



Multi-ML cross-platform meta programming at scale using compiler tech and C

PhD annual review, August 2022

Samuel Marks, PhD
Computer Science and Eng
September 30, 2022

Outline

Background

In 1968 the NATO SCIENCE COMMITTEE had a conference—and produced a report—on ‘software engineering’.[nau68]. This was a landmark event, spruning the formation of ‘software engineering’ as a discipline; culminating in a general appreciation of a crisis in programming.[MB87]

Focussed on in my research is the proceeding—from that NATO conference—on software components [Mcl68], i.e., mass development of mass customisable software.

Related foci in my research are: problems of deployment at scale; keeping up-to-date with the massive amount of new Machine Learning; and pulling all these together to facilitate *scalable rapid software engineering*.

bonâ fides

Work (since 2009)

0. Tech support;
1. Software engineer;
2. Senior technology specialist;
3. Senior software engineer
4. Lead software engineer
5. Head/director (running my own engineering consultancy)

Education

0. 2014: Bachelor of Science (wherein I took most all required to get all Computer Science, Information Technology, and Business Information Systems majors) from Macquarie University.
1. 2021: Doctor of Philosophy (PhD) from the University of Sydney.

Problem (0/1)

My experience, verified by speaking at dozens and attended ~100 industry technology events, and further confirmed from the literature, exposed major problems in software-engineering. The ones I am focussing on in this PhD are:

- Native development benefits from targeting iOS, macOS, Windows, Linux, Android, and Linux.
...but this greatly reduces development agility and quality metrics like consistency and test & doc coverage.
- Machine Learning field changes too quickly for anyone to keep up-to-date with. Meaning one cannot claim superiority of their solution across industry/academia.

Problem (1/1)

- The same stack cannot be deployed for: embedded (e.g., whole stack on smartphone); peer-to-peer; and client/server architectures.
- Deploying to different clouds requires a different knowledge set.




In summary, developing quality, complex and multi-tier systems requires:

- a team;
- increasingly specialised knowledge;
- lots of time; and
- an expansive budget.

Solution

Write compilers.

References I

-  Allen Macro and J. N. Buxton, *The craft of software engineering*, International Computer Science Series, Addison-Wesley, Wokingham, England ; Reading, Mass, 1987.
-  Doug McIlroy, *Mass-Produced Software Components*, Proceedings of NATO Software Engineering Conference (Garmisch, Germany) (P. Naur and B. Randell, eds.), October 1968, pp. 138–155.
-  *Software Engineering*, Report to NATO science committee, Garmisch, Germany, October 1968.