Research Statement

KIJIN AN

My research focuses on the System ends of Software Engineering to support the development and evolution of distributed systems and apps. Actively used software apps must be changed continuously to ensure their utility, correctness, performance, safety, etc. To perform these changes, programmers spend a considerable amount of time and effort pinpointing the exact locations in the code to modify, a particularly hard task for distributed apps. My dissertation research is concerned with facilitating the evolutionary modifications of distributed apps, including apps of machine learning, via automatic and architectural refactorings. By flexibly changing the application architecture, with different functionalities becoming local or remote at will, programmers can more easily perform various kinds of perfective and adaptive modifications. I have published 23 peer-reviewed conference/journal papers, with one best paper award and two best paper nominations.

Before commencing my Ph.D. studies, I worked as a researcher in Korea Institute of Science and Technology (KIST), one of the top research institutions in South Korea. Much of my work there focused on distributed systems for interactive robot services, a project with a $5 million budget, executed according to a rigorous delivery and evaluation schedule. During the three years of my involvement in this project, I contributed novel research results, published in 10 research papers, and used as proofs of feasibility for two large successful funding proposals. The project received the prestigious government award, having been selected as the Industry Technology of the Month.

For my Master's in POSTECH, I was a key contributor to the research project that implemented a Sensor Network's protocol using an actor-oriented framework (DGIST, UC Berkeley). Based on the successful completion of this work, we received another research grant to create a location recognition system (DGIST). I also participated in several proposals in SK telesys, where I was a network systems engineer, responsible for developing commercial 3·4G·WiFi and VoIP productions.

Dissertation Work: ***Automated Architectural Refactoring and Its Applications*** My dissertation research is concerned with creating novel automated program transformations that enhance the execution of web apps. Refactoring is a program transformation that changes the source code of an app while preserving its semantics. The key novelty of this research is a novel domain-specific refactoring—*Client Insourcing*—that transforms a distributed app into its semantically equivalent centralized variant, in which the remote parts are glued together and communicate with each other by means of regular function calls. The equivalent variant is then for adaptive and perfective modifications and subsequent automated redistribution, thereby enhancing app execution, with a particular emphasis on performance, energy efficiency, and reliability [1, 2]. Rather than encoding domain-specific preconditions, our approach uses domain agnostic semantic to identify the entry/exit points of middleware functionality. The main work was presented in *theWebConf 2020* [4] (19%, 217/1129).

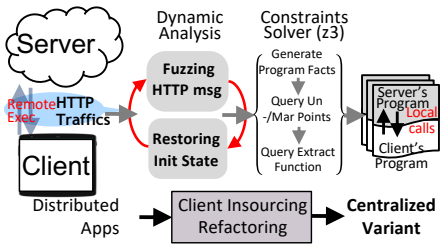

Figure 1: Client Insoucing Refactoring

Adapting Web Execution at Runtime: ***ICWE 2021 Best Paper Award***: A mobile web app's performance depends heavily on the underlying network and the client device's hardware capacity. Static distribution allocates some application functionality to run on the client and some on the server, with the resulting allocation remaining fixed throughout the app's execution. If the available network and the client device mismatch those assumed during the static distribution, app responsiveness and energy efficiency can suffer. To address this problem, we propose Communicating Web Vessels (CWV) [6], an adaptive redistribution framework that improves the responsiveness of full-stack JavaScript mobile apps. By monitoring the network conditions, CWV determines whether a dynamic redistribution would improve application performance. It then triggers a redistribution that moves server-side functionalities to the mobile client and vice versa. When CWV moves server-side functions and their reachable state to the client, they are invoked as regular local functions. The moved functionalities are accessed remotely again as soon as the network conditions improve. This work presented in ICWE 2021 (17%, 22/128).

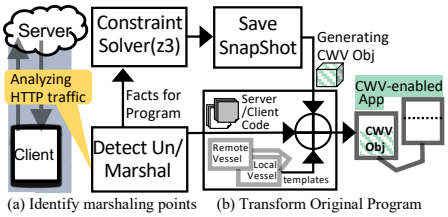

Figure 2: Communicating Web Vessels

Correcting Ill-Conceived Distribution Granularity for Cloud Services: Distributed applications enhance their execution by using remote resources. However, distributed execution incurs communication overheads, if these overheads are not offset by the yet larger execution enhancement, distribution becomes counterproductive. For maximum benefits, the distribution's granularity cannot be too fine or too crude; it must be just right. We introduce a novel approach—*D-Goldilocks* [5]— to re-architecting distributed applications, whose distribution granularity has turned ill-conceived. To adjust the distribution of such applications, our approach automatically reshapes their remote invocations to reduce aggregate latency and resource consumption by means of a series of domain-specific automatic refactorings. This work was presented in *SANER 2020* (21%, 42/199).

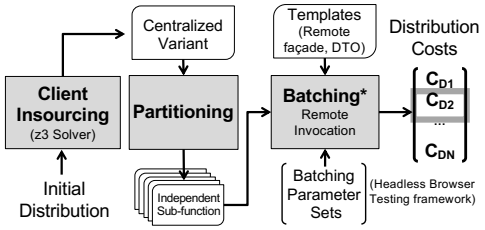

Figure 3: Correcting Dist. Granularity

Efficient Bug Fixing in Distributed Apps: We introduce a novel debugging approach to localizing bugs in distributed applications in *ICWE 2019* [3]. The debugged application is converted to its semantically equivalent centralized version with Client Insourcing, then this variant is debugged to fix various bugs. Finally, based on the bug fixing changes of the centralized version, a patch is automatically generated to fix the original application source files.

Please find the detail of my work in: `https://kjproj84.github.io/publications`.

Future Research Plans The automated software analysis and transformation toolset created by my dissertation research has a range of applications that can help solve some of the most salient problems faced by the modern computing ecosystem. Although edge computing is seen as an enabler of the next iteration of software technologies, including IoT, autonomous driving, and smart houses, integrating edge computing and storage resources is strewn with complex challenges. I am looking forward to applying my research expertise and skills to address these challenges. One of my ongoing research efforts focuses on integrating edge-based computing and storage resources into existing distributed systems by replicating cloud services at the edge.

Replicating Cloud Services (e.g. machine learning) at the Edge: To take advantage of edge computing, two-tier client-cloud apps need to be transformed to three-tier client-edge-cloud apps. (e.g. detecting objects of transmitted images in cloud by using deep learning) Such transformations are hard for programmers to perform correctly by hand. Many cloud services maintain runtime state that needs to be replicated at the edge. Once replicated, this state must then be synchronized efficiently and correctly. To facilitate the transition to edge computing, my research explores how to automatically transform client-cloud apps to their client-edge-cloud versions. My approach automatically replicates cloud-based services at the edge by synchronizing state via generated Conflict-Free Replicated Data Types (CRDT). CRDT provide strong eventual consistency, so each replica executes at full speed. By keeping and exchanging histories of state mutations, replicas eventually converge to the same state, even if the network temporarily disconnects.

Applying advanced program analysis and transformation techniques can not only enable fast, energy-efficient, and privacy-preserving computation and storage at the edge, but can also help solve problems that commonly occur in the engineering of other modern distributed systems. I am looking forward to applying my technical expertise and research experience to produce innovative solutions to these problems.

References

[1] Kijin An. Facilitating the evolutionary modifications in distributed apps via automated refactoring. In *Web Engineering*, pages 548–553. Springer International Publishing, 2019.

[2] Kijin An. Enhancing web app execution with automated reengineering. In *Proceedings of the Web Conference 2020*, 2020.

[3] Kijin An and Eli Tilevich. Catch & release: An approach to debugging distributed full-stack JavaScript applications. In *ICWE*, pages 459–473, 2019.

[4] Kijin An and Eli Tilevich. Client insourcing: Bringing ops in-house for seamless re-engineering of full-stack JavaScript applications. In *Proceedings of the Web Conference*, 2020.

[5] Kijin An and Eli Tilevich. D-Goldilocks: Automatic redistribution of remote functionalities for performance and efficiency. In *Proceedings of the 27th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2020.

[6] Kijin An and Eli Tilevich. Communicating web vessels: Improving the responsiveness of mobile web apps with adaptive redistribution. In *ICWE*, 2021.