는 아투로(Arturo) 학생의 멕시코식 아들 돌잔치에서 찍은 사진입니다.



대학원생들 모임

마지막으로, 나사(NASA) 프로젝트를 위한 여름 발표 워크숍에 참여하였습니다. 해당 워크숍에서 인상 깊었던 것은 하나의 나사 프로젝트 목표를 위해 각 교수님들과 학생들이 연구하고 이러한 연구 성과를 공유하고 있었습니다. 한국도 집단 연구 추세가 가속화되고 있는데, 여러 교수님들과 이런 공동 프로젝트를 많이 하면 좋을 것 같아서 아래 사진을 찍었습니다.

Jane Clelang-Huang 교수님 연구실에서 배운 점은 작고 강한 연구실이라는 점입니다. 박사 5명 정도로 이루어져 있고, 모두가 드론 소프트웨어 프로젝트에 초점을 맞추고 있었습니다. 또한 Drone Response<sup>11)</sup>라는 회사와 함께, 실제적으로 발생하는 문제를 관찰하고 해결하고 있었습니다.

해당 연구실에 제가 한 일은 드론과 사람의 인터페이스에서 발생하는 문제를 분석하고 방지하기 위한 안전성 분석 기법을 공부하였습니다. 그리고 3개월은 제가 원하는 개인 연구를 수행하였고, 12월에 돌아오기 바로 전날 발표하고 돌아왔습니다.

혹 해당 연구실에 방문을 고려하고 계신 분이 있어서 조언을 드리자면, 드론 프로젝트, 안전성 분석, UAM(Urban Air Mobility)에 관심이 많은 소프트웨어공학자면 좋을 것 같습니다. 또한 리눅스 환경의 랩탑 컴퓨터를 준비하시고, 기본 드론 시뮬레이션과 드론 조작을 알고 가시면 참여가 용이할 것 같습니다.

단 6개월이었지만, 정이 많이 드는 연구실, 대학과 도시였습니다. 처음에는 비행 노선도 몰라 해맸는데, 시카고 직항으로 타고 가시면, 사우스 밴드로 항공, 기차, 렌터카 모두 교통이 편리합니다.

끝으로 Jane Clelang-Huang의 위키피디아<sup>12)</sup>를 공유합니다. 감사합니다.



11) <https://droneresponse.ai/>

12) [https://en.wikipedia.org/wiki/Jane\\_Clelang-Huang](https://en.wikipedia.org/wiki/Jane_Clelang-Huang)



## 소사이어티 광장

### 축하합니다!

#### • 박사학위 수여

- 안희 (KAIST, 지도교수: 강성원): 2024년 8월 박사학위 취득 (논문제목: 커넥터 매핑 기반 실행 뷰 아키텍처 재구축 방법 / Reconstruction of an execution architecture view by identifying connector mapping rules ), 42dot 취업
- 김수민 (KAIST, 지도교수: 차상길): 2025년 2월 박사학위 취득 (논문제목: 차등 테스트를 적용하여 머신 코드 변환기의 정확도를 향상시키는 기술에 관하여 / Enhancing the Correctness of Machine Code Translator using Differential Testing), 졸업 후 KAIST 박사후연구원 취업 예정

#### • 수상

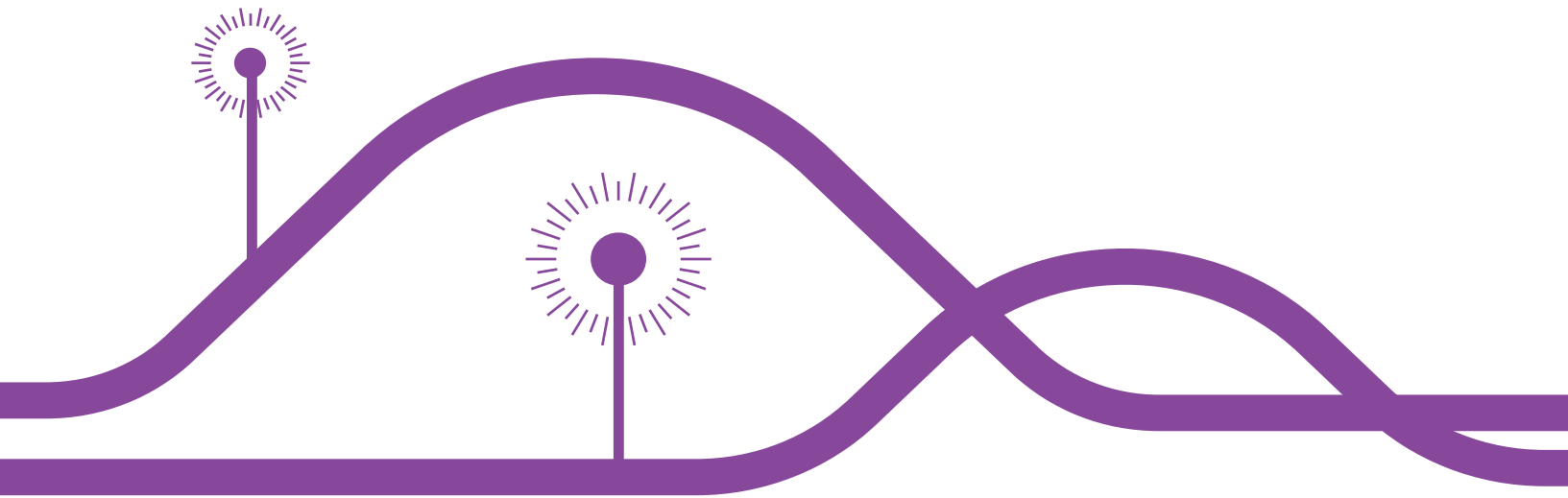
- KAIST 차상길 교수 연구팀, GCC, Clang, MSVC를 비롯한 컴파일러에서 자동으로 버그를 검출하는 테스트 알고리즘을 고안하여 소프트웨어공학 분야 최우수 학술 대회 중 하나인 ISSTA 2024에서 ACM Distinguished Paper Award 수상

#### • 퇴임

- 최병주 교수, 이화여자대학교 정년퇴임, 2024.08
- 차성덕 교수, 고려대학교 정년퇴임, 2025.02

#### • 오픈 저장소 소개: STAAR Bugs ( <https://staarbugs.org/> )

- 자세한 내용은 “8. STAAR Bugs 소개” 글 참고





## 2025 국내외 학술대회

Full name	학술대회	대회일자	논문마감	URL	기준 (행사, 제출)
International Conference on Software Testing, Verification and Validation	ICST 2025	2025/03/31~04/04	2024/10/2	<a href="https://conf.researchr.org/home/icst-2025">https://conf.researchr.org/home/icst-2025</a>	행사
International Conference on Software Engineering	ICSE 2025	2025/04/27~05/03	2024/03/22 (first cycle), 2024/08/02 (second cycle)	<a href="https://conf.researchr.org/home/icse-2025">https://conf.researchr.org/home/icse-2025</a>	행사
Software Engineering Education and Training	CSEE&T 2025	2025/04/28~29	2024/10/10	<a href="https://conf.researchr.org/home/icse-2025/cseet-2025">https://conf.researchr.org/home/icse-2025/cseet-2025</a>	행사
Mining Software Repositories Conference	MSR 2025	2025/04/28~29	2024/11/9		행사
Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering	FSE (ESEC/FSE) 2025	2025/06/23~27	2024/09/12	<a href="https://conf.researchr.org/home/fse-2025">https://conf.researchr.org/home/fse-2025</a>	행사
International Symposium on Software Testing and Analysis	ISSTA 2025	2025/06/25~28	2025/10/31	<a href="https://conf.researchr.org/home/issta-2025">https://conf.researchr.org/home/issta-2025</a>	행사
Requirements Engineering Conference	RE 2025	2025/09/01~05	2025/03/10	<a href="https://conf.researchr.org/home/RE-2025">https://conf.researchr.org/home/RE-2025</a>	제출
International Conference on Software Maintenance and Evolution	ICSME 2025	2025/09/07~12	2025/03/13	<a href="https://conf.researchr.org/home/icsme-2025">https://conf.researchr.org/home/icsme-2025</a>	제출
Conference on Object-Oriented Programming, System, Languages, and Applications	OOPSLA 2025	2025/09/12~18	2025/03/25 (round 2)		제출
International Conference on Model Driven Engineering Language and Systems	MODELS 2025	2025/10/05~10	2025/04/04	<a href="https://conf.researchr.org/home/models-2025">https://conf.researchr.org/home/models-2025</a>	제출
International Symposium on Software Reliability Engineering	ISSRE 2025	2025/11/04~07	2025/05/10	<a href="https://issre.github.io/2025/">https://issre.github.io/2025/</a>	제출
International Conference on Automated Software Engineering	ASE 2025	2025/11/16~20	2025/05/30	<a href="https://conf.researchr.org/home/ase-2025">https://conf.researchr.org/home/ase-2025</a>	제출



## : 기고문 및 소식 모집



Dream



Think



Idea



come true

소프트웨어공학 소사이어티 소식지는 여러 연구자분들의 생각과 소식을 나누는 광장입니다. 다음과 같은 구성으로 소식지를 구성하고자 하오니, 여러분들의 적극적인 참여를 바랍니다. 투고글의 형식은 자유형식이며, 분량은 A4 기준 2~4페이지입니다.



- 기고문: 소프트웨어공학 및 소사이어티에 대한 생각 (자유주제)
- 신진연구자 소개: 만 40세 이하 또는 박사학위 취득 후 7년 이내의 연구자 소개
- 국내외 학술행사 소개: 주요 학술행사 소개, 학술행사 참여 후기 등
- 기관소개: 소프트웨어공학연구 관련기관 소개
- 소사이어티 광장: 소사이어티의 새로운 소식 나눔

### ▶ 소사이어티 알림

- 소프트웨어공학 소사이어티에서는 매년 소프트웨어공학 우수논문상을 추천하여 시상하고 있습니다. 최우수 학술대회에 논문 발표로 참가하는 학생에게 장려금(약 100만원 수준)을 지원할 예정입니다.
- 소프트웨어공학 소사이어티 소개 동영상 : <https://www.youtube.com/watch?v=HWGsy-Pyle0>
- 소프트웨어공학 소사이어티 페이스북 : <https://www.facebook.com/groups/668196744037453>

#### 제출방법

- 이메일 제출 (소프트웨어공학 소사이어티 편집부: [ksepup@gmail.com](mailto:ksepup@gmail.com))

#### 문의처

- 이주용 교수 (UNIST, 052-217-2123, [jooyong@unist.ac.kr](mailto:jooyong@unist.ac.kr))
- 정우성 교수 (서울교육대학교, 02-3475-2562, [wsjung@snue.ac.kr](mailto:wsjung@snue.ac.kr))
- 지은경 교수 (한국과학기술원, 042-350-7810, [ekjee@se.kaist.ac.kr](mailto:ekjee@se.kaist.ac.kr))
- 손정주 교수 (경북대학교, 053-950-5556, [jeongju.sohn@knu.ac.kr](mailto:jeongju.sohn@knu.ac.kr))



SOFTWARE  
ENGINEERING  
SOCIETY





## 소사이어티 조직도



### 발행정보

**발행일** 2025년 3월 10일

**발행인** 이정원

**발행처** 사단법인 한국정보과학회 소프트웨어공학소사이어티

**연락처** 경기도 수원시 영통구 월드컵로 206 아주대학교 원천관 305호 (우: 16499)

이정원 (전화: 031 219 1813, 홈페이지: <https://eslab.ajou.ac.kr/members/professor>)

홈페이지 : <http://www.sigsoft.or.kr>

# CROWN 2.0

## C 프로그램 자동 유닛 테스트 도구

### | CROWN 2.0 소개 |

CROWN 2.0 은, C 프로그램 소스 코드를 자동으로 분석하여, 프로그램의 모든 가능한 시나리오들을 수행하는 다양한 테스트 입력값을 100% 자동 생성하는 SW 자동 테스트 도구입니다.



### CROWN 2.0 Workflow

- 1 프로그램의 소스 코드를 자동으로 분석하여, 유닛 테스트에 필요한 테스트 driver와 테스트 stub 코드를 자동으로 생성
- 2 테스트 대상 프로그램의 함수와 해당 함수를 위해 자동 생성된 테스트 driver/stub 코드를 대상으로, AI 기반 Concolic 테스트 엔진을 사용하여, 모든 가능한 수행 경로를 실행하는 다양한 테스트 입력 값들을 생성
- 3 CROWN 2.0을 통해 달성한 테스트 커버리지 및 테스트 과정에서 발견한 대상 프로그램의 오류를 사용자에게 보고

### 제품 특징

- 브이플러스랩이 10년간 자체 개발한 AI 기반 Concolic 자동 테스트 엔진을 활용한 테스트 입력값 자동 생성
- 테스트 대상 함수를 위한 driver와 stub 코드 자동 생성
- 화이트 박스 소스 코드 테스트 커버리지 리포트
- 초보자도 사용하기 쉬운 웹 기반 GUI 인터페이스
- 생성된 테스트 입력값을 CSV 파일로 저장하여, 다른 테스트 도구에서 활용 가능

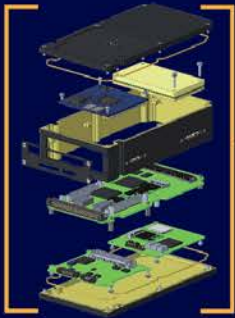
### 입증된 자동 테스트 성능

- 세계 최고의 SW 공학 국제학술대회 ICSE '19에, 현대모비스의 자동차 소프트웨어 적용 사례 연구 발표
- 분기 커버리지 90% 이상, MC/DC 커버리지 80% 이상 자동 달성 가능
- 현대자동차, LIG넥스원, ETRI, 국가보안기술연구소, 포항공대 등 납품 중



안전을 설계하는 소프트웨어 플랫폼

# 디지털 엔지니어링 플랫폼 구축 전문기업



모아소프트 자체개발  
비행제어컴퓨터(스마트 항공전자 Suite)



모아소프트 자체개발  
성능 검증용 비행체(MOA-EPISODE-1)

### 미션 요구사항 관리 및 추적성 확보

**Jama software**

- 요구사항 관리
- Verification & validation
- 3<sup>rd</sup> party 연동
- 실시간 협업
- 위험관리
- 프로젝트 관리

- 데이터의 분석과 예측을 통한 문제 해결 시간 최소화
- 요구 사항, 위험 및 테스트 관리를 위한 선도적인 솔루션

**SW 개발 프로세스 구축 - MBSE 프로세스 적용**

- JAMA를 활용한 소프트웨어 개발 프로세스 구축
- OFD, 비행제어법칙 MBSE 적용(SCADE Suite)

**Ansys STK 디지털기반 임무효과도 분석**

- 모든 도메인에서 미션 모델링, 시뮬레이션 및 분석
- 검증된 디지털 트윈을 이용한 제품 설계/테스트 수명 주기 가속화
- 다중 도메인 시스템 모델링

### MBSE를 활용한 설계 및 통합관리

**Ansys Scade**

- SCADE Architect: Embedded systems & Software Architecture
- SCADE Suite: Critical Embedded Control Software
- SCADE Display: Critical Embedded HTML Software
- SCADE Test: Embedded Control & HTML Software Testing

**Ansys medini**

- 모델 기반 안전성 분석
- 신뢰성 예측 및 분석
- 팀워크 및 리뷰 지원
- 엔지니어링 환경에 통합
- 모델 기반 안전성 분석 (Model-Based Safety Analysis)
- 추적성 및 검증
- 보고 및 문서화
- 사용자 지정 및 프로세스 적용

### 해석기반 설계 및 성능 검토

- 전자파 해석 HFSS
- 열유동 해석 Fluent
- 모델기반 시스템 해석 ModelCenter
- 제어기 설계 및 임베디드 코드 자동 생성 SCADE
- 광학 해석 Lumerical Zemax Speos
- 케이블 EMC 및 충전/방전 해석 EMC Plus Charge Plus
- 안전성 분석 medini analyze
- 구조해석 Mechanical
- 디지털 미션 엔지니어링 STK

### 소프트웨어 검증 & 인증

**SW의 신뢰성/안전성 확보를 위한 검증**

**LDRA**

- 요구사항 추적 관리
- 모델 기반 개발
- 정적분석
- SW 단위 시험
- SW 통합 시험
- 요구사항 검증
- 프로세스 준수 관리

**인증 컨설팅**

- DO-178C SW 컨설팅
- SW 안전성 확보를 위한 SW Safety 분석
- FACE 인증을 위한 VA(Verification Authority)
- 부품국산화 DO-254 HW 컨설팅

### 총수명주기관리

**CBM+특화연구센터 사업 참여**

**RAM/RAM-C 분석/관리 솔루션**

**opus suite** (by Systechon)

- 총수명주기 및 수리부속 최적화
- 운용 시뮬레이션 및 미션 성공률 분석
- 비용구조모델링 및 수명주기비용 분석

**부품 국산화/성능개량 사업**

- SI PI
- Schematic
- BOM
- X-ray
- Artwork



**사단법인 한국정보과학회 소프트웨어공학소사이어티**

주소: 경기도 수원시 영통구 월드컵로 206 아주대학교 원천관 305호 (우: 16499)

이정원 (전화: 031 219 1813, 홈페이지: <https://eslab.ajou.ac.kr/members/professor>)

홈페이지 : <http://www.sigsoft.or.kr>