

ICSE 2023 Trip Report

변지석, 고려대학교 소프트웨어 분석 연구실
Melbourne, 2023 May 14 ~ 20



들어가며

연구실에서 말로만 듣던 소프트웨어 공학 분야 최고 수준 학회들 중 하나인 ICSE (International Conference on Software Engineering) 에 참석했다. 연구실 명호 형의 Fuzzing 연구, 종욱이의 SMT Solver Testing 두 연구발표에 동행할 기회를 얻었다. 덕분에 2023년 5월 14일부터 5월 20일까지 학회 참석 일정을 연구실 사람들과 함께했다.

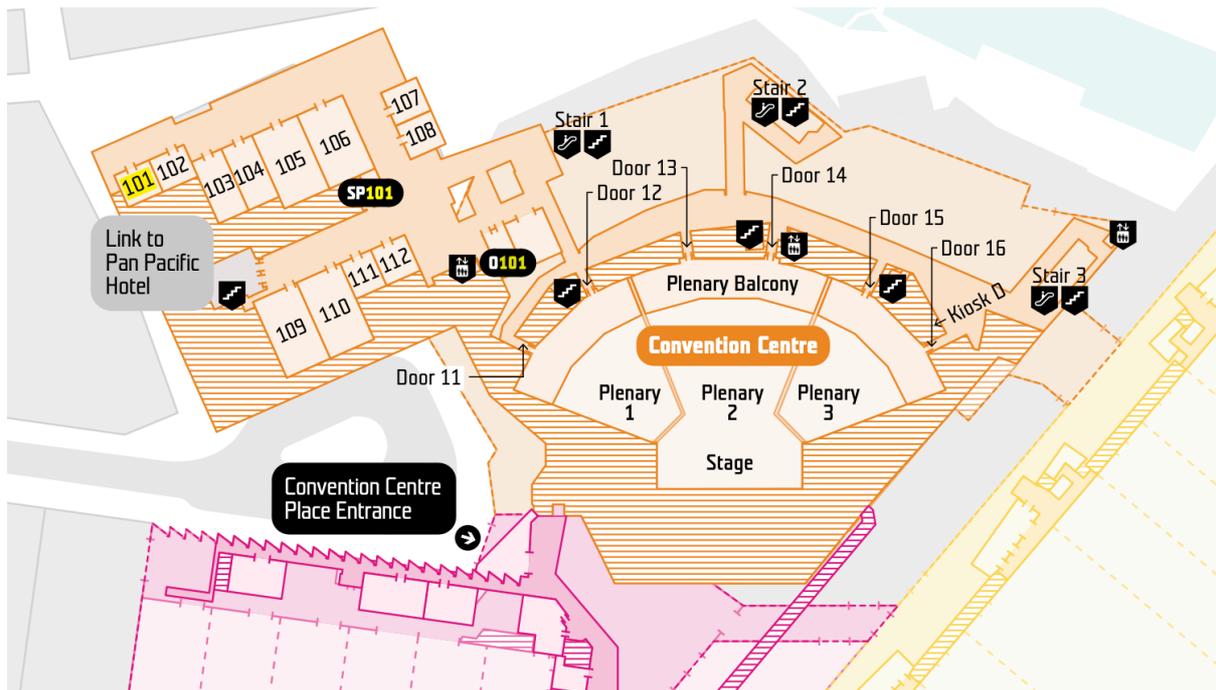
지난 FSE 와 이번 ICSE 에서의 경험을 비교해 보았을 때, 둘은 같은 소프트웨어 공학 분야 학회임에도 불구하고 세션 구성과 발표되는 연구주제들에 큰 차이가 느껴졌다. 특히 ML/AI 기술을 소프트웨어 공학 문제를 풀 때 적용하거나 ML/AI 시스템의 개발에 있어 소프트웨어 공학 문제를 제시하는 발표가 엄청 많아졌다. 그 이유가 무엇인지 아직 감이 잘 오지는 않는데, AI 소프트웨어가 전체 소프트웨어 중 큰 비중을 차지해서인지, ICSE

에서 발표되는 연구들의 방향성이 FSE 와는 원래 다른 것인지, 아니면 그저 내가 FSE 때와는 다른 세션들을 들어서 다르다고 느낀 것인지는 모르겠다.
 이 보고서를 통해 학회에서 경험하고 생각했던 내용들과 인상깊게 들었던 발표들, 그리고 호주라서 느낄 수 있던 독특한 감각들을 공유하고자 한다.

ICSE 2023

학회 진행

학회는 아침 9시부터 오후 5시 15분까지 멜버른 컨벤션 전시센터 (MCEC) 의 9개 방에서 주로 진행되었다. 수백명을 수용할 수 있으며 방송 장비가 갖춰진 1층 대강당 한 곳이 있었고, 2층의 수십명에서 백 명 이상이 모일 수 있는 다양한 크기의 방들을 왔다갔다하며 발표들을 들었다. 외벽과 맞닿은 방 맞은편에는 대형 유리 외벽이 설치되어 탁 트인 모습을 보여줬는데, 그 앞에 앉을 수 있도록 소파나 탁/방석이 마련되어 있어 운이 좋으면 창가의 빈 자리에 앉아 쉴 수 있었다. 복도 크기도 꽤 넓었지만 학회 참여인원은 훨씬 더 많아서, 휴식시간에는, 특히 점심시간에는 서서 얘기를 나누는 사람들 사이를 비집고 지나가야만 움직일 수 있었다.



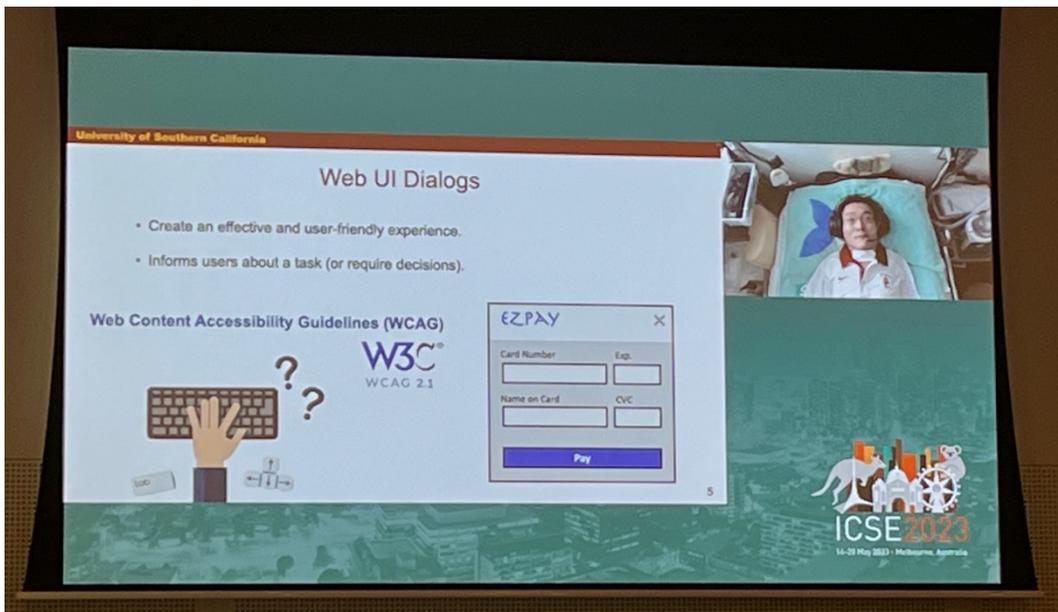
학회는 사진 왼쪽의 Room 101 ~ 110, Plenary 1, 그리고 이들과 연결된 복도에서 주로 진행되었다.
 (출처: MCEC Floor Plans and Space Capacities Guide)

Detecting Dialog-Related Keyboard Navigation Failures in Web Applications

대화창(dialog)을 가진 웹 응용프로그램 위를 키보드로 자유롭게 돌아다닐 수 있는지 알아내기 위한 연구이다. 마우스를 사용하기 어려운 장애인이 소프트웨어를 쓰기 위해서는 키보드 인터페이스 지원이 필요한데, 오늘날 웹 서비스들은 이런 인터페이스를 잘 지원하지 않는다고 한다. 예를 들어, 웹 페이지 위에 로그인 대화창이 띄워졌을 때,

대화창을 끌 수 있는 X 버튼에 키보드 tab 키로 접근을 할 수 없는 상황(Keyboard Dialog Failures, KDF)이 대표적인 예이다. 이런 오류 상황을 잡아내기 위해 웹 상의 키보드 탐색을 정의하고, 그 위에서 대화창을 발견하고 KDF 오류를 찾아내는 자동화된 과정을 제시했다.

발표자는 움직일 수 없어 침대에 가만히 누워 화상으로 발표와 질의응답에 참여했는데, 그런 모습에서 이 연구의 필요성과 그의 열정이 자연스럽게 전달되었던 것 같다.



DLInfer: Deep Learning with Static Slicing for Type Inference

이 연구는 신경망으로 파이썬 값의 타입을 더 잘 추론하기 위해 추론할 값과 관련 없는 구문들을 data flow analysis 결과를 이용해 미리 잘라낸 뒤 남은 값을 신경망에 질의하는 방법을 소개한다. 몇 달 전에 신경망 기반 파이썬 타입 추론 기술들의 성능을 살펴보던 중 DLInfer 가 Top-1 정확도 98% 를 달성했다는 얘기에 흥미가 생겨서 발표를 챙겨 들었지만, 발표 내용을 쉽게 이해할 수 없어 이후 논문을 직접 찾아보면서 세부적인 숫자를 찾아볼 수밖에 없었다. 다른 기존 신경망 기반 타입 추론기에 비하면 놀랄 만한 정확도 수치를 얻긴 했지만, 라이브러리 타입이나 사용자 정의 타입 추론능력의 재현율은 50%, 5% 수준으로 여전히 상당히 낮았기 때문에 이 연구결과가 정적 분석의 이점을 최대한 챙긴 신경망 기반 타입 추론기라고 생각되지 않았다.

Coverage Guided Fault Injection for Cloud Systems

이 연구는 어떤 노드 크래시 시나리오에서 분산시스템이 이 문제상황을 정상으로 복구하지 못하는 버그를 찾기 위한 연구이다. 분산시스템에서의 테스트는 시스템에 참여한 노드 수가 증가하면서 테스트해야 할 경우의 수가 조합적으로 늘어나기 때문에, 이 문제를 해결하기 위해 coverage 를 많이 늘렸던 테스트케이스를 우선적으로 변이시키는 방법을 제안했다. 결과적으로 네 개의 새로운 버그를 찾아낼 수 있었다고 한다.

내게는 분산시스템이 안전성/무결성 등의 성질 검증의 대상으로만 여겨지고 있었지 여기서 버그를 테스트로 찾는 것은 생각해보지 못했던 내용이라 인상깊었다.

Metamorphic Testing Session

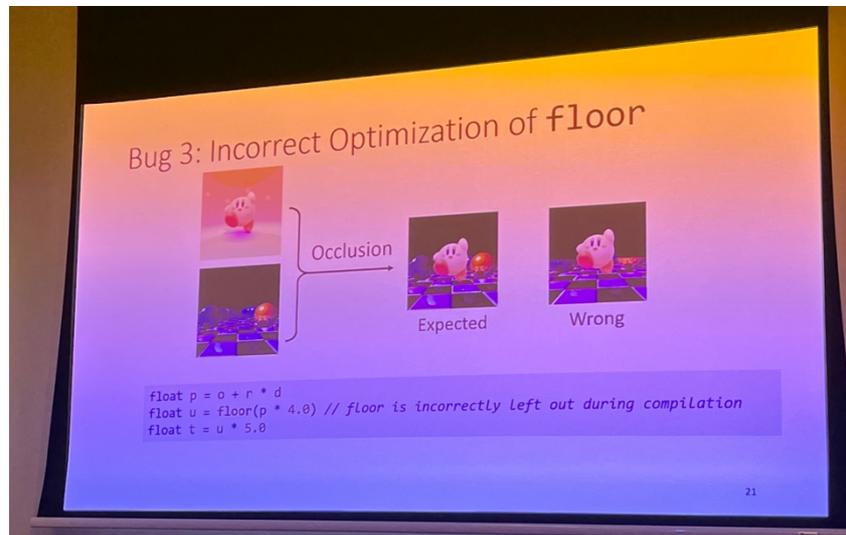
마지막 날 마지막 세션으로 Metamorphic Testing 세션을 들었다. 전혀 생소한 테스트 방법이었는데, 이번에 발표된 중옥의 Diver 와 비슷한 느낌의 테스트라는 설명을 듣고 얼추 알 것 같기도 했다. metamorphic testing 은 부분적인 입출력 명세로부터 그로부터 테스트할 수 있는 성질들을 도구를 가지고 프로그램을 테스트하는 방법이었다. 예를 들어, 프로그램 F 에 대해 입출력 $(F(1, 2, \text{"hello"}) = 1)$ 와 테스트할 수 있는 성질 “for all A, B. $P(A, B) \rightarrow F(A) < F(B)$ ” 를 가지고 있으면 값 $(1, 2, \text{"hello"})$ 을 변이시킨 값 B 가 $P((1, 2, \text{"hello"}), B)$ 를 만족할 때 $1 < F(B)$ 를 만족하는지 테스트하는 식으로 $(1, 2, \text{"hello"})$ 외의 테스트 입력값을 더 생성해낼 수 있다고 이해했다.

세션에 등장했던 Metamorphic 테스트 대상 프로그램들 중 인상깊었던 내용들을 소개하자면;

- Textual Content Moderation Software. 주어진 글이 나쁜 말인지 아닌지 판단하는 도구가 같은 의미의 문장에 다른 출력을 내놓으면 그것은 버그이다. 예를 들어 필터링에 걸리지 않기 위해 “Die” 를 “Dye” 라고 글자를 바꿔 썼다 할지라도 “Die” 가 나쁜 말로 쓰였다면 “Dye” 도 나쁜 말로 판단해야 한다는 식의 기준이었다.
- Graphics Shader Compiler. 셰이더 컴파일러가 임의의 셰이더를 정확히 컴파일/최적화 하는지 테스트하기 위해, 컴파일된 두 셰이더 $f(x), g(x)$ 의 두 픽셀 출력값을 합친 결과와 두 셰이더를 합쳐 표현한 셰이더의 출력결과가 같은지 테스트한다. 발표에서 두 셰이더를 합쳐 표현한 셰이더의 코드 크기가 원래 셰이더보다 몇 배는 더 크다는 자료를 보여줘서 의아했는데, 나중에 질문해보니 뭔가 두 셰이더를 합치면 당연히 그렇게 된다고 답을 줘서 의아해했었다.
- 미국 세금준비 프로그램. 미국의 세금신고를 위한 계산은 복잡하고 어렵기에 이를 도와주는 세무 프로그램이 여럿 있지만, 미국의 세법에 따르면 정확한 금액 계산의 의무는 납세자에게 있다고 한다. 이 연구는 세금준비 프로그램이 잘못된 계산을 하는 버그를 잡는 metamorphic 테스트를 제시하는데, 예를 들어 65 세 이상의 노인은 동등한 조건의 젊은 사람보다 얼마만큼 세금을 적게 내야 한다는 metamorphic relation 을 수집하고, 이를 통해 랜덤하게 생성한 다양한 사람과 계산된 세금 액수가 제약조건에 맞지 않는 경우를 찾아냈다.
- Qiskit Quantum Computing Platform. Qiskit 은 양자 회로를 파이썬으로 표현할 수 있게 해주는데, 이를 양자회로로 컴파일하고 시뮬레이션을 제대로 하는지 테스트하기 위해 양자 회로간의 metamorphic relationship 을 세워 테스트를 진행했다.

세션을 듣고 난 뒤, 망치를 들면 모든 문제가 못으로 보인다는 말마따나 metamorphic testing 방법을 모든 프로그램 테스트에 적용시킬 수 있을 것 같은 기분으로 학회장을

나왔다. 테스트 대상 프로그램의 모든 input/output 관계를 알지 않아도, differential testing 에 사용할 같은 스펙의 또 다른 구현체를 가지지 않아도 테스트케이스를 생성할 수 있다는 점이 매력적이었다.



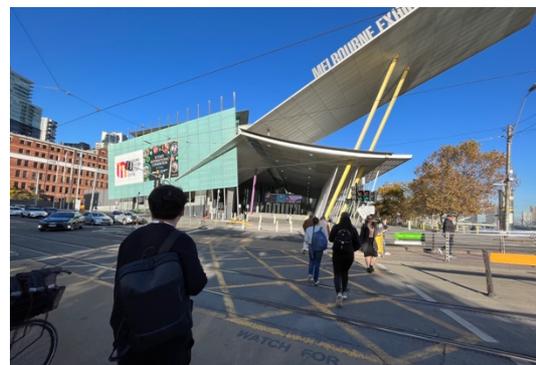
셰이더 컴파일러를 대상으로 한 metamorphic 테스트 연구의 발표자료. Expected 와 달리 캐릭터 양 옆의 빨강/파랑 공 절반씩이 렌더링되지 않은 버그를 발견했다고 한다. 이번 학회장에서는 이상하게도 흰 화면이 나오도록 빔 프로젝터 화면을 찍기가 어려웠었다.

멜버른에서 느꼈던 강렬한 색과 형태들, 그리고 광활한 자연

나는 멜버른에서 강렬한 색과 형태를 가진 조형물들에 자주 시선을 뺏겼다. 유독 멜버른에서 20 세기 후반 추상미술을 디자인적 요소로 자주 접하는 느낌을 받았는데, 집에 와서 생각해보니 그 이유로 눈에 띄는 조형물들에 과감한 원색을 사용했기 때문인 것 같다. 그 예로, 공항에서 멜버른 시내로 가는 버스에서 커다란 빨간 기둥들을 접했고, 학회가 열렸던 멜버른 전시센터에는 샛노란 두 기둥이 격하게 건물을 찌르는 처마를 받치고 있었다.



공항 가는 길에 마주한 빨간 기둥 조형물



표지의 MCEC 건물 입구를 오른편에서 본 모습. 처마를 받드는 노란 얇은 기둥이 눈에 띈다.

호주의 그레이트 오션로드는 내게 환상적인 자연환경을 보여주었다. 거대한 파도가 해안선을 향해 밀려오는 모습, 인간의 손이 닿지 않은 곳에서 오랫동안 침식되어온 대륙의 바위들이 보이는 아름다움은 말로 다 표현하기 어려울 정도였다.

그레이트 오션로드에서는 호주의 넓은 해안을 즐길 수 있었고, 해안을 보러 이동할 때와 다시 숙소로 돌아올 때에도 호주의 많은 면면을 접할 수 있었다. 넓은 목초지에 방목중인 소와 양들, 야생동물을 위해 설치되지 않은 가로등, 넓은 지평선 위의 석양, 농장 구역들을 시각화하기 위해 직각/일렬로 촘촘히 심어진 수천 그루의 나무들은 호주라는 넓고 평탄한 대지 위에서의 인간의 생활상을 내게 보여주었다.



마치며

이번 ICSE'23 학회에서는 내게 생소했던 도메인을 대상으로 한 테스트 발표들을 들으며 “저런 프로그램들의 어떤 성질을 증명하기 위해 테스트를 필요로 하는구나”를 듣는 것이 가장 재미있었던 것 같다. 지금까지 내가 생각지 못한 많은 테스트 도메인들과 이를 풀기 위한 방법들을 접하며 프로그램 성질 증명에 대한 내 식견이 좀 더 넓어진 기분이 들었다. 앞으로도 최고 수준의 PL 및 SE 학회를 자주 참석할 수 있었으면 하는 바람과 함께, 학회 참석 기회를 주신 오학주 지도교수님과, 함께한 연구실 선, 후배 모두에게 감사드린다.