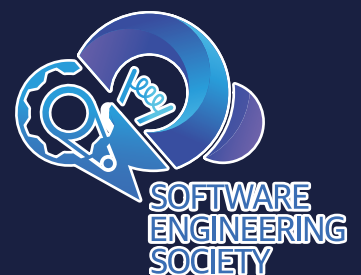




COMMUNICATIONS

OF SOFTWARE ENGINEERING SOCIETY

03/2026 VOL.6 NO. 1



CONTENTS



COMMUNICATIONS

OF SOFTWARE ENGINEERING SOCIETY

소프트웨어공학 소사이어티 소식

- 03 게스트 편집자 인사말 / 백종문 (KAIST)
- 04 기고문
AI 에이전트 시대, 소프트웨어공학의 과제와 에이전틱 문제 프레임 / 박찬진 (서울 AI 허브 센터장)
- 08 신진연구자 소개
유명성 (서울시립대)
최재승 (서강대)
- 12 Top Paper
LOSVER: Line-Level Modifiability Signal-Guided Vulnerability Detection and Classification (ASE'25) / 남도하 (백종문 교수 연구실, 석사과정, KAIST)
- 14 국내외 학술행사 소개
ASE 2025 개최 후기 / 유신 (KAIST)
APSEC 2025 참석 후기 / 이선구 (백종문 교수 연구실, 박사과정, KAIST)
KCSE 2026 개최 후기 / 이주용 (UNIST)
- 19 기관탐방
빅데이터 엣지 클라우드 서비스 연구센터 소개 / 고인영, 지은경 (KAIST)
- 23 소사이어티 광장
- 25 2026 국내외 학술대회
- 26 기고문 및 소식모집
- 27 소사이어티 조직도
- 27 발행정보

03/2026
VOL.6 NO. 1



• 발행인 이정원 • 편집인 손정주, 이주용, 정우성, 지은경, 김기섭 • 발행일 2026. 3. 9



백종문
KAIST 교수

한국 소프트웨어공학 소사이어티 회원 여러분, 안녕하세요. 저는 2026년 3월호(6권 1호) 게스트 편집을 맡게 된 KAIST 전산학부 백종문 교수입니다.

이번 호의 메인 테마는 “AI for SE”입니다. 생성형 AI와 에이전트 기술이 빠르게 확산되면서, 이제 AI는 소프트웨어 개발을 돕는 도구를 넘어 소프트웨어공학의 문제를 새롭게 정의하게 만드는 중요한 계기가 되고 있습니다. 요구사항 분석부터 시작하여, 구현, 테스트, 운영, 유지보수에 이르기까지 소프트웨어 개발 전 개발주기동안에 AI의 역할이 커지는 지금, 우리는 성능 향상뿐 아니라 신뢰성, 검증 가능성, 책임 경계, 운영 맥락까지 함께 고민해야 하는 시점에 서 있습니다.

이번 호 소사이어티 소식지에서는 이러한 문제의식을 정면으로 다루고 있습니다. 서울 AI 센터장을 맡고 계시는 박찬진 박사님께서 「AI 에이전트 시대, 소프트웨어공학의 과제와 에이전트 문제 프레임」이라는 제목으로 AI 에이전트를 단순한 모델 활용의 관점이 아니라, 실제 세계와 상호작용하는 공학적 시스템으로 바라보아야 한다는 점을 강조하며, AI 시대 소프트웨어공학의 새로운 과제를 제시하고 있습니다. AI를 잘 “쓰는” 문제를 넘어, AI를 포함한 시스템을 어떻게 설계하고 검증하며 운영할 것인가에 대한 중요한 질문을 던지는 글입니다.

New Leader 섹션에서는 두 분의 신진 연구자를 소개합니다. 서울시립대학교 유명성 교수는 클라우드·5G 환경의 고성능·고신뢰 네트워크 시스템과 보안, AI 기반 자동 생성·검증 연구를 통해 AI와 시스템 소프트웨어의 접점을 넓히고 있습니다. 서강대학교 최재승 교수는 프로그램 분석과 파징 기반의 소프트웨어 보안 연구를 중심으로, 정적 분석과 동적 테스트를 결합한 실용적인 취약점 탐지 방법론을 발전시켜 왔습니다. 두 연구자의 이야기는 AI 시대에도 소프트웨어공학의 핵심이 여전히 정확하고 믿을 수 있는 시스템을 만드는 데 있음을 잘 보여줍니다.

Top Paper 섹션에서는 KAIST 남도하 석사과정의 글을 통해 ASE 2025 발표 논문 LOSVER를 소개합니다. 이 연구는 코드 전체를 균일하게 보는 대신, 수정 가능성이 높은 라인에 주목하는 방식으로 취약점 탐지 성능을 높인 사례로, AI 기반 소프트웨어 분석에서 “어디에 집중할 것인가”라는 관점을 흥미롭게 제시합니다. 국내외 학술행사 소개 섹션에는 두 편의 글이 실렸습니다. 하나는 APSEC 2025 참가 후기로, KAIST 이선구 박사과정이 LLM4SE를 비롯한 최신 연구 흐름과 현장 분위기를 생생하게 전하고, 다른 하나는 ASE 2025 서울 개최 후기로, KAIST 유신 교수님이 대형 국제학술대회를 준비하고 운영한 경험을 통해 오늘날 학술대회의 의미와 과제를 깊이 있게 돌아봅니다.

이와 함께 국내학회 섹션에서는 UNIST 이주용 교수님이 KCSE 2026 개최 후기를 통해 UNIST에서 열린 학회의 운영 철학과 프로그램 구성, 그리고 공동체의 지속적인 발전을 위한 고민을 공유합니다. 기관탐방 섹션에서는 KAIST BECS 센터를 소개하며, 엠티 클라우드 환경에서의 고신뢰·고사용성 빅데이터 서비스 연구와 산학·국제 협력 활동을 살펴봅니다. 회원소식 섹션에는 회원 임용·수상 소식과 함께 신간 도서 소개를 담았고, 학회일정 섹션에서는 국내외 주요 소프트웨어공학 관련 학술대회 일정을 정리해 회원 여러분의 연구 및 학회 활동에 도움이 되도록 구성했습니다.

이번 호가 AI 시대를 맞이한 소프트웨어공학의 현재를 함께 점검하고, 연구와 실무의 다음 방향을 고민하는 작은 계기가 되기를 바랍니다. 귀한 원고를 보내주신 필자 여러분께 깊이 감사드리며, 소식지 발간을 위해 애써주신 소사이어티 회장단과 편집위원 여러분께도 감사의 말씀을 전합니다.

게스트 편집자 **백종문** 드림



AI 에이전트 시대, 소프트웨어공학의 과제와 에이전틱 문제 프레임



서울 AI 허브 센터장/서울대학교 산학협력 센터 박찬진 교수

1. AI 에이전트 시대와 산업 AX

매주 새로운 AI 기술 성과가 뉴스로 나오는 기술 혁명의 시기입니다. 많은 이가 ChatGPT, 제미니(Gemini), 클로드(Claude) 같은 파운데이션 모델의 놀라운 생성 능력에 열광하고 있지만, 이제 시장의 시선은 이러한 지능이 실제로 산업 현장에서 어떻게 가치를 실현하는가로 옮겨가고 있습니다. 단순한 답변을 넘어 IT 시스템을 직접 조작하고 업무를 완결하는 **에이전틱 AI(Agentic AI)**로의 진화를 통해 대화 상대에 머물지 않고 산업 현장의 시스템과 장비를 움직이는 **실질적인 전문 인력**으로 등장하고 있습니다.

특히 소프트웨어 분야에서는 이러한 **AI 전환(AX)**이 가장 먼저, 그리고 가장 파괴적으로 시작되었습니다. 올해 1월, AI 코드 에디터인 커서(Cursor)가 GPT-5.2를 활용해 약 300만 줄 규모의 웹브라우저 코드를 인간의 직접적인 개입 없이 단 7일 만에 구현해낸 사례는 상징적입니다. 또한, 신규 개발보다 훨씬 까다로운 1,250만 줄의 거대 레거시 코드에 새로운 기능을 추가하는 작업이 클로드 코드를 사용하여 단 7시간 만에 완료되었다는 뉴스도 있었습니다. 복잡하게 얽힌 기존 맥락을 파악해야 하는 사람들에게는 불가능에 가까운 “브라운필드(Brownfield)” 작업을 AI는 압도적인 문맥 파악 능력으로 해결하고 있습니다.

더구나 구글, 메타, 마이크로소프트 등 빅테크 기업들은 이미 신규 코드의 30% 이상을 AI로 생성하고 있으며, 이는 단순한 도구 도입을 넘어 대규모 구조 조정과 조직 재편으로 이어지는 산업적 지각변동을 일으키고 있습니다. 오픈소스의 거장 리누스 토발즈조차 AI를 “지루한 반복 작업을 대신해 주는 좋은 동료”라며 그 실질적 능력을 인정하기에 이르렀습니다.

우리가 목도하고 있는 AX의 실체는 단순히 코드 한 줄을 자동으로 짜는 것에 그치지 않습니다. 요구사항 분석부터 아키텍처 설계, 디버깅, 그리고 최종 통합에 이르기까지 다양한 전문 에이전트들이 협력하여 소프트웨어 생명주기(SDLC) 전체를 주도하는 방식입니다.

하지만 이 화려한 효율성 이면에는 거대한 질문이 숨어 있습니다. 수천만 줄의 코드를 순식간에 쏟아내는 이 **비결정론적(Non-deterministic) 지능**을 우리는 과연 통제하고 있는가? 청사진(Blueprint) 없이 이어지는 이 거대한 시스템은 과연 안전한가? 이제 우리는 확률적 에이전트를 산업의 규율 안으로 길들이기 위한 공학적 해답을 제시해야 합니다.

2. 확률적 에이전트와 청사진 없는 개발: 비결정론적 지능을 어떻게 통제할 것인가

현재의 에이전트 개발 현장은 명확한 공학적 청사진 없이 모호한 자연어 지시에만 의존하는 이른바 **“프레임 없는 개발(Frameless Development)”** 상태에 머물러 있습니다. 업계의 주된 노력은 더 정교한 프롬프트를 작성하거나 방대한 데이터를 컨텍스트 창에 밀어 넣는 프롬프트 및 컨텍스트 엔지니어링에 집중되어 있습니다. 하지만 이러한 접근은 본질적으로 거대 언어 모델(LLM)에 “정확한 의도”를 전달하려는 언어적 설득에 가깝습니다. 이는 마치 복잡한 정밀 공정을 숙련되지 않은 작업자에게 오직 구두로만 설명하며 작업이 잘 되기를 바라는 것과 같습니다.

문제는 이러한 구조적 틀 없이 던져지는 모호한 ‘언어적 지시’가 비결정론적 지능을 통제하기에는 지나치게 가변적이고 불완전하다는 점입니다. 확률론적 추론(Probabilistic Inference)을 수행하는 LLM을 공학적 규율 없이 복잡한 산업 시스템에 투입할 경우, 다음과 같은 세 가지 치명적인 위험이 필연적으로 발생합니다.

- **경계의 부재(Scope Creep): “선을 넘는 신입사원”** 관할 구역이 없으면 에이전트는 일반 상식에 의존해 권한 밖의 시스템에 간섭합니다. 이는 사고 시 실패의 원인이 모델의 오류인지 설계자의 방치인지 불분명하게 만드는 **책임의 공백(Responsibility Gap)**을 초래합니다.
- **근거의 부재(Absence of Grounding): “짐작으로 일하는 신입사원”** 도메인 지식(K)이라는 기준 정보가 없으면 에이전트는 모호한 입력을 확률적 추측(Hallucination)으로 매우며 잘못된 계획을 수립합니다. 이는 사용자의 의도와 에이전트의 실행 사이를 단절시키는 근본 원인이 됩니다.
- **피드백의 부재(Open-Loop): “보고를 생략하는 신입사원”** 물리적 상태 변화를 확인하지 않는 오픈 루프(Open-Loop) 상태입니다. “API 호출 성공이 곧 업무 완결”이라는 착각은 실행 결과의 지식 자산화를 가로막으며, 에이전트가 경험을 통해 성능을 개선하는 점진적 수렴을 불가능하게 합니다.

우리가 주목해야 할 본질적인 문제는 에이전트(M)에게 단일 일회성 프롬프트의

기계가 아닙니다. 기계의 부재를 막기 위해 에이전트의 관할 영역을 정의하고, 근거의 부재를 해결하기 위해 세상(W)의 도메인 지식(K)을 구조화 하여 주입함으로써, 에이전트의 **실행 명세(S)**로 변환해야 합니다. 나아가, 피드백의 부재를 극복하기 위해 실행 결과를 다시 검증된 지식 자산으로 환류시켜 다음 실행의 진화된 기반으로 삼는 공학적 선순환 체계를 구축하는 데 역량을 집중해야 합니다

결국, 내부 로직이 블랙박스인 **확률론적 기계(Stochastic Machine)**를 신뢰할 수 있는 엔지니어링 컴포넌트로 활용 하기 위해서는 모델 내부의 지능(IQ)을 튜닝하는 단계를 넘어서야 합니다. 이제는 지능 그 자체보다 시스템과 환경 간의 상호작용을 공학적으로 통제하고 지식화하는 새로운 패러다임이 절실한 시점입니다.

3. 패러다임의 전환: 기계의 내부가 아닌 “세계(World)”를 구조화하라

비결정론적 지능을 산업의 규율 안으로 길들이기 위해서는, 소프트웨어를 바라보는 공학적 관점의 근본적인 전환이 필요합니다. 그 핵심은 기계 내부의 논리를 정교하게 다듬는 것이 아니라, 기계가 발을 딛고 작용하는 “**세계(World)**”를 구조화하는 것입니다.

전통적인 요구공학은 사용자의 요구(R)를 기계에 대한 명세(S)로 변환하고, 이를 오류 없이 구현하는 데 모든 역량을 집중해 왔습니다. 이러한 경향을 Zave and Jackson [1997]는 환경을 도외시키고 코드에만 전념하는 **구현 편향(Implementation Bias)**이라 비판했습니다. 개발자가 기계가 오류 없이 수행되는 ‘내부적 정확성’에만 매몰될 때, 소프트웨어는 실제 세계와 동떨어진 닫힌 시스템이 될 위험에 노출됩니다.

마이클잭슨(Michael Jackson)이 제안한 **문제 프레임(Problem Frames)** 이론은 시선을 기계(M) 자체가 아닌, 기계가 상호작용하는 세계(W)에 대한 모델링과 기계와 세계(W)가 만나는 **접점(Interface, Shared Phenomena)**으로 돌립니다 [Jackson, 2001]. 이 관점에서 공학의 본질은 기계 그 자체가 아니라, 세계에 대한 도메인 지식(K)을 바탕으로 기계가 세상과 어떻게 상호작용하며 현상을 만들어낼 것인지를 설계하는 것입니다.

이 관계는 다음과 같은 **요구사항 충족 논증**으로 요약됩니다:

$$S, K \vdash R$$

이 공식은 기계의 명세(S)만으로는 결코 요구사항(R)을 충족할 수 없으며, 반드시 그 기계가 작동할 세계의 규칙과 특성(K)이 결합되어야 함을 의미합니다. 예를 들어, 특정 제어 신호를 보내는 것은 기계의 명세(S)일 뿐입니다. 실제 사용자가 원하는 요구사항(R)은 그 신호의 결과로 “엘리베이터가 원하는 층에 정지하고 문이 안전하게 열리는 상태”가 세계에서 실현되는 것입니다. 문제 프레임은 기계를 세상을 변화시키는 도구로 정의하며, 그 도구가 안전하게 작동할 수 있는 구조적 틀을 제공합니다.

지난 30년 동안 이 이론이 소수 의견으로 남았던 이유는 전통적인 디지털 환경이 지극히 인과적이고 결정론적이라고 가정되었기 때문입니다. 개발자는

“기계가 명령을 올바르게 내리면 세계는 설계된 대로 변할 것”라는 낙관적 가정 하에 내부 로직의 정확성만으로도 시스템의 안정성을 확보할 수 있었습니다.

하지만 스스로 판단하고 행동하는 비결정론적 에이전트의 등장은 이러한 낙관적 가정을 무너뜨렸습니다. 이제 기계(M)인 에이전트는 확률적으로 실행되는 블랙박스로 동작함에 따라, 내부 코드를 검증하는 방식으로 시스템을 통제할 수 없게 되었습니다. 여기서 문제 프레임은 에이전트 실행의 사전 및 사후에 세상과의 상호작용을 외부 구조화 하는 방식으로 새로운 해결책을 제시합니다.

본고에서 소개하는 **에이전틱 문제 프레임(Agentic Problem Frames, APF)**은 객체의 고전적 이론을 확률적 에이전트 시대에 맞게 재해석한 것입니다. 에이전트 내부의 추론을 파인튜닝하기 보다, 에이전트가 움직이는 “세계의 구조”를 공학적으로 정의함으로써 지능을 통제하고 적응시킵니다. 지능 그 자체가 아니라 지능이 활동하는 무대(Frame)를 설계하는 것, 이것이 비결정론적 지능을 산업에 활용되는 공학적 컴포넌트로 길들이는 APF의 핵심 철학입니다.

본 기고에서 소개하는 APF의 구조적 설계 원리와 접근적 수렴을 위한 논리적 근거는 필자의 최신 연구를 통해 상세히 확인할 수 있습니다¹⁾.

4. 에이전틱 문제 프레임(APF)의 실무 적용: 산업 장비 관리자 사례

본 절에서는 “산업 장비 관리자” 사례를 통해 실제 산업 현장에서 비결정론적 에이전트를 위한 공학적 명세를 어떻게 설계하는지 설명합니다. 해당 에이전트는 물리 장비의 실시간 데이터를 모니터링하여 자율적으로 이상 징후를 판단하고, 조치 후 결과를 보고하여 관리자로부터 평가받는 역할을 수행합니다.

에이전트의 직무 권한과 관할 영역을 정의하기 위해, 표 1과 같은 에이전틱 직무 기술서(AJD) 명세를 수립합니다. 이는 에이전트가 일반적인 확률적 추측에 의존하지 않고, 조직이 정의한 엄격한 가이드라인 내에서 동작하도록 강제하는 공학적 청사진이 됩니다.

표 1. 산업 장비 관리자 에이전트를 위한 AJD 명세

AJD 항목	공학적 명세 (Engineering Specification)
미션 (R)	안전 상태 유지 및 장비 다운타임 최소화
작업장 (W)	IoT 에지(Edge) 단말, ERP 시스템, 현장 안전 관리자
권한 경계 (M)	이상 탐지 및 긴급 정지 권한, 원인 분석 및 보고서 작성, 단, 임의의 재가동 및 결제는 금지
운영 컨텍스트 (S)	장비 사양서, 안전 규정 매뉴얼, 과거 고장 패턴 및 대응 이력
평가 방법 (K)	센서 데이터 기반 물리적 정지 확인, 관리자의 최종 승인

1) Chanjin Park, “Agentic Problem Frames for Engineering Reliable Domain Agents,” arXiv:2602.19065 [cs.SE], 2026. <https://arxiv.org/abs/2602.19065>

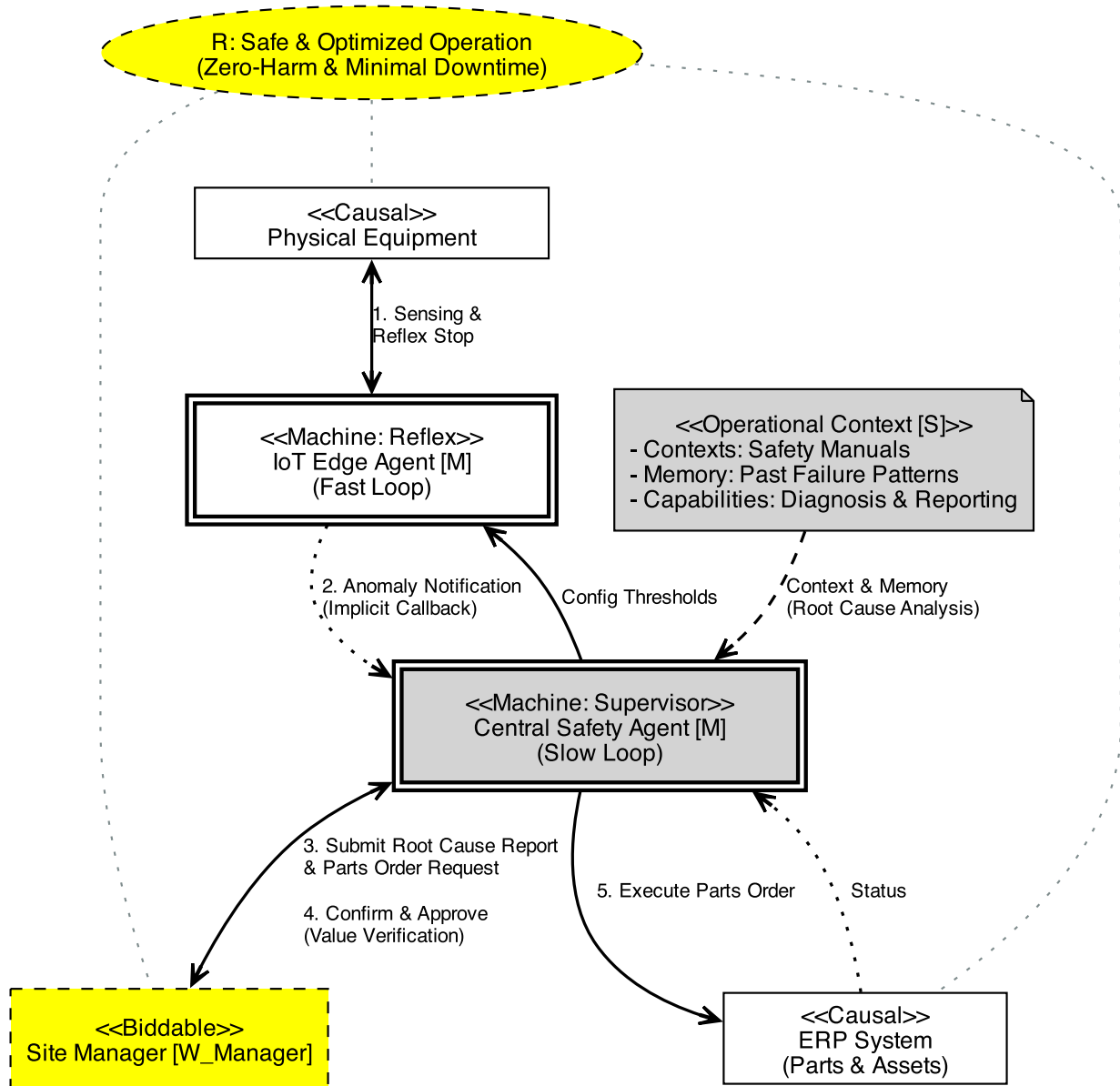


그림 1. 산업 장비 관리자 에이전트의 APF 다이어그램

그림 1은 APF를 통해 에이전트와 세상 간의 상호작용을 구조화한 모델입니다.

- **계층적 에이전트 엔티티 (M):** 에이전트는 이중 실선 상자로 표기되고, 단일 모델이 아닌 이중 루프 구조로 설계됩니다. 실시간 센싱과 긴급 정지를 담당하는 **IoT Edge Agent(Fast Loop)**와, 주입된 지식을 바탕으로 원인 분석 및 보고를 수행하는 **Central Safety Agent(Slow Loop)**로 분리하여 물리적 안전과 인지적 신뢰성을 동시에 확보합니다.
- **다중 도메인 상호작용 (W):** 에이전트가 작용하는 세계는 성격에 따라 세분화됩니다. 물리 장비와 ERP 시스템은 정해진 프로토콜에 따라 반응하는 **인과 도메인**으로 정의되며, 현장 관리자는 최종적인 가치 판단을 내리는 **자율 도메인(Biddable Domain)**으로 모델링됩니다. 각각 실선 상자와 점선 상자로

구분하여 제어 전략의 차이를 명시합니다.

- **운영 컨텍스트의 주입 (S):** 에이전트의 모든 판단은 일반 추론이 아닌, AJD를 통해 주입된 안전 매뉴얼, 과거 고장 패턴, 장비 설명서 등 구조화된 운영 컨텍스트를 근거로 이루어집니다. 점선 화살표는 이러한 지식의 흐름을 표현합니다.
- **결과 검증과 지식화 (K):** 에이전트의 실행 결과는 반드시 검증 절차를 거칩니다. 센서 데이터를 통해 장비의 물리적 상태 변화를 확인하는 콜백과, 관리자의 검토를 통해 실행의 적절성을 확정하는 승인 단계를 통해 확률적 추론의 실행 결과를 확정된 지식(K)으로 전환합니다. 콜백은 점선 화살표, 승인은 실선 화살표이고, 인과 및 자율 도메인과의 상호작용을 표현합니다.

에이전트는 다음과 같은 단계적 변환을 통해 비결정론적 상황을 통제된 공학적 이벤트로 전환합니다.

- 1. 이상 탐지 및 동적 명세 (S):** 모터의 진동 데이터(E_t)를 수신하면 주입된 안전 매뉴얼과 과거 패턴(C_t)을 참조하여 “장비 가동 정지”라는 구체적 실행 명세(S_t)를 도출합니다.
- 2. 물리적 개입 및 검증, 원인 분석과 부품 주문서 작성 (A):** 권한 범위 내에서 장비를 정지시킨 후, 센서 데이터(RPM=0)를 확인하는 콜백을 수행하여 실제 상태 변화를 확인합니다. 또한, 사고 원인을 분석하고, 필요한 교체 부품을 파악, ERP 시스템의 API를 호출하여 부품 구매 요청서를 작성합니다.
- 3. 가치 승인 및 지식 자산화 (K_{t+1}):** 에이전트가 작성한 원인 분석 보고서와 ERP 부품 주문 요청 건을 현장 관리자가 검토 후 승인(Confirm)하면, 이 조치 이력은 유효한 지식(K_{t+1})으로 자산화되어 다음 실행의 기반이 됩니다.

이 사례는 에이전트의 신뢰성이 모델의 지능(IQ)이 아닌, **실행과 검증의 반복을 통해 불확실성을 해소하는 구조적 루프**에서 비롯됨을 보여줍니다. 시스템은 식 (1)와 같이 누적된 지식 자산(K)을 통해 요구사항(R)에 점진적으로 도달하게 됩니다.”

$$\lim_{t \rightarrow T} \{S_t, K_{t+1}\} \vdash R \quad (1)$$

5. 결론: AI 에이전트의 산업적 확산을 위한 공학적 토대

AI 에이전트 시대의 소프트웨어공학은 “어떻게 불확실한 범용 지능을 구조화된 세계(World) 안에서 안전하게 안착(Grounding)시켜 산업적 신뢰를 확보할 것인가”에 대한 명확한 해답을 제시해야 합니다.

본고에서 제안한 **에이전틱 문제 프레임(APF)**과 **에이전트 직무 기술서(AJD)**는 유능한 신입사원을 산업 현장의 전문 엔지니어로 변모시키기 위한 최소한의 공학적 명세입니다. 신입사원에게 명확한 미션을 부여하지 않고 필요한 지식(입력)과 평가 기준(출력)을 제공하지 않은 채 발생한 오류는 에이전트의 지능 문제가 아니라, 이를 설계하고 관리하는 **엔지니어 혹은 경영자의 책임**입니다. APF는 도메인 지식(K)을 통해 모호한 요청을 구체적인 **실행 명세**로 전환하고, 실행 결과를 검증하여 다시 지식 자산으로 환류시키는 과정을 통해 비결정론적 지능을 산업 현장의 예측 가능한 컴포넌트로 변모시킵니다.

AI가 기술적 신기함을 넘어 산업 현장의 실질적인 생산력으로 확산되기 위한 선결 조건은 바로 이러한 **컴포넌트화**에 있습니다. 단순히 모델의 내부 지능(IQ)을 향상시키는 접근만으로는 산업이 요구하는 엄격한 신뢰성을 충족시킬 수 없습니다. 대신 에이전트의 상호작용을 공학적으로 설계하고 그 경험을 지식으로 자산화하는 구조적 접근만이, 대규모 미션을 수행하는 신뢰할 수 있는 AI 인력을 구축할 수 있게 합니다.

결국 앞서 살펴본 점진적 수렴의 논리가 보여주듯, 에이전트의 신뢰성은

모델의 똑똑함이 아니라 **반복적인 실행과 검증을 통해 지식을 정제하는 공학적 구조화**에서 비롯됩니다. 매 순간의 명세가 검증된 지식 자산에 의해 지속적으로 교정될 때, AI 에이전트는 필연적으로 신뢰(R)의 지점에 도달하게 됩니다. 이러한 공학적 규율화를 통해 에이전트는 단순한 도구를 넘어 산업 현장의 정당한 구성원으로 자리 잡을 것이며, 이것이 우리가 지향해야 할 진정한 산업 AI의 미래이며, **소프트웨어공학이 AX 시대의 불확실성을 마주하며 세워가야 할 새로운 이정표**가 될 것이라 확신합니다.

References

- [1] Pamela Zave and Michael Jackson. Four dark corners of requirements engineering. ACM Trans. Softw. Eng. Methodol., 6(1):1–30, January 1997. ISSN 1049–331X. doi:10.1145/237432.237434. URL <https://doi.org/10.1145/237432.237434>.
- [2] Michael Jackson. Problem Frames: Analysing and Structuring Software Development Problems. Addison-Wesley, 2001.”



신진연구자 소개

서울시립대학교 유명성 교수



• 유명성 박사

조교수

서울시립대학교 전자전기컴퓨터공학부

famous@uos.ac.kr

주요 약력

- 2025.09 ~ 현재 서울시립대학교 전자전기컴퓨터공학부 조교수
- 2022.03 ~ 2025.08 한국과학기술원 전기및전자공학부 박사 (지도교수: 신승원)
- 2020.03 ~ 2022.02 한국과학기술원 전산학부 석사 (지도교수: 신승원)
- 2013.03 ~ 2020.02 충북대학교 컴퓨터공학과 학사

주요 연구 분야

- 자동화된 소프트웨어 테스트 및 디버깅
- 소프트웨어공학을 위한 인공지능

대표논문

- Myoungsung You, Jaehyun Nam, Minjae Seo, Teajun Park, and Seungwon Shin. "HybridMesh: A Hardware-software Hybrid Approach for Accelerating Service Mesh Ingress." USENIX Symposium on Networked Systems Design and Implementation (NSDI). 2026.
- Myoungsung You, Minjae Seo, Seungwon Shin, and Jaehyun Nam. "RDNet: An RDMA-aware Container Network Interface for Cloud Environments." IEEE Conference on Computer Communications (INFOCOM). 2026.
- Minjae Seo, Myoungsung You, Jaehan Kim, Teajun Park, Seungwon Shin, and Jinwoo Kim. "MUFLER: Secure Tor Traffic Obfuscation with Dynamic Connection Shuffling and Splitting." IEEE Conference on Computer Communications (INFOCOM). 2025.
- Myoungsung You, Jaehyun Nam, Hyunmin Seo, Minjae Seo, Jaehan Kim, Dongmin Choi, and Seungwon Shin. "HardWhale: A Hardware-Isolated Network Security Enforcement System for Cloud Environments." IEEE International Conference on Distributed Computing Systems (ICDCS). 2024.
- Myoungsung You, Jaehyun Nam, Minjae Seo, and Seungwon Shin. "HELIOS: Hardware-assisted High-performance Security Extension for Cloud Networking." ACM Symposium on Cloud Computing (SoCC). 2023.

소개글

안녕하세요. 25년 9월부터 서울시립대학교 전자전기컴퓨터공학부에서

연구와 강의를 하고 있는 유명성입니다. 첫 학기와 방학을 보내고 있는데, 강의 준비와 연구실 세팅, 행정 업무까지 처음 겪는 일들이 한꺼번에 몰려와서, 어떻게 시간이 지나갔는지 잘 기억이 나지 않을 정도로 빠르게 한 학기가 지나간 느낌입니다. 그동안 저는 주로 정보보안과 클라우드 컴퓨팅, 특히 클라우드 환경에서의 네트워크 및 시스템 보안과 관련된 연구를 수행해 왔고, 최근에는 이러한 연구에 소프트웨어공학 기술을 접목하는 방향에도 관심을 가지고 있습니다. 개인적으로는 이름이랑 매칭되어서 기억하기 쉬운 이메일 계정인 famous@uos.ac.kr를 사용하게 되었고, 연구실 이름에 잘 어울리는 security.uos.ac.kr 도메인도 확보하게 되어 소소한 만족감을 느끼고 있습니다. 최근에는 연구실을 상징하는 방패 모양의 로고도 디자인해서 사용하고 있는데, 연구실 홈페이지나 발표 자료에 들어간 로고를 볼 때마다 괜히 조금 더 그렸듯해 보이는 느낌이 들어 혼자 흐뭇해하고 있습니다.



TNS LAB
Trustworthy Networked System

Q. 최근 연구하고 계신 분야에 대하여 소개 부탁드립니다.

저는 주로 클라우드 및 5G 환경에서 동작하는 고성능·고신뢰 네트워크 기능을 설계하고 구현하는 연구를 수행해 왔습니다. 최근의 클라우드 환경에서는 서비스 메시, 컨테이너 네트워크, 가상 네트워크 기능(VNF) 등 다양한 네트워크 기능이 소프트웨어 형태로 구현되는데, 이러한 기능은 단순한 프로그램이라기보다 복잡한 분산 시스템에 가깝고, 성능과 안정성, 보안성을 동시에 만족해야 하기 때문에 구현 난이도가 매우 높습니다. 또 하나의 특징은 이러한 네트워크 기능이 항상 CPU 위에서만 동작하는 것이 아니라, SmartNIC이나 FPGA와 같은 전용 하드웨어에서 함께 동작하는 경우가 많다는 점입니다. 따라서 기능을 설계할 때부터 하드웨어 특성과 병렬 처리 구조를 고려해야 하고, 성능과 정확성을 동시에 만족시키는 것이 중요한 과제가 됩니다. 이처럼 구현이 점점 어려워지다 보니, 자연스럽게 "이 과정을 더 자동화할 수 없을까?"라는 생각을 하게 되었고, 최근에는 소프트웨어공학 기법과 AI 기술을 접목하여 네트워크 기능을 자동으로 생성하는 연구를 진행하고 있습니다. 예를 들어 요구사항이나 정책을 기반으로 네트워크 기능을 자동으로 합성하거나, 기존 구성으로부터 새로운 기능을 자동으로 생성하는 방향을 연구하고 있습니다. 또한 자동으로 생성된 네트워크 기능이 실제로 개발자의 요구사항에 맞게 정확하게 동작하는지 검증하는 연구도 함께 수행하고 있습니다. 이 과정에서는 formal verification 기법을 이용해 동작을 수학적으로 검증하거나, 다양한 unit test를 자동 생성하여 기능을 검증하는 방법을 연구하고 있습니다.

Q. AI 기술의 확산 속에서, 성능과 신뢰성을 함께 고려한 시스템 연구의 핵심 방향은 무엇이라고 보시나요?

최근 시스템 연구에서 중요한 흐름은 크게 세 가지라고 생각합니다.

첫째는 AI를 활용한 자동 생성과 자동 검증 기술입니다. 복잡한 시스템을 사람이 직접 설계하고 구현하는 것은 점점 어려워지고 있기 때문에, AI를 활용하여 시스템 구성 요소나 정책을 자동으로 생성하고 이를 자동으로 검증하는 기술이 중요해지고 있습니다.

둘째는 AI-integrated system, 즉 시스템 자체가 AI와 긴밀하게 결합된 구조입니다. 예를 들어 네트워크 보안 환경에서는 매우 많은 로그와 이벤트가 발생하는데, AI를 활용하여 네트워크 로그로부터 필요한 보안 정책을 자동 생성하거나, 복잡한 네트워크 보안 결정을 자동으로 수행하는 시스템을 만드는 연구처럼 시스템에 AI를 얼마나 잘 통합하는지가 중요합니다.

셋째는 Systems for AI, 즉 AI를 더 잘 동작하게 하기 위한 시스템 연구입니다. 최근 대규모 모델의 분산 학습이나 추론 환경에서는 네트워크가 이미 주요 병목으로 작용하고 있습니다. 따라서 모델 구조와 통신 패턴에 더 잘 맞는 고성능 네트워크 인프라를 설계하는 연구가 중요해지고 있다고 생각합니다. 예를 들어 파라미터 동기화 방식이나 텐서 병렬화 구조에 최적화된 통신 계층을 설계하는 방향이 좋은 연구 주제가 될 수 있습니다.

Q. 향후 연구 방향이나 포부에 대해 말씀 부탁드립니다.

그동안 제 연구는 주로 클라우드 환경을 중심으로 이루어져 왔지만, 앞으로는 이를 5G 및 차세대 네트워크 환경으로 확장하려고 합니다. 최근의 5G 네트워크는 ORAN(Open RAN) 구조나 코어 네트워크의 클라우드 배포가 확산되면서, 전통적인 통신 시스템이라기보다 점점 클라우드 시스템과 유사한 형태로 발전하고 있습니다.

이러한 환경에서는 기존 네트워크 기능을 그대로 사용하는 것이 아니라, 고성능과 고신뢰성을 동시에 만족하는 새로운 형태의 네트워크 기능이 필요합니다. 예를 들어 높은 처리량을 유지하면서도 장애 상황에 빠르게 대응할 수 있는 기능, 기존 네트워크 스택의 취약점이나 성능 병목을 해결하는 기능 등이 중요한 연구 주제가 될 것입니다.

또한 이러한 기능을 사람이 직접 구현하는 대신, AI를 활용하여 설계와 최적화 과정을 자동화하는 방향도 적극적으로 연구하고자 합니다. 자동 생성, 자동 검증, 자동 최적화까지 이어지는 연구를 통해 복잡한 네트워크 시스템을 보다 효율적으로 개발하고 운영할 수 있는 기술을 만드는 것이 목표입니다.

Q. 끝으로 소식을 읽는 분들께 한 말씀 부탁드립니다.

저는 비교적 최근까지는 전통적인 시스템 및 보안 분야 연구를 주로 해왔고, 소프트웨어공학 분야는 비교적 최근에 관심을 갖고 공부하기 시작했습니다. 새로운 분야를 배우는 과정이 쉽지는 않지만, 동시에 매우 재미있고 많은 영감을 얻고 있습니다.

앞으로 소프트웨어공학 분야 연구자분들과 더 많이 교류하면서, 보안과 소프트웨어공학을 함께 결합한 연구를 해볼 수 있으면 좋겠다고 생각하고 있습니다. 두 분야가 만나면 흥미로운 문제들이 정말 많이 나올 수 있을 것 같습니다.

혹시 회기나 청량리 근처에 오실 일이 있다면, 연구 이야기 나누면서 커피 한 잔 같이 하실 수 있으면 좋겠습니다. 앞으로 학회와 연구 현장에서 많은 분들과 교류할 수 있기를 기대합니다.



서울시립대 중앙로



서울시립대 해방터



신진연구자 소개

서강대학교 **최재승** 교수



• 최재승 박사

조교수

서강대학교 컴퓨터공학과

jschoi22@sogang.ac.kr

주요 약력

- 2022.09 ~ 현재 서강대학교 컴퓨터공학과 조교수
- 2022.03 ~ 2022.07 한국과학기술원 사이버보안연구센터 선임연구원
- 2017.03 ~ 2022.02 한국과학기술원 전산학부 박사 (지도교수: 차상길)
- 2015.03 ~ 2017.02 서울대학교 컴퓨터공학부 석사 (지도교수: 이광근)
- 2011.03 ~ 2015.02 서울대학교 컴퓨터공학부 학사

주요 연구 분야

- 정적 분석
- 퍼즈 테스팅
- 기호 실행

대표논문

- M. Park, J. Choi, H. Lee, and T. Kwon, "PAVE: Information Flow Control for Privacy-preserving Online Data Processing Services," in Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Mar. 2025.
- T. Kim, J. Choi, S. Im, K. Heo, and S. Cha, "Evaluating Directed Fuzzers: Are We Heading in the Right Direction?," in Proceedings of the ACM International Conference on the Foundations of Software Engineering (FSE), Jul. 2024
- T. Kim, J. Choi, K. Heo, and S. Cha, "DAFL: Directed Grey-box Fuzzing Guided by Data Dependency," in Proceedings of the 32nd USENIX Conference on Security Symposium (USENIX), Aug. 2023
- J. Choi, D. Kim, S. Kim, G. Grieco, A. Groce, and S. Cha, "Smartian: Enhancing Smart Contract Fuzzing with Static and Dynamic Data-Flow Analyses," in Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Nov. 2021
- J. Choi, K. Kim, D. Lee, and S. Cha, "NtFuzz: Enabling Type-Aware Kernel Fuzzing on Windows with Static Binary Analysis," in Proceedings of the IEEE Symposium on Security and Privacy (S&P), May 2021

소개글

안녕하세요. 이번에 소프트웨어 소사이어티에 합류하게 된 서강대학교 컴퓨터공학과 조교수 최재승입니다. 이렇게 소식지를 통해 인사드릴 기회가 생겨서 기쁘게 생각합니다.



그림 1. 서강대학교 알바트로스탑

저는 서울대학교 컴퓨터공학과에서 프로그래밍 언어 분야를 연구하여 석사과정을 마치고, 이후 카이스트에서 정보보안을 전공하여 박사학위를 받았습니다. 박사과정에 진입하면서 연구 분야를 변경했던 셈인데, 다소 힘들었지만 결과적으로 넓은 시야를 갖게 된 소중한 경험이었습니다. 2022년에 박사과정을 졸업한 후에는 카이스트 사이버보안연구센터 선임연구원을 거쳐 같은 해에 서강대학교에 임용되었습니다.

저는 소프트웨어에 존재하는 보안 취약점을 자동으로 탐지하는 기술을 연구해 왔습니다. 오늘날 소프트웨어는 점점 더 많은 분야에서 활용되고 있으며, 그 규모와 복잡성 또한 빠르게 증가하고 있습니다. 이러한 환경에서 사람이 일일이 코드를 살펴보고 취약점을 찾아내는 것은 점점 어려워지고 있기 때문에, 프로그램 분석과 테스팅 기법을 활용하여 취약점을 자동으로 찾아내는 접근의 중요성이 점점 커지고 있습니다. 최근에는 거대언어모델을 사용한 취약점 탐지 기술이 연구되기 시작하면서, 더욱 흥미워지고 있는 분야라고 생각합니다.

Q. 지금까지 연구한, 또는 최근에 연구하고 계신 주제에 대해 말씀해 주세요.

박사과정에서 집중했던 주제는 정적 분석과 퍼징(fuzz testing) 기법을 결합하여, 프로그램의 구조와 의미를 이해한 후 테스트 입력 생성에 활용하는 기술입니다. 퍼징은 다양한 입력을 자동으로 생성하여 프로그램을 반복 실행함으로써 오류를 찾아내는 효율적인 테스팅 기법이지만, 무작위 입력(random input)을 생성하는 전략이기 때문에 프로그램의 내부 구조를 충분히 고려하지 못한 채 탐색이 이루어질 위험이 있습니다. 정적 분석을 통해 프로그램의 구조와 동작에 대한 정보를 미리 파악하면, 퍼징이 이를 가이드 삼아 더 의미 있는 입력을 생성하고 깊은 코드 경로를 탐색하도록 유도할 수 있습니다.

이러한 접근 방법을 Windows 커널과 스마트 컨트랙트 두 개의 도메인에 적용하여 퍼징 기술을 개발하고 그 실효성을 입증했습니다. S&P 2021에 발표

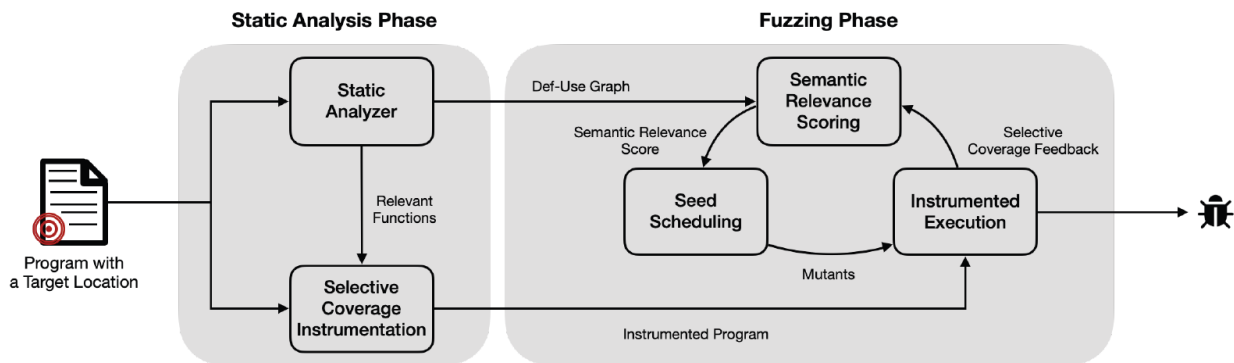


그림 2. DAFL 아키텍처

NtFuzz라는 퍼징 도구는 Windows 바이너리 코드를 분석해서 타입 명세를 추론하고, 이를 바탕으로 커널의 깊은 코드 영역에 도달할 수 있는 시스템 콜 입력을 생성합니다. ASE 2021에 발표한 Smartian은 스마트 컨트랙트의 데이터 흐름 분석을 통해 함수 간의 의존성을 찾아내고, 이를 고려하여 함수 호출 시퀀스를 생성하여 테스트에 활용합니다.

최근에는 이러한 프로그램 이해 기반 접근을 더욱 확장하여, 지향성 퍼징(directed fuzzing) 분야를 집중적으로 연구하고 있습니다. 기존 퍼징은 전체 경로 공간을 폭넓게 탐색하는 데 초점을 맞추는 반면, 지향성 퍼징은 관심 있는 특정 코드 위치를 집중적으로 테스트하는 것을 목표로 하기에 새로운 도전 과제를 안고 있습니다. USENIX 2023에 발표한 지향성 퍼징 도구 DAFL은 정적 분석을 통해 목표 지점의 취약점과 의미적으로 연관 있는 지점을 추출하고, 퍼징 피드백을 정교하게 수집하는 데 활용합니다. 또한 이 연구를 진행하면서 지향성 퍼징 기술의 성능을 정밀하게 평가할 때 발생하는 어려움과 빠지기 쉬운 함정들을 발견하였는데, 이를 정리하여 공유하는 논문을 FSE 2024에 발표하기도 했습니다.

Q. 소프트웨어 보안 연구에서, 정확성과 효율성 사이의 균형은 어떻게 찾고 계신지 궁금합니다.

소프트웨어 취약점을 탐지하는 연구를 하다 보면 항상 겪게 되는 고민 중 하나입니다. 제 연구들은 대부분 정적 분석으로 알아낸 정보를 활용하여 최종적으로는 테스트(퍼징)를 시행하기 때문에, 찾아낸 오류의 오탐(false positive) 문제로부터는 비교적 자유로운 편입니다. 하지만 정적 분석이 프로그램의 행동을 안전하게 추정하는(over-approximate) 과정에서 결과가 부정확해지면, 퍼징 모듈에 잘못된 정보가 전달되어 취약점 탐지 성능을 오히려 떨어뜨릴 수도 있기에, 정적 분석을 정확하게 만드는 것이 중요한 과제 중 하나입니다.

연구 수행의 초기 단계에서는 보통 정확한 분석이 이뤄지도록 분석기를 정교화하는 것에 집중하는 편입니다. 그러다 간단한 벤치마크에 대해 어느 정도 만족스러운 결과가 나오면, 그때부터 분석의 효율성을 개선하여 큰 벤치마크에 적용하기 위한 방법을 모색합니다. 한 가지 흥미로웠던 것은, 가끔은 효율성의 개선과 정확도의 향상이 동시에 이뤄지는 경우도 있습니다. 불필요한 정보를 걸러가며 더 정교하게 분석을 수행한 결과, 분석 시간도 짧아졌던 것이죠. 이런 점에서 정확성과 효율성은 항상 상충하는 것은 아니고, 연구 내내 그 균형을 염두에 두고 고민해야 하는 문제라고 생각합니다.

Q. 향후 연구 방향이나 포부에 대해 말씀 부탁드립니다.

오늘날 대부분의 연구자가 그렇겠지만, 저도 거대언어모델(LLM)을 어떻게 취약점 탐지에 활용할 수 있을지 탐구하는 데 가장 큰 관심을 갖고 있습니다. 언어모델이 가진 코드 이해 능력이 취약점 자동 탐지에 어떤 역할을 할 수 있을지 여러 방면으로 탐구해 보고 있습니다. 지금까지 관찰해 본 바로는, 거대언어모델이 모든 문제를 완벽하고 안정적으로 해결할 수 있는 것은 아니며, 여전히 한계가 존재한다고 생각합니다. 따라서, 전통적인 접근 방법에 비해 언어모델의 유연함이 강점을 발휘할 수 있는 좋은 문제를 찾아내고, 이를 기존 기술과 결합하는 방향을 연구하고 있습니다. 미래에 거대언어모델이 어디까지 발전하고 기존의 기술을 대체할지, 버튼 하나만으로도 복잡한 프로그램의 취약점을 찾아낼 수 있는 날이 올 수 있을지는 저 역시 흥미롭게 지켜보고 있습니다.

또 하나 주목하고 있는 변화는, 점점 더 많은 코드가 사람이 아니라 인공지능에 의해 생성되고 있다는 점입니다. 이러한 추세는 프로그래밍의 근본적인 패러다임을 변화시키고 있다고 생각합니다. 앞으로는 거대언어모델이 안전한 코드를 생성하도록 학습시키는 문제나, 인공지능이 생성한 코드를 효과적으로 분석하고 테스트하여 잠재적인 취약점을 조기에 발견하는 기술 등이 중요해질 것으로 생각합니다. 나아가, 프로그래밍 언어 자체도 우리가 생각하지 못했던 형태로 진화할 수 있다고 생각합니다.

제 포부는 이러한 급변하는 흐름 속에서, 프로그램 분석과 자동화된 테스트 기술이 소프트웨어공학에서 할 수 있는 역할을 모색하고, 그 가능성을 확장해 나가는 것입니다. 특히 거대언어모델이 가진 코드 이해 능력과 기존 분석 및 테스트 기술이 서로 보완적으로 작동할 수 있는 지점을 찾고, 다양한 접근 방법의 시너지를 통해 더욱 실용적인 취약점 탐지 기술을 만드는 방향으로 연구를 이어가고자 합니다.

Q. 끝으로 소식지를 읽는 분들께 한 말씀 부탁드립니다.

소프트웨어가 사회의 거의 모든 영역에서 활용되고 있는 오늘날, 소프트웨어 보안과 소프트웨어공학은 우리 모두가 함께 고민해야 할 중요한 주제가 되고 있다고 생각합니다. 특히 인공지능의 발전과 함께 소프트웨어공학의 지형도 빠르게 변화하고 있으며, 앞으로 여러 흥미롭고 도전적인 문제들이 계속 나타날 것이라고 기대하고 있습니다. 저 역시 연구를 통해 이러한 문제를 조금이라도 더 잘 이해하고, 실제로 도움이 되는 기술을 만드는 데 기여하고자 합니다. 앞으로 소프트웨어 소사이어티의 여러 연구자 분들과 교류하며 다양한 관점과 경험을 나눌 수 있도록 하겠습니다. 잘 부탁드립니다. 감사합니다.



AI에게 ‘형광펜’을 쥐여주다: 코드의 수정 가능성을 활용한 정밀한 취약점 탐지 기술



• 남도하 석사과정
한국과학기술원 전산학부
waroad@kaist.ac.kr

본 원고에서는 지난 2025년 11월 서울에서 열린 ASE 2025 (IEEE/ACM International Conference on Automated Software Engineering)에서 발표된 ‘LOSVER: Line-Level Modifiability Signal-Guided Vulnerability Detection and Classification’ 논문(이하 LOSVER)을 소개하고자 한다.

소프트웨어 취약점(Software Vulnerability)은 종종 단순한 시스템 오류를 넘어 사회적 재난으로까지 이어진다. 최근 발생한 SKT 해킹 사건과 같은 대규모 보안 침해 사고들은 코드 결함이 얼마나 막대한 피해를 초래할 수 있는지 여실히 보여주었다. 그렇기에 소프트웨어를 잘 검수하고 리뷰하는 것이 정말 중요하데, 나날이 커지는 소프트웨어의 규모와 복잡도는 이를 정말 어렵게 한다. 사람이 일일이 코드를 확인하는 것은 정말 많은 노력을 필요로 하기에, 인공지능을 활용한 자동 소프트웨어 취약점 탐지 연구가 그 어느 때보다 활발히 진행되고 있다.

이러한 연구의 중심에는 사전 학습된 언어 모델(Pre-trained Language Model, PLM)이 있다. 우리에게 익숙한 GPT나 Gemini 같은 거대 언어 모델(LLM)이 범용적인 지식을 다룬다면, CodeBERT나 Unixcoder와 같은 PLM은 특정 도메인에 특화가 가능한 모델이다. 이들은 LLM에 비해 모델 크기가 작아 미세 조정(Fine-tuning)하기가 용이하며, 코드의 문법적·의미적 특성을 깊이 있게 학습할 수 있어 취약점 탐지 분야에서 널리 활용되고 있다.

본 연구팀이 제안한 LOSVER(Line-level mOdifiability Signal-guided VulnERability analyzer)는 이러한 PLM 기반 탐지 기술을 한 단계 발전시킨 프레임워크다. LOSVER는 코드 내에서 미래에 ‘수정될 가능성(Modifiability)’이 높은 위치를 예측하고, 이를 취약점 탐지의 결정적인 힌트로 활용한다. 이 연구는 그 독창성과 실용성을 인정받아 이번 ASE 2025에서 ‘ACM SIGSOFT Distinguished Paper Award(최우수 논문상)’를 수상하는 영예를 안았다. 이 글에서는 LOSVER가 탄생하게 된 배경과 핵심 아이디어, 그리고 연구 과정에서의 고민을 공유하고자 한다.

1. 연구의 시작: ‘더 많이’에서 ‘더 자세히’로의 전환

이 연구의 시작은 아주 단순하면서도 본질적인 질문에서 출발했다. ‘어떻게 하면 PLM이 입력된 코드를 더 잘 이해하게 만들 수 있을까?’

초기에는 PLM이 가진 고질적인 문제인 ‘입력 길이 제한(Input Token Limitation)’에 집중했다. CodeBERT나 Unixcoder와 같은 모델들은 한 번에 처리할 수 있는 토큰 수가 512개 혹은 1,024개로 제한되어 있어, 긴 함수가 입력되면 뒷부분이 잘려 나가는 문제가 있었다. 처음에는 이 문제를 해결하기 위해

슬라이딩 윈도우(Sliding Window) 기법 등을 적용하며 입력 길이를 늘리는 시도를 했고, 이 과정에서 얻은 중간 결과를 정리해 KCSE 2025에 투고하기도 했다.

물론 입력 길이를 늘려 더 넓은 문맥을 포괄하는 것도 여전히 중요하고 도전적인 연구 주제이다. 하지만 이번 연구에서는 물리적인 길이를 늘리는 복잡한 문제와 씨름하기보다, 조금 다른 각도에서 접근해 보고자 했다.

‘입력 길이 제한을 해결하기 어렵다면, 주어진 한계 내에서라도 모델이 내용을 더 밀도 있게 이해하도록 도울 수는 없을까?’

이러한 질문은 연구의 초점을 자연스럽게 ‘양(Quantity)’에서 ‘집중(Attention)’으로 옮기는 계기가 되었다. 단순히 많은 코드를 넣으려 애쓰기보다는, 보안상 ‘중요한 코드’를 선별해 모델이 그곳에 집중하도록 유도하는 방식을 택한 것이다. 이는 곧 기존 연구들이 코드의 모든 부분을 균일하게 처리(Uniform processing)하여, 정작 취약점과 직결된 중요한 신호를 놓치고 있다는 문제의식으로 이어졌다.

2. 핵심 아이디어: AI에게 쥐여준 ‘형광펜’

우리는 사람이 코드를 리뷰할 때의 방식에 주목했다. 보안 전문가는 코드를 처음부터 끝까지 기계적으로 읽지 않는다. 대신, 최근에 수정되었거나 버그가 발생할 것 같은 ‘의심스러운 부분’에 집중한다. LOSVER는 이러한 사람의 인지 방식을 모방하여 AI에게 일종의 ‘형광펜’을 쥐여주는 것과 같다.

취약점이라는 것은 사실 함수 전체가 문제라기보다, 극히 일부 라인에서 발생하는 경우가 많다. 아래 [그림 1]의 예시를 보자.

```
176 176 static void emulated_push_error(EmulatedState *card, uint64_t code)
177 177 {
178 - EmulEvent *event = (EmulEvent *)g_malloc(sizeof(EmulEvent));
178 + EmulEvent *event = g_new(EmulEvent, 1);
179 179
180 180 assert(event);
181 181 event->p.error.type = EMUL_ERROR;
182 182 event->p.error.code = code;
183 183 emulated_push_event(card, event);
184 184 }
```

그림 1. 단일 라인 취약점으로 인해 발생한 안전하지 않은 메모리 할당 예시

위 그림에서도 볼 수 있듯이, 함수 전체 코드 중 g_malloc을 사용하는 딱 한 줄이 메모리 할당 취약점의 원인이다. 하지만 기존 모델들은 소스 코드 전체를 동일한 가중치(Weight)로 처리하기 때문에, 정작 문제를 일으키는 이 결정적인 한 줄을 놓치는 경우가 빈번했다. 특히 파인 튜닝 과정에서 단순히 코드 전체와

정답지(Vulnerable/Clean)만 주고 '자, 이 함수는 취약해. 특성을 파악해'라는 식으로 학습시키면, 모델은 함수 내부의 수많은 줄 중에서 어디가 진짜 문제인지 스스로 찾아내야 하므로 학습 효율이 매우 떨어진다.

우리는 입력 코드에서 모델이 '주목해야 할 영역'을 먼저 정의하고, 해당 부분에 모델이 더 집중할 수 있게 유도한다면 성능이 향상될 것이라는 가설을 세웠다. 그리고 이 주목해야 할 영역을 '수정 가능 라인(Modifiable Line)'으로 정의했다.

이 개념은 데이터셋에 포함된 모든 함수의 커밋 히스토리(Commit History)에서 착안했다. 취약한 함수는 당연히 취약점을 패치하기 위해 특정 라인이 수정된다. 반면, 취약하지 않은 깨끗한 함수라 하더라도 코드의 불완전함이나 복잡성으로 인해 추후 수정이 발생하곤 한다. 우리는 이러한 '수정 가능성(Modifiability)'이 곧 코드 내에서 불안정하거나 주의 깊게 봐야 할 지점을 나타내는 강력한 신호라고 판단했다. 따라서 학습 단계에서는 데이터셋의 커밋 내역을 분석하여 실제 수정된 라인(Modified Lines)을 정답 데이터로 활용했다.

이러한 논리를 바탕으로 LOSVER는 크게 두 단계로 작동한다.

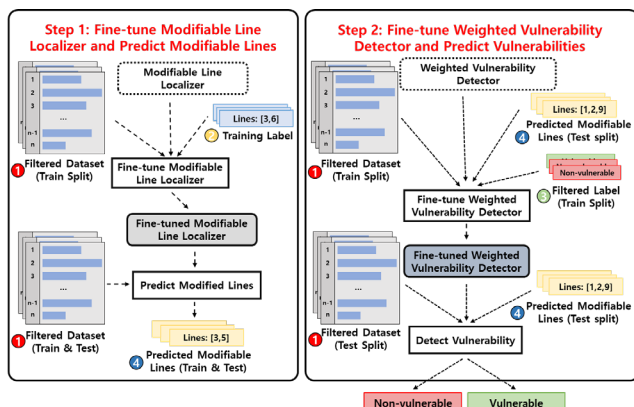


그림 2. 취약점 탐지를 위한 LOSVER 프레임워크 개요

1단계 (Modifiable Line Localizer): 실제 테스트 환경에서는 미래에 코드가 어떻게 수정될지 알 수 없다. 따라서 우리는 먼저 모델이 '미래에 수정될 가능성이 높은 라인(Modifiable Lines)'을 예측할 수 있도록 PLM을 파인 튜닝 한다. 이를 통해 새로운 함수가 입력되더라도, 모델은 과거의 패턴을 바탕으로 불안정한 라인을 스스로 찾아낼 수 있게 된다.

2단계 (Weighted Vulnerability Detector): 취약점을 탐지하는 2단계 모델 역시 1단계와 마찬가지로 PLM을 기반으로 파인 튜닝을 거친다. 이때, 단순히 코드만 학습하는 것이 아니라 1단계에서 예측된 수정 가능 라인(Modifiable Lines) 정보를 학습 과정에서부터 적극적으로 활용한다. 구체적으로는 모델 내부적으로 소스 코드 전체의 정보를 취합하여 하나의 확률값을 계산할 때, 예측된 라인에 해당하는 정보에는 다른 라인 대비 5배의 가중치를 부여하게 된다. 이를 통해 모델의 최종 판단이 해당 라인에 더 크게 의존하도록 설계했다. 즉, 시에게 '이 줄이 수상하니 여기를 더 꼼꼼히 봐'라고 명시적인 힌트(Guidance)를 주는 셈이다.

실험 결과, 이러한 직관적인 접근은 매우 효과적이었다. 대표적인 취약점 탐지 벤치마크인 Devign 데이터셋에서 LOSVER는 기존 베이스라인 모델인 UniXcoder 대비 정확도를 약 4%포인트 향상시켰다. 또한, 단순히 취약점 유무를 판단하는 것뿐만 아니라, CWE(Common Weakness Enumeration) ID를 분류하는 다중 클래스 분류 문제에서도 F1 점수를 2%포인트 이상 높이는 성과를 거두었다.

당연하게도, 라인 레벨의 예측 정확도가 높아질수록 최종 탐지 성능 또한 비례해서 상승하는 경향을 보였다. 이는 우리 프레임워크가 예측된 라인 정보에 얼마나 깊이 의존하고 있는지를 보여주는 동시에, 향후 발전 가능성을 시사한다. 만약 결함 위치 추정(Fault Localization) 분야의 연구가 더욱 성숙해져 1단계의 예측 정확도가 높아진다면, 우리의 가치치 기반 탐지기(Weighted Vulnerability Detector) 역시 별도의 구조 변경 없이도 더 높은 성능을 달성할 수 있을 것이다. 이는 LOSVER가 단독 기술로서뿐만 아니라, 다른 연구 분야의 발전과 함께 성장할 수 있는 유연한 프레임워크임을 의미한다.

3. 연구를 돌아보며

LOSVER 연구는 '입력값을 다 넣지 못한다면, 중요한 것만이라도 확실히 보자'라는 지극히 현실적인 타협점에서 시작했지만, 결과적으로는 취약점 탐지 기술의 정확성을 한 단계 끌어올리는 의미 있는 계기가 되었다. 우리가 제시한 '라인 레벨 신호 기반의 가이던스(Line-Level Modifiability Signal-Guided)' 기술은 향후 금융, 국방, 자율주행 등 소프트웨어의 안전성이 필수적인 분야에서, 개발자들이 미처 발견하지 못한 보안의 구멍을 메우는 데 실질적인 기여를 할 수 있을 것이다. 더 나아가, 이 아이디어는 단순히 취약점 탐지에 그치지 않고 자동화 코드 리뷰나 결함 예측 등 코드의 특정 부분에 대한 집중(Attention)이 필요한 다양한 소프트웨어공학 분야로 확장될 수 있을 것으로 기대된다.

개인적으로, 이번 연구가 필자의 첫 해외 학회 도전이었기에 더욱 각별하다. 논문의 구성부터 영문 작성까지 모든 과정이 처음이라 서툴고 막막한 순간이 많았다. 이때, 랩 세미나를 통해 날카롭고 애정 어린 피드백을 아끼지 않았던 연구실 동료들—김태현 박사과정, 이선구 박사과정, 홍현선 석사과정, 여인석 석사과정—에게 이 자리를 빌려 깊은 고마움을 전한다. 무엇보다 연구의 방향이 흔들릴 때마다 핵심을 짚어주시고, 끝까지 제자를 믿고 지지해 주신 백종문 교수님의 지도가 있었기에 과분한 결실을 맺을 수 있었던 것 같다. 이 모든 분께 깊은 감사의 마음을 바친다.



ASE 2025 개최의 소감



• 유신 교수
한국과학기술원 전산학부

2025년 11월 16일부터 20일까지 제 40회 국제 자동화 소프트웨어공학 학회 - IEEE/ACM International Conference on Automated Software Engineering - 가 서울에서 진행되었다. ASE의 운영위원회(Steering Committee)가 25년 개최지 입찰에 한국도 참여해보면 어떨겠냐고 제안한 것이 2022년 여름이었고, 여러 고민 끝에 조직위원장(General Chair) 자리를 맡기로 결심하고 ASE 2022 에서 정식으로 입찰을 위한 발표를 한 것이 22년 10월 미시건이었으니, 만 3년 넘게 준비한 일을 마무리한 셈이다. 학회를 마치고 두달 반 남짓 시간이 지났는데, 벌써 까마득한 옛날 일인 것 같은 기분도 들면서 동시에 아직 그것밖에 시간이 가지 않았나 하는 생각도 든다. 다음에 비슷한 기회가 생겨서 국제행사를 유치하시는 분들께 조금이라도 도움이 되기를 바라면서, ASE 2025를 마치고 남은 생각을 정리해본다.

학술대회의 규모와 형식

이전에도 소식이 지면을 빌어 학술대회의 미래에 대한 이런 저런 고민을 풀어놔왔는데, ASE 2025는 다양한 고민이 미래의 것이 아니라 바로 눈 앞에 닥쳤고 있음을 경험하게 해주었다. 무엇보다 주최하는 입장에서 반갑지만 동시에 두려웠던 것은 어마어마하게 커진 학회의 규모였다. 일단 접수된 논문 수가 엄청나게 증가했다. Program Chair들은 논문 접수 마감일 3일 전에 이미 110 편이 접수된 것을 보고 마음의 준비를 하긴 했지만, 마감까지 이 숫자가 10배 증가할 줄은 상상도 못했다고 한다. 최종적으로 1,181편의 논문이 접수되었고, Program Committee는 이를 위해 부랴부랴 Area Chair 8명, Program Committee 위원 71명을 추가로 모집했다. 전체적인 논문 심사가 매끄럽게 진행된 것은 전적으로 Program Chair인 Marcel Böhme (MPI S&P, Germany)과 Lingming Zhang (UIUC, USA), 그리고 갑자기 늘어난 논문 양에도 불구하고 심사를 제 시간 안에 마쳐준 심사위원들 덕분이다. 이전과 비교도 할 수 없는 규모때문에 도입한 새로운 기술 중 하나는 Toronto Paper Matching System(TPMS)이다. TPMS는 기계학습 기법을 이용해 심사위원 자신의 논문과 접수된 논문 사이의 의미적 유사도를 바탕으로 자동으로 심사위원에게 논문을 배정해준다. 각 심사위원은 천여편의 논문에 대해 본인의 심사 선호도를 입력하는 대신, 이미 자동으로 입력된 점수를 약간 손보지만 하면 된다. 구체적인 몇몇 경우에 자동으로 제안된 점수가 마음에 들지 않는다는 의견이 있기는 했지만 천 편이 넘는 논문을 300명이 넘는 심사위원들에게 배정하기 위해서는 어쩔 수 없는 선택이었다. 참고로 TPMS는 일정한 사용료를 받고 유료로 제공하는 서비스이다.



그림 1. 개최식 시작 직전 Program Chair인 Lingming Zhang, Marcel Böhme와 함께

천 편이 넘는 논문 심사라는 첫 번째 관문을 넘고 나니, 이번에는 실제 참가자가 수가 얼마나 될 것인지가 걱정이었다. 2024년에 세크라멘토에서 열린 ASE 2024의 전체 참가인원은 500명 아래였고, ASE 2025 조직위원회 내부에서는 600명이 오면 성공이지 않을까 생각했는데, 막상 조기등록 마감일이 가까워오자 등록한 사람의 숫자는 800명을 훌쩍 넘어갔다. 우선 식사는 너무 붐비지 않고 할 수 있을지, 만찬장에 자리가 넉넉할지가 걱정되었고, 만약 1천명도 훌쩍 넘어갈 것 같으면 등록을 마감해야 하는지도 고민이었다. 다행히 조기등록 마감일이 지나가자 등록 인원 증가 추세는 한 풀 꺾였고, 1천명이 넘는 역대 최대 인원으로 - 하지만 준비한 공간 안에서 큰 불편함은 없이 - 학술대회를 진행할 수 있었다.

ASE 2025의 규모가 안겨준 마지막 고민은 프로그램 형식에 대한 것이었다. 총 다섯 개 트랙에 게재가 확정된 논문을 모두 합하면 426편이라는 엄청난 숫자였다. 점점 짧아지는 구두 발표 시간 대신 포스터를 적극 활용해보겠다고 운영위원회에 큰소리(?)를 쳤는데, 막상 3일에 걸쳐 426편의 포스터를 게재한다고 생각하니 시간도 공간도 부족했다. 우선 부랴부랴 호텔 측에 사정해서 포스터 세션을 진행할 커다란 홀을 추가로 계약했다. 다음은 실제 프로그램 구성이 문제였다. 포스터 앞에서 충분한 시간을 보내야 새로운 실험의 의미가 있는데, 오후에는 모두가 포스터 세션에만 오게 하려고 보니, 준비한 공간을 가지고는 논문 한 편에 채 10분도 배정할 수 없었다. 결국 무거운 마음으로 포스터 세션과 일부 세션을 오후에 병행하는 프로그램을 짰고, 논문 한 편당 10분의 발표 시간을 확보할 수 있었다. 근처에서 커피와 다과를 계속 제공하는 등의 여러 가지 방법을 동원해 포스터 앞으로 모으려고 노력했는데, 참여한 사람들의 반응은 긍정적인 것 같았다.

손님 맞이의 즐거움

11월 16일이 되자, 발표 자료를 취합해서 방마다 설치한 노트북 컴퓨터에 다운로드하고, 다음 날 포스터 세션에 필요한 포스터를 전날 밤에 미리 걸어 놓아야 하는 등 여러가지 일이 밀려와서 눈코 뜰 새 없이 바빠졌다. 하지만 동시에 이제 정말 ASE 2025가 궤도에 올라 움직이기 시작했다는 흥분된 마음도, 상상만 했던 프로그램이 눈 앞에 직접 구현되는 것을 지켜보는 뿌듯함도 있었다. 큰 틀에서 봤을 때는 사소한 일일지라도 고민해서 준비한 무언가를 참석자들이 알아봐주면 기분이 좋을 수 밖에 없다. 리셉션과 만찬 장소 앞에 놓은 열음 조각은 호텔에서 선물로 제공해준 것인데 다들 너무나 좋아해서 기억에 남는다. 리셉션에 초대된 라이브코딩 아티스트 박다희님의 공연을 본 다음에 자기도 비슷하게 코딩으로 음악을 만든다며 테이블로 찾아와 자랑한 학생들, 만찬에서 노래한 장서윤님이 한 소절 부르자마자 대단한 가수임을 한 눈에 알아봤다고 말해준 또 다른 학생들이 있었다. 40주년을 기념해 자기가 가진 가장 오래된 ASE 학회 기념품을 가져오면 상품을 주겠다고 선언한 뒤에 아무도 아무것도 가져오지 않으면 어떡하나 걱정했는데, 학회 논문집을 담아서 나눠줬던 USB 메모리부터 학회 티셔츠까지 가져 와 접수해준 많은 참가자들이 있었다. 점심식사 자리가 “부족하면 어떻게 하나 많이 걱정했는데, 짬뽕계 발표장 벽을 타고 의자를 옮겨 공간을 마련해주신 호텔 직원분들이 있었다. 자기 일이 아님에도 밤 늦게 포스터를 붙이는 자원봉사자들을 보고 지나가던 발걸음을 멈추고 함께 도와주던 참가자들이 있었다. 공간도 부족한데 이런 걸 괜히 만들었나 하고 생각했던 기도실과 조용한 방(Quiet Room)에 잠깐 들러봤더니 시끌벅적한 학회장에서 잠깐 벗어나 휴식을 취하고 있는 사람들이 있었다.

여러 사람들이 성공적인 개최를 축하한다고 인사를 건네주었지만, 가장 기억에 남는 것은 일면식도 없고 (미안하지만) 이름도 기억하지 못하는 참가자에게서 들은 인사였다. 포스터 홀에서 원가를 하고 있는데, 지나가던 누군가가 다가와서

채식은 물론이고 할랄, 無글루텐 등 대여섯가지의 식사를 제공하는 학회는 처음 봤다며 쉽지 않은 일이었을텐데 정말 고맙다고 말해주고 사라졌다. 기본 마음으로 호텔측에 감사인사를 전해드렸다. 조직위원회의 다른 분들께서도 모두 같은 마음으로 일해주셨기 때문에 가능했던 것이겠지만, 행사를 성공적으로 주최하는 비결은 참가자들이 최대한 즐겁고 유익한 시간을 보내고 가기를 바라는 마음으로 매사를 결정하는 것임을 다시 한 번 배웠다.

앞으로의 학술대회

ASE 2025 운영을 마치고 나니 시원섭섭한 마음과 함께 학술대회란 대체 무엇인가, 무엇이어서 하는가 하는 생각을 떨치기 어렵다. 나는 참가자의 입장에서 학술대회 참여를 힘들다 생각하지 않고 좋아하는 편인데, 주최 하는 입장이 되어 대형 학술대회 개최에 얼마나 많은 사람의 시간과 노력이 들어가는지를 직접 경험하고 나니 조금은 어깨가 무거워지는 기분이다. 과연 우리는 수많은 사람들이 전 세계에서 비행기를 타고 날아와 모여 며칠 동안 지내다 갈만큼의 가치를 만들고 있을까? 생성형 모델의 등장과 연구공동체의 엄청난 성장으로 인해 동료 심사(Peer Review)가 견딜 수 있는 한계가 머지 않았다는 전망이 쏟아져나오고 있다. 매 년 돌아오는 마감일에 맞춰 논문을 제출하고, 어딘가에 가서 발표를 하고 돌아온다는 익숙한 1년 단위의 일상이 더 이상 당연하지 않은 않을 수도 있는 시대가 된 것이다. 매 년 반복하는 학술대회의 내용을 어떻게 개선할 것인지, 그리고 어떻게 조금이라도 더 지속가능하게 할 것인지를 운영자와 참가자 모두 치열하게 고민해야 할 것 같다. 마침 올해 ICSE 2026에서는 Future of Software Engineering 트랙을 만들어 이런 고민을 나누는 자리를 준비한 것 같다. 소프트웨어공학 공동체가 어떤 아이디어를 내놓고, 어떤 실험을 계속할 것인지 기대가 크다.



그림 2. 포스터 세션이 진행 중인 모습



APSEC 2025 참가 후기



• 이선구

백종문 교수님 연구실, 박사과정, KAIST

작년 12월 초, 2일부터 5일까지 총 4일 동안 진행된 Asia-Pacific Software Engineering Conference (APSEC 2025)에 다녀왔습니다. 아시아 및 태평양 권역의 소프트웨어공학 연구자와 실무자가 한자리에 모여 최신 연구 성과와 현장 문제를 공유함으로써 그 기술적인 깊이와 실무적인 관점을 눈여겨볼 수 있었습니다.

다른 학회들과 마찬가지로, LLM 및 인공지능을 사용하는 연구들이 부쩍 늘어났습니다. LLM을 사용하는 여러 연구들을 들었지만, 인상적이었던 키노트 하나를 소개해드리려고 합니다. 두 번째 날 아침 진행되었던 이 키노트는 Navigating the Impact of Large Language Models on Software Engineering: Challenges and Opportunities라는 제목을 갖고 있습니다. AI4SE, 특히 LLM4SE 쪽 분야에서 사용되는 LLM들의 성능이 낮다는 문제를 제기하고, 어떻게 해결해야 할지에 관해서 논의합니다. 많은 LLM들은 (특히 ChatGPT의 경우), 특히 코드 생성 task에서 코드가 복잡할 때 요청한 코드를 정확히 생성하지 못한다든가, 안전하지 않은 코드를 생성한다든가, 또는 요청하지 않은 내용의 코드를 생성한다든가 등의 문제를 가지고 있습니다. 이는 근본적으로 LLM의 훈련 데이터가 코드뿐만 아니라 다른 도메인의 지식도 가지고 있기 때문에 코드라는 도메인으로 데이터를 정렬하기가 어렵다는 문제를 가지고 있기 때문입니다. 이에, Chain-of-Thought 구조의 프롬프트 형식으로 훈련 데이터를 구성하고, 강화학습 프레임워크를 통해서 새로 모델을 훈련시키는 방법을 제안했습니다. 이 기법으로 state-of-the-art (비록 그 당시지만)의 LLM들과 비교해보았을 때, 코드 생성 정확도가 더 높아졌다고 합니다. 이에 추가로, LLM들의 훈련 과정을 잘 이해하기 위해 뇌공학 분야까지 연구해가면서 데이터에서 패턴을 학습하는 신경망의 강점과 기호적 지식/규칙으로 표현되어 있는 명세와 논리의 neuro-symbolic integration이라는 개념도 흥미로웠습니다.

실제로 학회가 끝나고 몇 개월도 지나지 않은 현재 더 강력하고 새로운 LLM들이 나오고 있는데, LLM들에 대한 활발한 연구로 성능이 더 좋아지는 시대가 머지 않았음을 느낍니다.

학회에 가는 이유로는 연구 트렌드 조사, 연구자들끼리의 네트워킹도 있지만, 학회 장소 근처의 도시 탐방도 이유를 들 수 있습니다.

학회장 자체가 하나의 거대한 호텔 (Wynn Palace) 안에 있었는데, 호텔이 워낙 커서 처음에 들렀을 때 길을 잃을 뻔했습니다. 마치 영화 <나홀로 집에>에 등장하는 호텔같이 생겼는데, 마침 크리스마스 분위기가 내서 그랬던 것 같습니다. 근처에 다른 호텔들도 각각 다른 컨셉을 가지고 있었습니다. 어떤 곳은 런던, 어떤 곳은 베네치아 느낌으로 꾸며냈고, 쇼핑물 내에서 곤돌라를 탈 수 있는 시설까지 마련된 곳도 있었습니다.

마카오가 포르투갈의 지배와 영향을 많이 받았기 때문에, 포르투갈식 식사를 제공하는 식당들이 많이 있었습니다. 특히 호텔은 많은 곳을 빠져나와 조금 북쪽으로 지나가면 타이파 빌리지라고 불리는 소박한 동네가 나오는데, 여기에서 포르투갈식 양고기를 먹었던 것이 기억에 납니다.

APSEC 2025는 연구 트렌드를 확인하는 자리이기도 했지만, 연구가 실제로 “어떤 문제를 풀기 위해 존재하는가”를 다시 생각하게 만든 시간이었다고 느낍니다. LLM4SE의 성장은 분명 빠르지만, 개발자가 믿고 사용할 수 있는 수준으로 가기 위해서는 정확성과 안전성, 그리고 명세 기반의 검증처럼 소프트웨어공학이 오랫동안 다뤄온 가치가 더 강하게 결합되어야 한다는 점이 특히 인상 깊었습니다.

국제 학술대회를 참여해볼 기회가 많이 없었는데, 이번 기회를 통해서 열심히 연구를 진행하여 제 연구를 소개할 수 있는 자리를 가졌으면 합니다.



APSEC 2025 학회

KCSE 2026 개최 후기



• 이주용 교수
UNIST

2026년 2월 4일부터 2월 6일까지 UNIST에서 제28회 한국 소프트웨어공학 학술대회(KCSE 2026)가 개최되었습니다. 제가 조직위원장을, 같은 학교의 김미정 교수님께서 학술위원장을 맡았습니다. 소사이어티 회원 분들을 비롯해 많은 분들께서 참석해 주신 덕분에 대회를 무탈하게 치를 수 있었습니다. 이 글을 통해 학회에 참석하신 분들과는 함께 행사를 되돌아보고, 아쉽게 참석하지 못하신 분들과는 지면으로나마 학회 소식을 전해드리고자 합니다. KCSE에 익숙하신 독자들이 많을 것으로 생각하여, 이번 대회의 특징 위주로 말씀드리겠습니다.

KCSE 2026 특징 1: 장소

이번 대회는 울산과학기술원에서 개최되었습니다. KCSE가 대학에서 개최된 것은 2002년 서울대학교에서 개최된 이래 24년만에 처음입니다. 또한, 스키장 리조트가 아닌 곳에서 개최된 것도 24년만에 처음입니다. 오랜만에 대학에서 열리는 대회인 만큼, 학회 참가자들이 <<지적 유희>>를 느끼실 수 있도록 하는 데에 중점을 두었습니다. 예년보다 많은 기조연설 세션을 배치하여 참가자들이 매일 새로운 각도에서 소프트웨어공학을 바라볼 수 있는 기회를 제공하였고, IITP의 김태호 PM님을 모시고 “AI-Native SW 산업 혁신 기술 개발 사업”을 주제로 공청회를 열어, AI가 가져오는 소프트웨어공학의 변화에 대해 토론하는 시간을 가졌습니다.

이번 대회에서 <<지적 유희>>만을 추구한 것은 아닙니다. 대학은 지성의 장소이자 낭만의 대명사입니다. 참가자들이 낭만도 만끽하실 수 있도록 첫날 저녁 식사는 태화강 국가정원 근처에서 기호에 따라, 꾸꾸미 정식, 중화 요리, 감자탕 중에 원하시는 메뉴를 선택해 드실 수 있도록 하였습니다. 식사 후에는 자유시간을 갖고 태화강국가정원을 산책하거나 근처 카페에서 담소를 나누는 등 여유로운 시간을 가졌습니다.

KCSE 2026 특징 2: 주제

이번 KCSE의 주제는 <<소프트웨어공학 가치의 확산: DevSecOps, MLSecOps, 그리고 그 너머>>였습니다. 이 주제에 맞추어 다음의 세 가지 기조강연이 진행되었습니다.

1. 학회 첫째 날에는 LG 전자 webOS SW 개발그룹 DevOps개발실장이신 박인욱 상무님께서 “AI-Native DevOps: What Should We Do —

Accelerating Software Engineering with Shift-Left, CI/CD, and IDP”를 주제로 기조강연을 해 주셨습니다. DevOps를 비롯한 소프트웨어공학의 가치와 원칙이 현재 현장에서 어떻게 적용되고 있는지 볼 수 있는 좋은 기회였습니다. 특히, AI 기술의 활용이 확대될수록 소프트웨어공학의 기본에 충실한 소프트웨어 개발 프로세스 기준 확립이 중요하다는 취지의 말씀이 인상적이었습니다.

2. 학회 둘째 날에는 보안 분야의 권위자이신 고려대학교 이희조 교수님께서 “보이지 않던 소프트웨어를 투명하게: SBOM과 취약점 관리가 던지는 질문”을 주제로 기조 강연을 해 주셨습니다. 소프트웨어 보안 관리를 SBOM(Software Bill of Materials)을 매개로 어떻게 체계적으로 수행할 수 있는지에 대해 말씀해 주셨습니다.
3. 학회 마지막 날에는 설명가능 AI의 대표적인 연구자이신 KAIST의 최재식 교수님께서 “설명가능 인공지능을 통한 대형 인공지능 모델의 기능 및 안전성 확인”을 주제로 기조 강연을 해 주셨습니다. 전통적인 소프트웨어와 마찬가지로 AI 모델의 품질도 체계적으로 관리될 필요가 있다는 말씀을 해주셨습니다.

아마, 기조 강연을 모두 들으신 참가자분들은 <<소프트웨어공학 가치의 확산>>에 대해 충분히 공감하실 수 있었을 것입니다.

KCSE 2026 특징 3: 조직위원 구성

이번 대회에서는 프로그램의 전문성을 확보하기 위한 노력의 일환으로, 각 트랙별로 별도의 위원장을 배정하였습니다. 학술 위원장을 맡으신 UNIST 김미정 교수님 외에도, POSTECH 배경민 교수님께서 튜토리얼 위원장, 성신여자대학교 강종구 교수님께서 신진연구자 세미나 위원장을 맡아 주셨습니다. 또한, 현재 소프트웨어공학 소사이어티 총무를 맡고 계신 충북대학교 홍신 교수님께서 재무 위원장을 맡아 재무 관리를 담당해 주셨습니다. 마지막으로, UNIST의 김영재 박사과정생과 한동대학교 남재창 교수님께서 웹 위원장을 맡아 학회 웹사이트 관리를 해 주셨습니다. 특히, 남재창 교수님은 역대 KCSE 웹사이트를 취합하여 다음의 포털 링크를 통해 손쉽게 볼 수 있도록 해주셨습니다: <https://kcse.sigsoft.or.kr/>

학회 종료 시기와 소식지 3월호 원고 마감 기한이 거의 붙어있는 관계로, 각 위원장님의 회고를 취합해 신지 못해 몹시 아쉽습니다. 모두 성심을 다해 준비를 해 주신 것에 대해 진심으로 감사드립니다.

KCSE 2026 특징 4: 경품 추천

KCSE의 전통에 따라 이번 대회에서도 폐회식 때에 경품 행사를 진행하였습니다. 다만, 이번에는 경품 내용에 변화를 주었는데요, 제가 선정한 책 6권을 2단계 추천(곧 설명 드리겠습니다)을 통해 나누어 드렸습니다. 재미를 위해 경품 행사 시작 전에는 책의 제목을 공개하지 않았습니다. 첫번째 추천 단계에서는 물렛 프로그램을 실행하여 참석자 6명을 무작위로 선정하였습니다. 두 번째 단계에서는 아래 화면과 같은 질문을 선정된 6명에게 드려, 응답에 따라 책이 골라지도록 했습니다. 예를 들어, 비인물분야와 소프트웨어공학을 고르신 분께는 차성덕 교수님의 소프트웨어공학 이야기 책이 수여되었습니다.

책 고르기 게임

어떤 종류의 책에 관심이 있으신가요?

인물 분야

비인물 분야

책 고르기 게임

어떤 분야에 관심이 있으신가요?

수학

소프트웨어공학

천문학

소회 및 의견

앞서 언급했듯이 이번 KCSE는 24년 만에 처음으로 대학에서 개최되었습니다. 또한, 13년 만에 평창 한화 리조트가 아닌 다른 장소에서 열린 대회이기도 합니다. 오랜 기간 동안 이미 잘 정립해 놓은 학회 개최 프로세스를 벗어나 대회를 준비하는 데에는 용기가 필요했습니다. 하지만 사실, 제 용기보다는 13년만에 한화 리조트에서 “가출”을 결단하신 이정원 회장님과 이사진 분들에게 더 큰 용단과 책임감이 필요했으리라고 생각합니다. 이 지면을 빌려 다시 한 번, 이번 KCSE의 UNIST 개최를 제안해주시고 적극적으로 지원해주신 이정원 회장님을 비롯한 소사이어티 회원 여러분께 감사의 인사를 드립니다.

대회를 준비하면서 제가 세운 원칙은 두 가지였습니다. 첫째, KCSE의 전통을 유지하기. 둘째, 소프트웨어공학 소사이어티의 지속적 발전을 위한 개선을 도모하기. 얼핏 보기에는 이 두 가지 원칙이 상충되어 보일 수도 있습니다. 하지만, 지속적인 발전을 위해서는 튼튼한 틀이 필요한 법입니다. 이번 대회를 준비할 때에도 기존 대회 준비 절차를 튼튼한 뼈대로 활용하였습니다. 다만, 시대 변화에 맞춰 필요하다면 뼈대 안의 내용을 바꾸는 데에는 주저하지 않으려 노력했습니다.

이번 KCSE의 대학 개최가 일회성 이벤트로 끝나지 않기를 바랍니다. 앞으로 평창을 비롯해 다양한 곳에서 KCSE 참석자 분들을 만날 수 있으면 좋겠습니다. KCSE와 소프트웨어공학 소사이어티의 지속적인 발전을 바라며 이만 줄입니다.



KCSE 2026 폐회식



빅데이터 엣지 클라우드 서비스 연구센터 소개

Research Center for Big Data Edge Cloud Services



KAIST BECS **고인영** 센터장



KAIST BECS **지은경** 전임연구원

1. 센터 소개

KAIST 빅데이터 엣지 클라우드 서비스 연구센터(Research Center for Big Data Edge Cloud Services, 이하 BECS센터)는 차세대 컴퓨팅 인프라로 빠르게 부상하고 있는 엣지 클라우드 환경에 특화된 빅데이터 및 서비스 컴퓨팅 핵심 인재 양성을 목표로 2020년 7월 출범하였습니다. 센터의 공식 과제는 “엣지 클라우드에서 고신뢰·고사용성 빅데이터 플랫폼 및 분석·예측 서비스 기술 개발”이며, 연구 기간은 2020년 7월부터 2027년 12월까지입니다. 엣지

클라우드 환경은 센서·단말 등 데이터 발생 지점에 가까운 엣지 노드와 중앙 클라우드가 계층적으로 연결된 분산 인프라로, 지연(latency), 대역폭, 신뢰성 요구를 고려하여 데이터 처리와 서비스 실행을 동적으로 분산·조정하는 컴퓨팅 환경입니다. 우리 센터는 이러한 환경에서 효율적인 데이터 수집, 고성능 분석 및 예측, 그리고 신뢰할 수 있는 서비스 제공을 가능하게 하는 핵심 기술을 개발하고 있습니다. 궁극적으로는 스마트 시티, 스마트 팩토리 같은 다양한 사회·산업 현장에서의 실제 문제를 엣지 클라우드 기반 빅데이터 서비스로 해결하는데 기여하는 것을 목표로 하고 있습니다.

다양한 고신뢰·고사용성 엣지 클라우드 빅데이터 분석 및 예측 응용



그림 1. 고신뢰·고사용성 엣지 클라우드 빅데이터 분석 및 예측 응용 서비스 개요



그림 2. 센터 조직도 (2026년)

센터에는 (2026년 기준) 교수 7명, 박사과정 28명, 석사과정 30명, 학부생 6명, 전임연구원 1명, 산업체 참여 인력 2명 등 총 74명이 함께 연구를 수행하고 있으며, 기술 개발과 더불어 고급 전문 인력 양성에 힘을 쏟고 있습니다.

2. 참여 교수 및 세부과제 소개

BECS 센터는 고신뢰 엣지 클라우드 빅데이터 분석 및 예측 실현을 위해 연구주체별로 역할을 가지고 협력하고 있으며, (주)솔루션링크, (주)모아소프트, (주)데이터스트림즈, 슈어데이터랩(주), (주)구름네트웍스, 한국과학기술정보연구원, (주)자이온아이티에스, (주)스테이피파이브, UC San Diego, Urban Technology Alliance와 협력하고 있습니다. BECS 센터의 연구는 네 개의 세부과제로 구성되어 있으며, 각 과제는 엣지 클라우드 기반 빅데이터 서비스의 서로 다른 핵심 축을 담당하고 있습니다.

[1세부과제] 사용자 의도 기반 직관적 빅데이터 매쉬업 프로그래밍 및 고성능 분산 실행환경 기술 개발

(KAIST 고인영, 박종세, 강지훈(~2025))

1세부과제는 “빅데이터 서비스를 얼마나 쉽게, 그리고 얼마나 효율적으로 만들 수 있는가?”라는 질문에서 출발합니다. 사용자의 의도(intent)를 기반으로 빅데이터 분석 및 예측 서비스를 직관적으로 개발할 수 있는 매쉬업 프로그래밍 언어와 런타임 컴파일러를 개발하고 있습니다. 엣지 클라우드의 다양하고 분산된 자원을 최대한 활용할 수 있도록 실행 환경을 가상화하고, 자원 효율성과 성능

요구사항을 동시에 만족시키는 가속화 및 런타임 시스템 기술을 연구합니다. 또한 사용자 중심의 빅데이터 서비스 검색, 코디네이션, 클라우드소싱 기법을 통해 데이터 수집과 활용을 보다 효과적으로 지원합니다. “고수준에서 프로그래밍을 쉽게 하며, 시스템은 똑똑하게 빠르게 실행하는” 환경을 만드는 것이 핵심 목표입니다.

[2세부과제] 개발자·사용자 관점의 빅데이터 엣지 클라우드 서비스 품질 보증 기술 개발

(KAIST 백종문, 배두환(~2022), 전북대 류덕산)

2세부과제는 서비스의 신뢰성과 품질을 연구합니다. 개발자 관점에서는 빅데이터 엣지 클라우드 서비스의 데이터 활용/관리 계층과 엣지 클라우드 계층 간 상호작용을 고려한 확장성 있는 시험 및 결함 위치 추정 기술을 개발합니다. 사용자 관점에서는 QoS(서비스 품질) 저해 요인을 분석하고, 서비스 배포 이후 발생하는 불확실성에 대응할 수 있는 항시적 QoS 예측 기술을 연구합니다. 이 과제는 “잘 동작하는가?”를 넘어서 “서비스를 지속적으로 믿고 사용할 수 있는가?”를 다루는 연구라고 할 수 있습니다.

[3세부과제] 표상 학습 기반 다차원 스트림 처리·분석·예측 기술 개발

(KAIST 김명호(~2024), 충남대 박지훈)

3세부과제는 엣지 클라우드 환경에서 실시간으로 발생하는 대규모 데이터 스트림을 다룹니다. 엣지 환경에 적합한 빅데이터 스트림 처리 및 예측 기술을 개발하고, 다차원 스트림 분석을 통해 주요 패턴을 추출합니다. 실시간으로

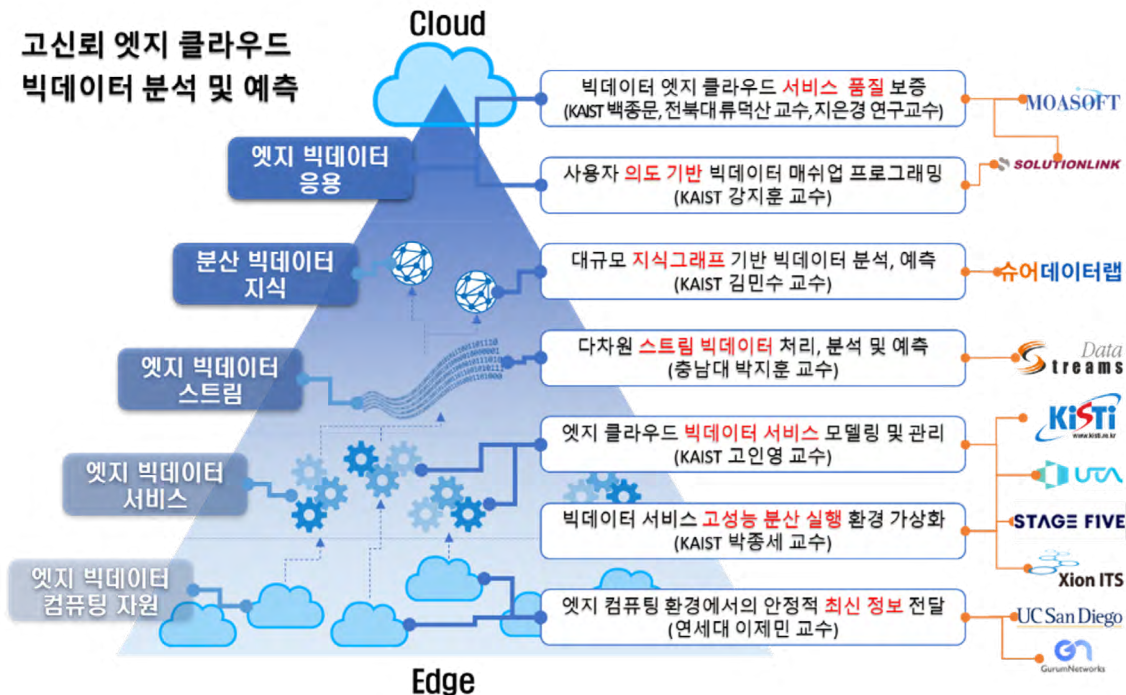


그림 3. 고신뢰 엡지 클라우드 빅데이터 분석 및 예측 실험을 위한 연구주체별 역할

발생하는 데이터를 반영해 예측 모델의 성능을 지속적으로 최적화하는 분산 학습 및 동적 예측 모델 기술도 함께 연구합니다. 끊임없이 흐르는 데이터를 “적시에 가장 잘 이해하고 예측하는” 지능형 분석 기술이 핵심입니다.

[4세부과제] 대규모 이종 그래프 기반 뉴럴 네트워크 분석 및 예측 기술 개발

(KAIST 김민수, 연세대 이재민)

4세부과제는 엡지 클라우드 환경에서 생성되는 복잡한 관계 데이터를 그래프 관점에서 분석합니다. 안정적이고 효율적인 빅데이터 기반 지식 관리를 위해 정보 최신성 분석 기술을 개발하고, 통신 환경의 동적 변화에 적응적인 데이터 전달 기술을 연구합니다. 특히 분산된 대규모 이종 그래프에 대해 정보 최신성을 고려한 실시간 분석이 가능한 GNN(그래프 신경망) 모델을 개발하고 있습니다. 복잡하게 얽힌 관계 속에서 의미 있는 지식을 빠르게 추출하는 기술을 목표로 합니다.

3. BECS 국제 워크샵 개최

BECS 센터에서는 2021년부터 International Conference of Web Engineering (ICWE) 확회와 연계하여 International Workshop on Big data driven Edge Cloud Services (BECS) 국제 워크샵을 매년 개최해 오고 있습니다. 올해 6회째를 맞이하는 BECS Workshop이 프랑스 리옹에서 6월 9일에 열릴 예정입니다. 웹사이트를 참고하시어 관련 분야 논문 투고 및 워크샵 참여에 관심과 참여를 부탁드립니다. <https://becs.kaist.ac.kr/iwbecs2026/>

4. 맺으며

빅데이터 엡지 클라우드 서비스 연구센터는 “쉽게 만들고, 빠르게 실행하며, 믿고 사용할 수 있는” 엡지 클라우드 기반 빅데이터 서비스를 구현하기 위해 플랫폼, 품질, 학습, 그래프 분석까지 전 주기를 아우르는 연구를 수행하고 있습니다. 앞으로도 소프트웨어공학 커뮤니티와 활발히 교류하며, 엡지 클라우드 시대를 선도할 기술과 인재를 함께 만들어가고자 합니다. 소프트웨어공학 소사이어티 여러분들의 관심과 응원 부탁드립니다.

[홈페이지] <https://becs.kaist.ac.kr/>

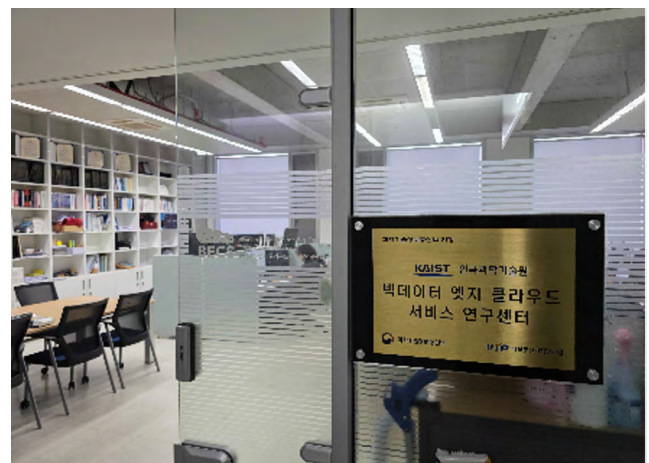


그림 4. BECS 센터 연구실



그림 5. BECS 센터의 정기적 워크샵



그림 5. BECS 센터의 정기적 워크샵



소사이어티 광장

축하합니다!

• 임용

- 김수민 박사 (KAIST, 지도교수: 차상길), 이화여자대학교 컴퓨터공학과 조교수 임용, 2026.03

• 수상

- KSC 2025 최우수논문상 수상 (소프트웨어공학 분야)
 - CommitChrono: 시간적 연속성과 개발자 이력 정보를 활용한 문맥 인식 기반 커밋 메시지 생성 기법, 안도경·양근석(한경국립대학교)
- KSC 2025 우수논문상 수상 (소프트웨어공학 분야)
 - 모듈화 분석 기반 교차 타입을 사용한 Python 정적 타입 분석, 정민경·이성호(충남대학교)
 - 단일 모델 기반 커버리지 최적화 테스트 생성: 브랜치-우선-라인-가드와 퍼즈-메타모픽 결합, 김윤기·양근석(한경국립대학교)
- KSC 2025 우수발표논문상 (소프트웨어공학 분야)
 - 자동 버그 리포트 관리에서 구성 요소 누락이 성능에 미치는 영향, 조혜민·김미수(전남대학교)
 - 안저 이미지로부터 심혈관위험인자를 예측하는 멀티모달 대조학습 프레임워크, 이현민·고성현·추현승(성균관대학교)
 - The Impact of Identifiers and Data Flow on LLM-Based Code Clone Detection, 이세현·최용석(한양대학교)

• 회원등정

- KAIST 유 신 교수 2026년 한국공학한림원 컴퓨팅 분야 신입 일반회원으로 선임
- 한국공학한림원은 10개월간 진행된 엄격한 심사를 거쳐 2026년도 신입회원 명단을 발표했습니다. 최종 확정된 회원은 정회원 49인(산업계 26인, 학계 23인)과 일반회원 84인입니다. 한국공학한림원은 학계와 산업계, 국가 연구기관 등에서 혁신적인 기술 개발과 탁월한 연구 성과를 통해 국가 발전에 크게 기여한 전문가를 발굴하는 국내 최고의 공학 기술 분야 석학 단체입니다. KAIST 전산학부 유신 교수가 소프트웨어 테스트 및 디버깅의 자동화 기술에 대한 기여를 인정받아 2026년 한국공학한림원 컴퓨팅 분야 신입 일반회원으로 선임되었습니다.

소개합니다!

• 책 출간 I



- 도서명: Hands-on Software Architecture: Unified Architecture Process
- 저자: Soo Dong Kim (숭실대학교 소프트웨어학부), Mira Kim (Department of Computer Science, California State University, Fullerton)
- 출판사: Springer / 출간월: 2025년 11월 / 페이지: 517쪽 / ISBN: 978-3-032-01183-1
- Supplementary Website: <https://handsonse.org/>
- 본 도서는 소프트웨어 아키텍처의 이론적 기반과 체계적인 설계 절차를 통합적으로 제시하는 전문 학술서로, Unified Architecture Process(UAP)를 중심으로 실제 적용 가능한 아키텍처 설계 방법과 산출물을 구체적으로 설명합니다. 도서는 다음과 같이 3개 Part로 구성되어 있습니다.
 - Part I. Overview of Software Architecture: 소프트웨어 아키텍처의 정의, 진화, 구성 요소, 설계상의 도전 과제 및 아키텍처의 역할을 체계적으로 설명하고, Unified Architecture Process의 개념과 방법론적 접근을 제시합니다.
 - Part II. Guidelines for Unified Architecture Process: Unified Architecture Process의 6개 핵심 활동(A1~A6)을 Step 및 Task 단위로 세분화하여 구체적으로 설명합니다.
 - A1. Requirements Refinement
 - A2. System Context Analysis

- A3. Schematic Architecture Design
- A4a-A4d. Functional / Information / Behavior / Deployment View Design
- A5. Design for Non-Functional Requirements
- A6. Architecture Evaluation

각 활동에는 체크리스트와 연습문제를 포함하여 교육 및 실무 적용이 가능하도록 구성되었습니다.

• Part III. Resources for Software Architecture Design: Part III에서는 아키텍처 설계를 지원하기 위한 체계적인 참고 자료를 제공합니다.

- 총 22개의 대표적 소프트웨어 아키텍처 스타일을 구조(Structure), 협력 모델(Collaboration Models), 장단점(Strengths and Limitations) 관점에서 설명하였습니다.
- 다양한 아키텍처 평가 방법이 정리되어 있습니다. 시나리오 기반(ATAM, CBAM, SAAM 등), 모델 기반(ADL, UML 등), 형식 기법 기반(Z, Petri Net, Temporal Logic 등), PoC 기반 및 Prototype 기반 평가 방법을 포함합니다.

이를 통해 독자는 시스템 특성에 적합한 아키텍처 스타일을 선택하고, 다양한 평가 기법을 적용하여 설계의 타당성을 검증할 수 있도록 구성하였습니다.

• 책 출간II



- 도서명: AI 시대, 개발자는 어떻게 진화하는가

- 저자: 금창섭

- 출간일: 2026년 2월 25일

- 생성형 AI의 급속한 확산은 소프트웨어 개발 현장의 패러다임을 근본적으로 바꾸고 있습니다. “일주일을 밤낮으로 고민하여 짠 코드를, AI는 커피 한 잔 마실 시간에 이미 구현을 끝냈습니다.” 이 현실 앞에서 개발자의 가치는 코딩 속도가 아니라, 시스템의 구조를 결정하는 **아키텍처 설계 역량**으로 재정의되고 있습니다.

이 책은 “이 시스템은 왜 존재하며, 누구를 위해 무엇을 해야 하는가”라는 질문에서 출발하는 **아키텍처 스토리**를 핵심 개념으로 삼습니다. 설계의 출발점을 코드가 아닌 맥락과 의도에 두고, 그것을 실행 가능한 구조로 전환하는 체계적 방법론을 제시하는 것이 이 책의 가장 큰 특징입니다. 생성형 AI를 단순한 코딩 도구가 아닌 **아키텍처 판단 파트너**로 활용하는 실전 프롬프트 패턴 6가지와, 성능 vs 보안, 일관성 vs 가용성 사이의 트레이드-오프 분석 기법도 구체적으로 다룹니다.

요즘 주목받는 ‘바이브 코딩’의 개발 속도에 견고한 아키텍처 원칙을 결합하면 어떤 결과가 나올까요? 이 책은 그 답을 이커머스 플랫폼 ShopSmart의 A to Z 설계 시뮬레이션을 통해 실증적으로 보여줍니다. 설계와 구현의 간극을 줄이는 ‘워킹 스켈레톤’ 구축까지, 아키텍트로의 전환을 준비하는 개발자에게 실질적인 이정표가 될 것입니다.



2026 국내외 학술대회

Full name	학술대회	대회일자	논문마감	URL	기준 (행사, 제출)
International Conference on Software Engineering	ICSE 2026	2026/04/12~04/18		https://conf.researchr.org/home/icse-2026	행사
Mining Software Repositories Conference	MSR 2026	2026/04/13~04/14		https://2026.msrrconf.org/	행사
International Conference on Software Testing, Verification and Validation	ICST 2026	2026/05/18~05/22		https://conf.researchr.org/home/icst-2026	행사
Programming Language Design and Implementation	PLDI 2026	2026/06/15~06/19		https://pldi26.sigplan.org/	행사
Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering	FSE (ESEC/FSE) 2026	2026/07/05~07/09		https://conf.researchr.org/home/fse-2026	행사
Software Engineering Education and Training	CSEET 2026	2026/07/20~07/22		https://cseet26.techconf.org/	행사
Requirements Engineering Conference	RE 2026	2026/08/17~08/21		https://conf.researchr.org/home/RE-2026	행사
International Conference on Software Maintenance and Evolution	ICSME 2026	2026/09/14~09/18	2026.03.06	https://conf.researchr.org/home/icsme-2026	행사 및 제출
Conference on Object-Oriented Programming, System, Languages, and Applications	OOPSLA 2026	2026/10/03~10/09	2026/03/17 (Round 2)	https://conf.researchr.org/track/splash-2026/oopsla-2026	제출
International Conference on Automated Software Engineering	ASE 2026	2026/10/12~10/16	2026.03.26	https://conf.researchr.org/home/ase-2026	제출
International Conference on Model Driven Engineering Language and Systems	MODELS 2026	2026/10/04~10/09	2026.03.27	https://conf.researchr.org/home/models-2026	제출
International Symposium on Software Reliability Engineering	ISSRE 2026	2026/10/20~10/23	2026.04.17	https://cyprusconferences.org/issre2026/	제출
Asia-Pacific Software Engineering Conference	APSEC 2026	2026/12/07~12/10	2026/07/13	https://conf.researchr.org/home/apsec-2026 (개설 예정)	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)

문의처

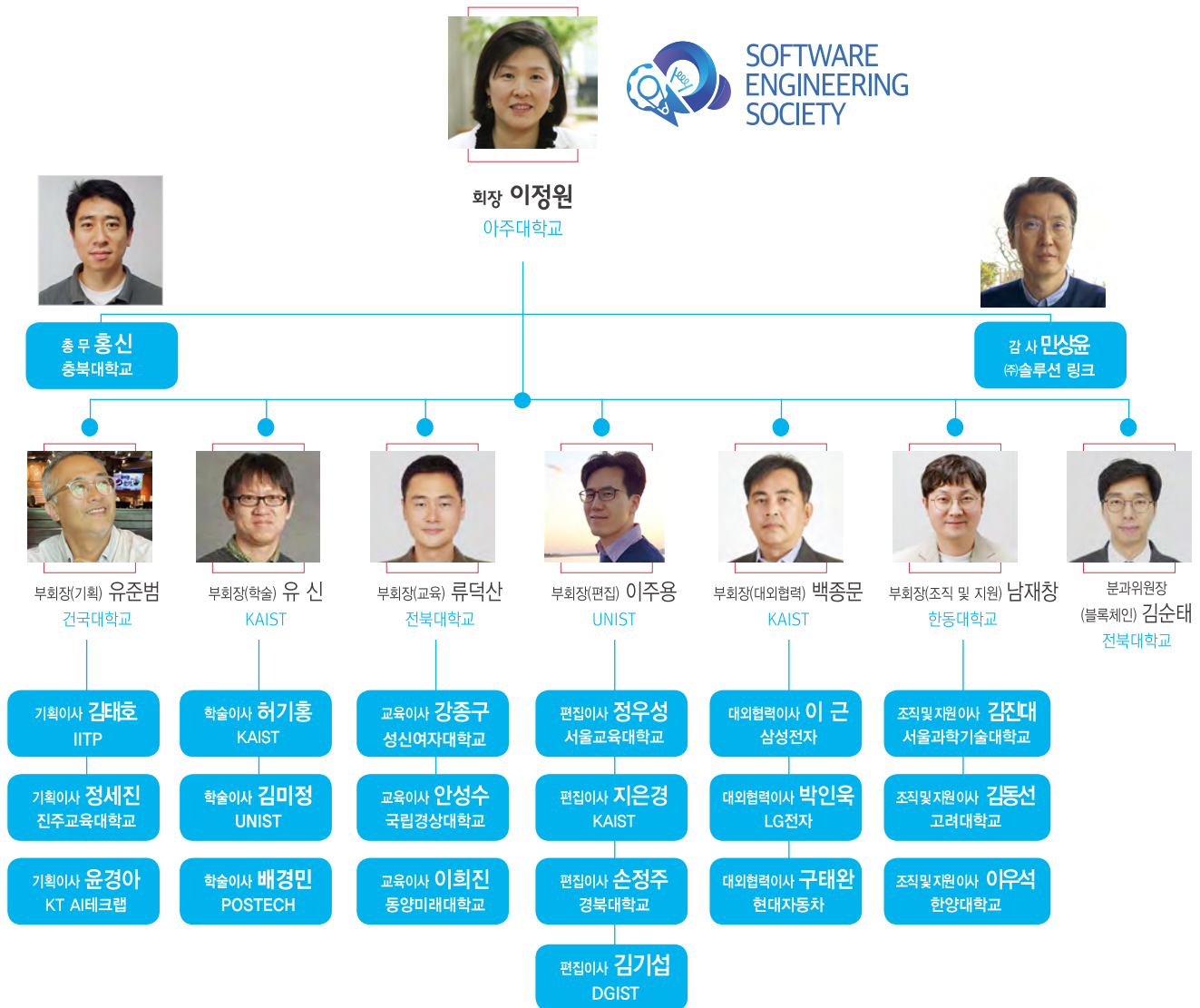
- 이주용 교수 (UNIST, 052-217-2123, jooyong@unist.ac.kr)
- 정우성 교수 (서울교육대학교, 02-3475-2562, wsjung@snu.ac.kr)
- 지은경 교수 (한국과학기술원, 042-350-7810, ekjee@se.kaist.ac.kr)
- 손정주 교수 (경북대학교, 053-950-5556, jeongju.sohn@knu.ac.kr)
- 김기섭 교수 (DGIST, 053-785-6313, kisub.kim@dgist.ac.kr)



SOFTWARE
ENGINEERING
SOCIETY



소사이어티 조직도



발행정보

발행일 2026년 3월 9일

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 <https://www.sigsoft.or.kr>



SOFTWARE
ENGINEERING
SOCIETY

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: <https://www.sigsoft.or.kr>