CV of Dr Alex Potanin, Associate Professor in Software Engineering, VUW, NZ

Personal Details

Name: Alex Potanin
DOB: 24 April 1981
Work: School of Computing, 108 North Road, Acton, ACT 2601, Australia
Home: 12 Tea Tree Lane, Woodridge, Wellington 6037, New Zealand
Email: alex.potanin@anu.edu.au
Phone: +64 (21) 201 6740 (NZ) or +61 (410) 970 498 (Australia)
Web: https://potanin.github.io

Academic Qualifications

2007 – PhD in Computer Science, Victoria University of Wellington
2002 – BSc (Honours) in Computer Science, Victoria University of Wellington
2001 – BSc in Mathematics and Computer Science, Victoria University of Wellington

Employment Record

2022 – NOW	Associate Professor, Australian National University
2022 – 2022	Deputy Head of School, Victoria University of Wellington
2021 – 2022	Associate Dean (Students), Victoria University of Wellington
2018 – 2022	Associate Professor, Victoria University of Wellington
2019/20 (sabbatical)	Associate Professor, Kyoto University, Japan
2010 – 2017	Senior Lecturer, Victoria University of Wellington
2013 (sabbatical)	Research Scholar, Carnegie Mellon University
2006 – 2009	Lecturer, Victoria University of Wellington
2005 – 2006	Software Engineer, Innaworks Limited
2004	Visiting Researcher, Purdue University
2001 – 2003	Software Engineer, Catalyst Systems Limited (part time)

Professional Memberships:

2002 – NOW, Member of the ACM (currently: Senior Member of the ACM)
2018 – 2022, Member of Engineering New Zealand
2010 – 2014, Member of CORE (CS Departments of Australia/NZ) Executive

Grants and Awards (in NZD):

2020 – 2021 – Robonomics.Network Research Grant ($72,000)
2021 – 2023 – SHEADI Faculty Strategic Initiative PhD Scholarship ($100,000)
2017 – 2018 – Oracle Corporation Research Grant ($70,000)
2017 – ECOOP 2017 *Distinguished Artefact Award.*
2014 – ECOOP 2014 *Distinguished Paper Award*.
2009 – 2011 – RSNZ Fast Start Marsden Grant ($300,000)
2012 – Mozilla Foundation Research Grant ($15,000)

2016, 2008, 2007 – VUW URF Grant ($18,000 in ’16, $12,000 in ’08, $17,000 in ’07)
2009 – RSNZ ISAT Grant ($5420)
2007 – ESEC/FSE ACM SIGSOFT *Distinguished Paper Award*
2005 – 2nd Prize and the Judges Prize at the ICFP Programming Contest 2005 and 2003
2004 – Claude McCarthy Fellowship and 2003 – J. L. Stewart Scholarship
2003 – 2nd Place in the ACM International Research Competition (via OOPSLA2002 SRC)
2002 – Freemasons Scholarship and 2001 – Datacom Scholarship in Computer Science
1997 – Finalist in the George Soros International Mathematics Olympiad
1996 – Winner of the Russian City Tournament in Mathematics in 1996

PC Chair of:

2016: Asia-Pacific Software Engineering Conference (APSEC) – with Gail Murphy
2010, 2011: Computing: The Australasian Theory Symposium (CATS) – with Taso Viglas

General Chair of:

2022: SPLASH/OOPSLA 2022 (Object-Oriented Programming, Systems, Languages, and Applications) to be held in Auckland, New Zealand, combined with APLAS 2022, Dec’ 22.
2018: Asian Symposium on Programming Languages, and Systems (APLAS) 2018, held in Wellington, New Zealand, Dec ’18.
2015: New Object-Oriented Languages (NOOL) Workshop.
2009: Australasian Computer Science Week (ACSW) in Wellington, NZ, Jan ’09.
2004: 15th PhD Workshop and Doctoral Symposium at ECOOP 2005.
2004: 2nd International Dylan Conference. Wellington, NZ. 4-6 February 2004.

PC Member of:

OOPSLA 2021 (ERC), 2020, 2019, 2014, 2011 (ERC), APLAS 2021, 2016, ECOOP 2023, 2022, 2019, 2016, HOTSOS 2019, PLDI SRC 2019, GPCE 2019, NOOL 2016, FTfJP 2016, PLDI 2015 (ERC), FOOL 2014, ACSC 2010, 2013, 2015, 2017, IWACO 2022, 2011, 2007, FormaliSE 2022, 2023, Programming 2022, 2023.

OC Member of:

Steering Committee Member (from 2019) of SPLASH/OOPSLA, Virtualisation Co-Chair for SPLASH/OOPSLA 2021, 2020, Publications Chair of SPLASH/OOPSLA 2015, 2016, 2017, Workshops Chair of SPLASH/OOPSLA 2017, 2018, Publications Chair of ICFP 2018.

Research

I am one of the leading experts internationally in the fields of object ownership (restricting access using graph theoretical properties) and immutability (precise definition of when objects are allowed to change). I have an international research track record in secure programming language design and type systems.

My earlier ownership and immutability work benefitted established programming language designs such as Mozilla's Rust. One of my recent works on exploiting object graph structures uncovered security flaws in Java servers such as Jenkins/Tomcat and JBoss as well as their equivalents for .NET.

Since 2013, my major research project is the Wyvern Secure Programming Language that I co-lead with Associate Professor Jonathan Aldrich (CMU). I bring expertise on formal verification and security while Jonathan provides expertise on software architecture which we applied to produce the first secure and verified software architecture support (we co-authored at least 8 recent papers). Wyvern is an open source project with several major publications every year, 3 PhD students (1 at VUW, 2 at CMU), 1 Master student, and 8 Honours students and research assistants so far.

In 2008 – 2012 I led the Unification of Immutability and Ownership project which resulted in Royal Society of NZ (equivalent of NSF in the US) Marsden Grant and 1 PhD and 3 Master thesis students. I hold position as Publications Chair of SPLASH/OOPSLA and ICFP conference series (working closely with Association for Computing Machinery (ACM) Publications Board and Special Interest Group on Programming Languages (SIGPLAN)) as well serve as the Workshops Co-Chair in 2017/2018 for SPLASH. I was also editing two major journal issues in 2017 (Science of Computer Programming and Journal of Object Technology).

Finally, my research reputation is acknowledged by invitations to Shonan Workshop and IFIP Working Group 2.4 Meetings (in 2019, 2020, and further 2 in 2021) which are all invitation-only events where small number of leading international researchers present and discuss their latest work in an intensive week of interactions.

I have 80 quality assured publications (94 with technical reports and theses) as listed on my homepage: http://homepages.ecs.vuw.ac.nz/~alex/ . So far, I had 5 PhD, 10 Masters, and over 30 Honours students.

Teaching

As a teacher, I get a consistent "overall effectiveness" score around 1.6 (on the scale of 1 to 5). I taught classes in a variety of sizes (5-25 at 400 level, 60-120 at 300 level, 150-300 at 200 level, 300-500 at 100 level) and always receive positive feedback with "perfect" 1.0 score for "Attitude towards students". My overall teaching evaluation for the newly developed course on Software Development for Mobile Platforms in 2018 was the best possible 1.0 and in several courses as recently as 2020 I had close to 1.0 or equal to 1.0 (in SWEN 430 in 2020) teaching evaluations.

In 2019, I led the introduction of the postgraduate AI and ML program at ECS as part of my role as Program Director (Science). This involved staff management and conflict resolution in a timely fashion to meet the required CUAP deadlines.

In 2017, I led a review of Software Engineering (SWEN is the largest major in BE) involving all the academic staff in Engineering and Computer Science (ECS). I led a discussion by over 20 software-related academic staff members over the first half of 2017 which led to a proposal I wrote that was approved by the VUW Academic Board in the middle of 2017. This is the first and most significant overhaul of SWEN and the BE since its creation 8 years ago. This will feed directly into our 5-yearly Engineering re-accreditation in 2017 which I will lead for SWEN. Some specific examples of teaching innovation and course development include:

1. COMP 2120 re-design at the ANU. I took over 2nd year Software Engineering course and has revised both the assessment and content to create a modern group project oriented software engineering course with the ultimate goal of contributing a pull request to a large open source project by every one of the 50+ teams.
2. SWEN 325 creation of a brand-new course in 2018. To fill the gap in teaching our students high level frameworks and software architecture as well as the in demand skills in development of software for mobile platforms and IoT, I developed from scratch a brand new final year undergraduate course that covers architecture for the mobile systems, UX for mobile, being able to learn and work with multiple modern frameworks such as React Native and Ionic, and writing apps for and interacting with IoT devices. The course was very successful with overall teaching evaluation for me 1.0 and course overall evaluation of 1.8.
3. COMP361 Full Course Redesign in 2014. I have addressed low enrolments in a third year Algorithms course (formerly known as COMP303) by reinventing it as COMP361 with a readjusted aim and course objectives to make it more practical for modern software engineers. The redesign was a success: increasing the enrolments tenfold from 5-6 for COMP303 to around 60 from 2014 onwards. I am the sole person responsible for the course and its design and delivery. The final 2014 overall course feedback was 2.0 - which is quite reasonable given that from the smallest COMP300 course it has now turned into the second largest after COMP307 and the largest in T2 for COMP300.
4. SWEN302 Full Course Redesign in 2014. In 2013, SWEN302 ended up in need of a “rescue” when it received one of the worst course evaluations in ECS history: 4.0. I volunteered for the task as the course coordinator and completely revamped the way this agile project course is run by bringing in mostly internal, very highly involved clients and introducing “time boxing”: all the students were asked to pick one day every week that they had the least amount of lectures. The teams were then formed for those days of the week and the requirement was that students should only work 8 hours on the day picked (in addition to 2 hours spent in a lecture). This worked wonders as it allowed students to control their time on one hand and to avoid any team members “slacking” as it was clear who had to be around when. More importantly, this made all the students learn the most essential aspect of agile management: being aware of the time spent on different parts of the project and be able to make accurate estimates at all times. The course received anecdotally positive feedback and the overall score decreased significantly (to 2.0 in 2014).
5. COMP261: Together with Dr Peter Andreae in 2010 we developed a brand-new course on Algorithms to fill the need for our up and coming Computer Graphics

program as well as providing an elective foundational course on Computer Algorithms for our Bachelor of Engineering students. The course was designed from the ground up to fit into the new 15 points system to keep the workload for students manageable. I put together a custom textbook with the help of the Pearson publisher that we use to teach COMP261. After nine years, COMP261 remains the most popular course in undergraduate computer science degree at VUW.

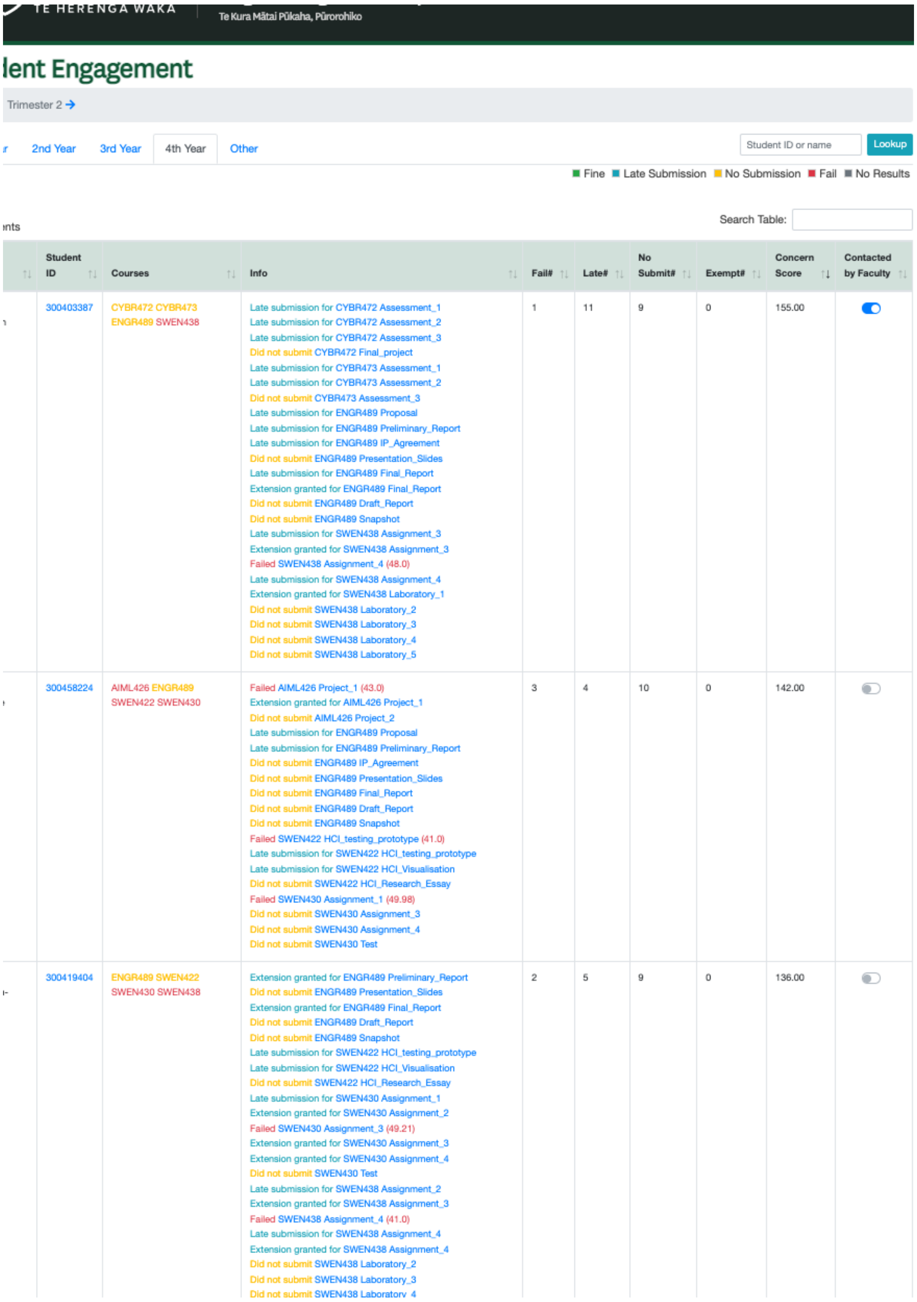
6. SWEN430 Full Course Redesign in 2010. Together with Dr David Pearce in 2010 we redesigned our graduate course to offer our up and coming Bachelor of Engineering students (especially SWEN majors) a strong course on compiler engineering and advanced Java programming. This course was a success: Innaworks and other Wellington start-ups sought after the students that took this particular course. In 2012, it was rated as the most useful course in course evaluations of SWEN400 courses.

Service

Since I started at the ANU in June 2022, I am the degree convenor for the Bachelor of Engineering in Software Engineering.

In 2022, I was employed as Deputy Head of School as additional 0.4 FTE on top of my other admin role described next to help with the transitionary period as we are split into 4 sub-schools. I have 15 academic staff as direct reports ranging from Assistant Lecturer to full Professor in the areas of Software Engineering and Cyber-Security. I have already organised all of them to conduct annual performance development plans for the year for February 2022 – less than a week from starting my role. One of the main reasons I have been asked to fill this role was because I managed to both accredit and revise our BE in Software Engineering during my tenure as Program Director (described below) while keeping all the academics involved both engaged and happy with the proposed changes and getting them through all the necessary academic proposal stages, including VUW Academic Board, and eventual approval by the NZ national CUAP committee, as well as the scheduled accreditation by the Engineering NZ.

Since 2021, I am employed as 0.4 FTE Associate Dean (Students) at the Faculty of Engineering that looks after approximately 1600 EFTS (or 2000 actual students as some are part time) in the School of Engineering and Computer Science and School of Mathematics and Statistics. One of the first achievements to date was my creation of the Student Engagement System that allows me to pro-actively monitor the progress of ~2000 students in our Faculty: picking up those who are asking for extensions, missing deadlines, or not coping with the courses that they are currently enrolled in. With the help of the Faculty Office staff, we can reach out early to them before they become non-engaged and place appropriate support measures in place ranging from connecting them with counselling services to offering withdrawals and re-weighting of the assessments. Alongside the Student Engagement System (SES), I have also centralised our assessment scheduling that is now done pro-actively before each trimester begins because the academic staff saw the clear benefits of the SES in picking up struggling students early and were willing to provide the relevant information early to help track the progress and help the students succeed. Finally, our extension requests are now integrated into our Assessment System to both ensure consistency (especially in the times of Covid) and the ability to easily monitor the students' progress. Below is the screenshot with no names showing one of the tabs in the system tracking the students and sorting by the "concern score". Finally, one of the examples of numerous examples of external engagement, was my negotiation in January 2022 of a special pathway from a regional polytechnic (WITT) that allows the students doing first year locally as BEngTech in the regional NZ (e.g. still staying with their parents in the time of Covid) to then do our summer papers in COMP remotely in the summer to join alongside our 2nd year students to be able to complete BE at VUW – easing the transition from the regions to the far away city University study.


TE HERENGA WAKA
Te Kura Mātai Pūkaha, Pūrorohiko

lent Engagement

Trimester 2 →

r
2nd Year
3rd Year
4th Year
Other
Student ID or name
Lookup
🟩 Fine 🟦 Late Submission 🟨 No Submission 🟥 Fail ⬛ No Results
Search Table:
ınts

	Student ID	Courses	Info	Fail#	Late#	No Submit#	Exempt#	Concern Score	Contacted by Faculty
	300403387	CYBR472 CYBR473 ENGR489 SWEN438	Late submission for CYBR472 Assessment_1 Late submission for CYBR472 Assessment_2 Late submission for CYBR472 Assessment_3 Did not submit CYBR472 Final_project Late submission for CYBR473 Assessment_1 Late submission for CYBR473 Assessment_2 Did not submit CYBR473 Assessment_3 Late submission for ENGR489 Proposal Late submission for ENGR489 Preliminary_Report Late submission for ENGR489 IP_Agreement Did not submit ENGR489 Presentation_Slides Late submission for ENGR489 Final_Report Extension granted for ENGR489 Final_Report Did not submit ENGR489 Draft_Report Did not submit ENGR489 Snapshot Late submission for SWEN438 Assignment_3 Extension granted for SWEN438 Assignment_3 Failed SWEN438 Assignment_4 (48.0) Late submission for SWEN438 Assignment_4 Extension granted for SWEN438 Laboratory_1 Did not submit SWEN438 Laboratory_2 Did not submit SWEN438 Laboratory_3 Did not submit SWEN438 Laboratory_4 Did not submit SWEN438 Laboratory_5	1	11	9	0	155.00	●
	300458224	AIML426 ENGR489 SWEN422 SWEN430	Failed AIML426 Project_1 (43.0) Extension granted for AIML426 Project_1 Did not submit AIML426 Project_2 Late submission for ENGR489 Proposal Late submission for ENGR489 Preliminary_Report Did not submit ENGR489 IP_Agreement Did not submit ENGR489 Presentation_Slides Did not submit ENGR489 Final_Report Did not submit ENGR489 Draft_Report Did not submit ENGR489 Snapshot Failed SWEN422 HCI_testing_prototype (41.0) Late submission for SWEN422 HCI_testing_prototype Late submission for SWEN422 HCI_Visualisation Did not submit SWEN422 HCI_Research_Essay Failed SWEN430 Assignment_1 (49.98) Did not submit SWEN430 Assignment_3 Did not submit SWEN430 Assignment_4 Did not submit SWEN430 Test	3	4	10	0	142.00	○
	300419404	ENGR489 SWEN422 SWEN430 SWEN438	Extension granted for ENGR489 Preliminary_Report Did not submit ENGR489 Presentation_Slides Extension granted for ENGR489 Final_Report Did not submit ENGR489 Draft_Report Did not submit ENGR489 Snapshot Late submission for SWEN422 HCI_testing_prototype Late submission for SWEN422 HCI_Visualisation Did not submit SWEN422 HCI_Research_Essay Late submission for SWEN430 Assignment_1 Extension granted for SWEN430 Assignment_2 Failed SWEN430 Assignment_3 (49.21) Extension granted for SWEN430 Assignment_3 Extension granted for SWEN430 Assignment_4 Did not submit SWEN430 Test Late submission for SWEN438 Assignment_2 Extension granted for SWEN438 Assignment_3 Failed SWEN438 Assignment_4 (41.0) Late submission for SWEN438 Assignment_4 Extension granted for SWEN438 Assignment_4 Did not submit SWEN438 Laboratory_2 Did not submit SWEN438 Laboratory_3 Did not submit SWEN438 Laboratory 4	2	5	9	0	136.00	○

In 2017 – 2018, I served as Program Director and lead the Software Group delivering the SWEN, NWEN, and CYBR majors in BE and HSWD major in Bachelor of Health which involves working with over 20 academic and other teaching staff and assisting Head of School in running these programs. Major part of this role is to enable staff to work productively and constructively together to deliver the best learning experience to students. In 2019 I moved to Program Director (Science) role to lead the introduction of the AI & ML postgraduate program before leaving on sabbatical in 2020.

As ECS Postgraduate Coordinator in 2012 – 2016, I led an interdisciplinary postgraduate research program with nearly 100 PhD and Master students, chaired every PhD proposal meeting, and attended to all matters related to thesis students. I significantly clarified and improved the procedures for induction, reporting, handling of problem cases, PhD progression to full registration, Masters progress reporting and examinations at the Engineering Faculty during my term, which was my key responsibility. My work earned praise from the FGR and received a commendation from BE Review Panel in 2016. When I finished, my responsibilities were handed over to three Associate Professors and a professional staff member.

In 2018 I was part of the Engineering New Zealand Accreditation Panel to re-accredit University of Canterbury's Bachelor of Engineering with nine majors and I represented NZ Qualifications Authority to accredit a new Bachelor of Information Technology degree proposed by Whitireia and WelTec polytechnics. I remain an annual monitor for the "W and W" IT programs at postgraduate and undergraduate level for 2020, 2021 and ongoing basis.

Finally, I am a member of the [senior] research track program committees for many international conferences including A and A* ranked: OOPSLA (2021, 2020, 2019, 2014, 2011) and ECOOP (2019, 2016) and many others as listed on my web page.

International:

- Steering Committee Member and General Chair of SPLASH/OOPSLA 2022.
- General Chair for Asian Programming Languages and Systems (APLAS) Conference in 2018 to be held in December in Wellington, New Zealand.
- Program Committee Co-Chair for Asia-Pacific Software Engineering Conference (APSEC) 2016 – a highly prestigious appointment where a co-chair (Gail Murphy, Associate VP International, University of British Columbia) agreed to join solely because of prior good working relationship with me.
- Publications Chair for SPLASH/OOPSLA conference series in 2015 – 2017. I oversaw a major shift from Sheridan publishing arrangement producing SIGPLAN Notices (minor journal) issue for OOPSLA proceedings to Conference Publishing Consulting (different publisher) producing Proceedings of the ACM (major journal) issue for OOPSLA proceedings.
- General Chair of a major international conference with over 300 attendees held in Wellington: Australasian Computer Science Week (ACSW) 2009. ACSW includes 10 subconferences and received a commendation from the President of CORE on the very high quality of its organisation. As a result, I was invited to serve on the CORE executive to oversee the ACSW conferences for the five years that followed.

University:

- member of the University Learning and Teaching Committee and Chair of the Faculty Learning and Teaching Committee, 2021 - ongoing;
- VUW Civic Engagement Group, 2021 – ongoing;
- Engineering Representative, Academic Board, 2015 – 2018 and 2021 – ongoing;
- member of Your Voice Workload Working Group in 2016/2017;
- member of Conference and Research and Study Leave Review Working Group in 2010/2011.

Faculty:

- member of Faculty of Engineering Executive;
- member of Faculty of Health Teaching, Learning, and Equity Committee in 2017 as we create the new courses and programs in BHlth;
- member of Faculty of Engineering Academic Development Committee.

School:

- member of ECS Teaching and Learning Committee.

Courses Taught

NB! 100 level courses are usually in the 400 or so range. 200 level is in the 300 or so range. 300 level is in the 100-200 or so range. 400 level is around 20-30 people range.

1. COMP 2120 Semester 2 of 2022 (Software Engineering)
2. COMP 103 T3 (Jan/Feb) 2022 (Introduction to Data Structures and Algorithms)
3. COMP 102 T3 (Summer, Half) 2021 Nov/Dec (Introduction to Computer Program Design)
4. COMP 261 T1 (Second Half) 2021 (Algorithms and Data Structures)
5. COMP 361 T1 (First Half) 2021 (Design and Analysis of Algorithms)
6. COMP 103 T3 (Second Half) 2020 (Introduction to Data Structures and Algorithms)
7. SWEN 430 T2 (Second Half) 2020 (Compiler Engineering)
8. SWEN 325 T2 (Weeks 2, 3, 4) 2020 (Software Development for Mobile Platforms)
9. Kyoto University Graduate Seminars in Winter 2019/2020
10. SWEN 325 T2 2019 (Software Development for Mobile Platforms)
11. SWEN 325 T2 2018 (Software Development for Mobile Platforms)
12. COMP 361 T2 2017 (Design and Analysis of Algorithms)
13. COMP 261 T1 (Course Coordinator Only) 2017 (Algorithms and Data Structures)
14. SWEN 221 T1 (Course Coordinator Only) 2017 (Software Development)
15. COMP 103 T2 (Coordinator Only) 2016 (Intro to Data Structures and Algorithms)
16. COMP 361 T2 2016 (Design and Analysis of Algorithms)
17. COMP 261 T1 (9 of 12 weeks) 2016 (Algorithms and Data Structures)
18. COMP 361 T2 2015 (Design and Analysis of Algorithms)
19. SWEN 302 T2 (Essays and Course Coordinator Only) 2015 (Agile Methods)
20. ENGR 123 T2 (Labs Only) 2015 (Engineering Mathematics with Logic and Statistics)
21. COMP 261 T1 (First Eight Weeks) 2015 (Algorithms and Data Structures)
22. COMP 361 T2 2014 (Design and Analysis of Algorithms)
23. SWEN 302 T2 (Coordinator Only) 2014 (Agile Methods)

24. COMP 261 T1 (Second Half) 2014 (Algorithms and Data Structures)
25. SWEN 223 T1 (First Half) 2014 (Software Engineering Analysis)
26. SWEN 430 T1 (First Half) 2014 (Compiler Engineering)
27. COMP303 T2 2012 (Design and Analysis of Algorithms)
28. SWEN430 T2 2012 (Compiler Engineering)
29. COMP103 T1 2012 (Introduction to Data Structures and Algorithms)
30. COMP303 T2 2011 (Design and Analysis of Algorithms)
31. COMP261 T2 (Last 8 weeks) 2011 (Algorithms and Data Structures)
32. SWEN430 T2 (First 4 weeks) 2011 (Compiler Engineering)
33. SWEN423 T1 (First 2 weeks and last 4 weeks) 2011 (OO Paradigms)
34. COMP303 T2 2010 (Design and Analysis of Algorithms)
35. COMP261 T2 (2nd half) 2010 (Algorithms and Data Structures)
36. SWEN430 T2 (1st half) 2010 (Compiler Engineering)
37. COMP303 T2 (2nd half) 2009 (Design and Analysis of Algorithms)
38. COMP431 T2 (2nd half) 2009 (Compilers)
39. COMP304 T1 (1st half) 2009 (Programming Languages)
40. COMP462 T1 (1st half) 2009 (Object-Oriented Paradigms)
41. COMP303 T2 2008 (Design and Analysis of Algorithms)
42. COMP431 T1 2008 (Compilers)
43. COMP103 T2 (2nd half) 2007 (Introduction to Data Structures and Algorithms)
44. COMP471 T2 (1st half) 2007 (Special Topic: Compiler Technologies)
45. COMP462 T1 (1st half) 2007 (Object-Oriented Paradigms)
46. COMP103 T3 (2nd half) 2006 (Introduction to Data Structures and Algorithms)
47. COMP103 T2 (2nd half) 2006 (Introduction to Data Structures and Algorithms)
48. COMP101 T3 2005 (Introduction to Dynamic Web Development)
49. INET101 T3 2004 (Introduction to Internet Technology)

Students Supervised

In February 2022, I currently supervise 4 PhD students and 2 RA’s. My current and former 58 students are listed here: https://potanin.github.io/students.html and range from Research Assistants to Honours Students, Masters, and PhD’s.

Full Publications List

Book Chapters

1. James Noble, Alex Potanin, Toby Murray, Mark S. Miller. *Abstract and Concrete Data Types vs Object Capabilities*. In P. Müller and Ina Schaefer (Eds.): *Principled Software Development - Essays Dedicated to Arnd Poetzsch-Heffter on the Occasion of his 60th Birthday*. Springer, Heidelberg. 2018.
2. Alex Potanin, Johan Ostlund, Yoav Zibin, Michael D. Ernst. *Immutability*. In D. Clarke et al. (Eds.): *Aliasing in Object-Oriented Programming*, LNCS 7850, pp. 233-269. Springer, Heidelberg. 2013.

Journal Papers

3. Isaac Oscar Gariano, Marco Servetto, Alex Potanin. Using Capabilities for Strict Runtime Invariant Checking. Science of Computer Programming. (Accepted in September 2022).
4. Darya Melicher, Anlun Xu, Valerie Zhao, Alex Potanin, Jonathan Aldrich. *Bounded Abstract Effects*. ACM Transactions on Porgramming Languages and Systems, Volume 44, Issue 1. March 2022. Article No 5, pp 1 – 48.
5. Isaac Oscar Gariano, Marco Servetto, Alex Potanin, Hrshikesh Arora. Iteratively Composing Statically Verified Traits. VPT-2019 Post-Proceedings as a volume of Electronic Proceedings in Theoretical Computer Science (EPTCS). Issue 299, Paper 7.
6. Chris Male, David Pearce, Alex Potanin, and Constantine Dymnikov. *Formalisation and Implementation of an Algorithm for Bytecode Verification of @NonNull Types*. Science of Computer Programming. Volume 76, Issue 7, Pages 587 - 608, July 2011.
7. Alex Potanin, James Noble, Dave Clarke, and Robert Biddle. *Featherweight Generic Confinement*. Journal of Functional Programming. Volume 16, Number 6, Pages 793 - 811, September 2006.
8. Alex Potanin, James Noble, Marcus Frean, and Robert Biddle. *Scale-free Geometry in Object-Oriented Programs*. Communications of the ACM. Pages 99 - 103. May 2005. (ACM Link)
9. Alex Potanin, James Noble, and Robert Biddle. *Checking Ownership and Confinement*. Concurrency and Computation: Practice and Experience. Volume 16, Issue 7, Pages 671 - 687, 2004.

Refereed Conference Papers

10. Tobias Runge, Alexander Kittelmann, Marco Servetto, Alex Potanin, and Ina Schaefer. *Information Flow Control-by-Construction for an Object-Oriented Language*. In proceedings of SEFM 2022.
11. Tobias Runge, Alex Potanin, Thomas Thum, and Ina Schaefer. *Traits: Correctness-by-Construction for Free*. In proceedings of FORTE 2022. **Won FORTE 2022 Best Paper award**.
12. Manish Singh, Lindsay Groves, and Alex Potanin. *A Relaxed Balanced Lock-free Binary Tree*. In proceedings of PDCAT 2020.
13. Julian Mackay, Alex Potanin, Jonathan Aldrich, and Lindsay Groves. *Syntactically Restricting Bounded Polymorphism for Decidable Subtyping*. In proceedings of APLAS2020.

14. Julian Mackay, Alex Potanin, Jonathan Aldrich, and Lindsay Groves. *Decidable Subtyping for Path Dependent Types.* In proceedings of POPL2020. Additionally, the artifact successfully passed the Artifact Evaluation and was given Reusable Artifact badge.
15. Aaron Craig, Alex Potanin, Lindsay Groves and Jonathan Aldrich. *Capabilities: Effects for Free*. In proceedings of ICFEM2018. Pp 231-247. Springer.
16. Jens Dietrich, Kamil Jezek, Shawn Rasheed, Amjed Tahir, Alex Potanin. *EvilPickles: DoS attacks based on Object-Graph Engineering*. In proceedings of ECOOP2017. *Additionally, the artifact successfully passed the Artifact Evaluation. This paper won ECOOP 2017 Distinguished Artifact Award.*
17. Darya Melicher, Yangqingwei Shi, Alex Potanin, Jonathan Aldrich. *A Capability-Based Module System for Authority Control*. In proceedings of ECOOP2017. *Additionally, the artifact successfully passed the Artifact Evaluation.*
18. Garming Sam, Nicholas Cameron and Alex Potanin. *Automated Refactoring of Rust Programs*. Article No.: 14. In proceedings of ACSC2017.
19. Joseph Lee, Jonathan Aldrich, Troy Shaw, Alex Potanin. *A Theory of Tagged Objects*. In proceedings of ECOOP2015. Pp 999-1026. *Additionally, the artifact on the software section of my home page successfully passed the Artifact Evaluation.*
20. Cyrus Omar, Darya Kurilova, Ligia Nistor, Benjamin Chung, Alex Potanin, and Jonathan Aldrich. *Safely Composable Type-Specific Languages*. In proceedings of ECOOP2014. Pp 105-130. Springer-Verlag. *This paper won ECOOP 2014 Distinguished Paper Award.*
21. Marco Servetto, Julian Mackay, Alex Potanin, and James Noble. *The Billion-Dollar Fix: Safe Modular Circular Initialisation with Placeholders and Placeholder Types*. In proceedings of ECOOP2013. Pp 205-229. Springer-Verlag.
22. Constantine Dymnikov, David Pearce and Alex Potanin. *OwnKit: Inferring Modularly Checkable Ownership Annotations for Java*. In proceedings of the Australasian Software Engineering Conference (ASWEC) 2013. Pp 181-190. IEEE.
23. Alex Potanin, Monique Damitio and James Noble. *Are Your Incoming Aliases Really Necessary? Counting the Cost of Object Ownership*. In proceedings of the International Conference on Software Engineering (ICSE) 2013. Pp 742-751. ACM/IEEE.
24. Daniel Atkins, Alex Potanin and Lindsay Groves. *The Design and Implementation of Clocked Variables in X10*. In proceedings of Australasian Computer Science Conference (ACSC) 2013. Pp 87-96. CRPIT.
25. Jan Larres, Alex Potanin and Yuichi Hirose. *A Study of Performance Variations in the Mozilla Firefox Web Browser*. In proceedings of Australasian Computer Science Conference (ACSC) 2013. Pp 3-12. CRPIT.
26. Hien Tran, Craig Anslow, Stuart Marshall, Alex Potanin, Mairead de Roiste. *Lessons Learnt from Collaboratively Creating Maps on a Touch Table*. In proceedings of the 12th Annual Conference of the New Zealand Chapter of the ACM Special Interest Group on Computer-Human Interaction (CHINZ) 2011. Pp. 105-108. ACM SIGCHI.
27. Yoav Zibin, Alex Potanin, Paley Li, Mahmood Ali, Michael D. Ernst. *Ownership and Immutability in Generic Java*. In proceedings of Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 2010. Pp. 598-617. ACM.
28. Radu Muschevici, Alex Potanin, Ewan Tempero, and James Noble. *Multiple Dispatch in Practice*. In proceedings of Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 2008. Pp. 563-582. ACM. (ACM Link)

29. Chris Male, David Pearce, Alex Potanin, and Constantine Dymnikov. *Java Bytecode Verification for @NonNull Types*. In proceedings of Compiler Construction (CC) 2008. Pp 229-244.
30. Neil Ramsay, Stuart Marshall, and Alex Potanin. *Annotating UI Architecture with Actual Use*. In proceedings of Australasian User Interface Conference (AUIC) 2008. Pp. 75-78. CRPIT.
31. Yoav Zibin, Alex Potanin, Mahmood Ali, Shay Artzi, Adam Kiezun, and Michael D. Ernst. *Object and Reference Immutability using Java Generics*. In proceedings of the Foundations of Software Engineering (FSE) Conference, Dubrovnik, Croatia, September 2007. Pp. 75-84. ACM. (Won ESEC/FSE 2007 ACM SIGSOFT Distinguished Paper Award.) (ACM Link)
32. Alex Potanin, James Noble, Dave Clarke, and Robert Biddle. *Generic Ownership for Generic Java*. In proceedings of the Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) Conference, Portland, Oregon, October 2006. Pp. 311-324. ACM. (ACM Link)
33. Alex Potanin, James Noble, and Robert Biddle. *Snapshot Query-Based Debugging*. In proceedings of the Australian Software Engineering Conference, Melbourne, Australia, April 2004. Pp 251-261.

Refereed Workshop Papers

34. Baptiste Pauget, David Pearce, and Alex Potanin. *Towards Compilation of an Imperative Language for FPGAs*. In VMIL 2018.
35. James Noble, Sophia Drossopoulou, Mark S Miller, Toby Murray and Alex Potanin. *Abstract Data Types in Object-Capability Systems*. In IWACO 2016.
36. Du Li, Alex Potanin, and Jonathan Aldrich. *Delegation vs Inheritance for Typestate Analysis*. In FTfJP 2015.
37. Darya Kurilova, Alex Potanin, and Jonathan Aldrich. *Wyvern: Impacting Software Security via Programming Language Design*. In PLATEAU 2014. Pages 57-58.
38. James Noble and Alex Potanin. *On Owners-as-Accessors*. In IWACO 2014.
39. Jonathan Aldrich, Cyrus Omar, Alex Potanin and Du Li. *Language-Based Architectural Control*. In IWACO 2014.
40. Cyrus Omar, Benjamin Chung, Darya Kurilova, Alex Potanin and Jonathan Aldrich. *Type-Directed, Whitespace-Delimited Parsing for Embedded DSLs*. In proceedings of the First Workshop on the Globalization of Domain Specific Languages (GlobalDSL) 2013. Pp 8-11.
41. Ligia Nistor, Darya Kurilova, Stephanie Balzer, Benjamin Chung, Alex Potanin and Jonathan Aldrich. *Wyvern: A Simple, Typed, and Pure Object-Oriented Language*. In Proceedings of the 5th Workshop on MechAnisms for SPEcialization, Generalization and inHeritance (MASPEGHI) 2013. Pp 9-16.
42. Marco Servetto, David Pearce, Lindsay Groves, and Alex Potanin. *Balloon Types for Safe Parallelisation over Arbitrary Object Graphs*. Workshop on Determinism and Correctness in Parallel Programming (WoDeT) 2013.
43. Daniel Atkins, Alex Potanin, Lindsay Groves. *Clocked References in X10*. Languages for the Multicore Era (LaME) 2012.
44. Julian Mackay, Hannes Mehnert, Alex Potanin, Lindsay Groves, Nicholas Cameron. *Encoding Featherweight Java with Assignment and Immutability using The Coq Proof*

Assistant. In the proceedings of Formal Techniques for Java-like Programs (FTfJP) 2012. Pp 11-19.

45. Yoav Zibin, Alex Potanin, Paley Li, Mahmood Ali, Michael D. Ernst. *Ownership and Immutability in Generic Java (OIGJ).* IBM Programming Languages and Development Environments Seminar 2010 (PLDE2010).
46. Paley Li, Stephen Nelson, and Alex Potanin. *Ownership for Relationships*. International Workshop on Aliasing, Confinement, and Ownership (IWACO) 2009. Article No. 8.
47. Paley Li, Alex Potanin, James Noble, and Lindsay Groves. *Towards Unifying Immutability and Ownership*. International Workshop on Aliasing, Confinement, and Ownership (IWACO) at ECOOP2008.
48. Christo Fogelberg, Alex Potanin, and James Noble. *Ownership Meets Java*. International Workshop on Aliasing, Confinement, and Ownership (IWACO) at ECOOP2007.
49. Alex Potanin, James Noble, Tian Zhao, Jan Vitek. *A High Integrity Profile for Memory Safe Programming in Real-time Java.* In proceedings of the 3rd workshop on Java Technologies for Real-time and Embedded Systems, San Diego, CA, USA, October 2005.
50. Alex Potanin, James Noble, Dave Clarke, Robert Biddle. *Featherweight Generic Ownership*. In proceedings of the workshop on Formal Techniques for Java-like Programs (FTfJP) in European Conference for Object-Oriented Programming, Glasgow, Scotland, July 2005.
51. Alex Potanin, James Noble, Dave Clarke, Robert Biddle. *Defaulting Generic Java to Ownership*. In proceedings of the workshop on Formal Techniques for Java-like Programs (FTfJP) in European Conference for Object-Oriented Programming, Oslo, Norway, June 2004.
52. Alex Potanin, James Noble, Dave Clarke, and Robert Biddle. *Featherweight Generic Confinement*. In proceedings of the International Workshop on Foundations of Object-Oriented Languages (FOOL), affiliated with Symposium on Principles of Programming Languages, Venice, Italy, January 2004.
53. James Noble, Robert Biddle, Ewan Tempero, Alex Potanin, and Dave Clarke. *Towards a Model of Encapsulation*. In proceedings of the International Workshop on Aliasing, Confinement and Ownership (IWACO) in European Conference for Object-Oriented Programming, Darmstadt, Germany, July 2003.
54. Alex Potanin and James Noble. *Checking Ownership and Confinement Properties*. In proceedings of the workshop on Formal Techniques for Java-like Programs (FTfJP) in European Conference for Object-Oriented Programming, Malaga, Spain, June 2002.

Edited Journals

55. Alex Potanin and Gail Murphy (Editors). *Special Issue on the 23rd Asia-Pacific Software Engineering Conference (APSEC) 2016.* Science of Computer Programming. Volume 163, 1 October 2018.
56. Alex Potanin (Editor). *Special Issue on New Object-Oriented Languages (NOOL) 2015.* Journal of Object Technology. Volume 16, no. 2 (April 2017).
57. Taso Viglas and Alex Potanin (Editors). *Special Issue on Computing: The Australasian Theory Symposium 2011.* International Journal on Foundations of Computer Science (IJFCS), 2013. Vol. No. 24, Issue No. 1.

58. Taso Viglas and Alex Potanin (Editors). *Special Issue on Computing: The Australasian Theory Symposium 2010.* Chicago Journal of Theoretical Computer Science (CJTCS), May 2011. Volume 2010.

Other Refereed Publications

59. Manish Singh, Lindsay Groves, Alex Potanin. *A Relaxed Balanced Non-Blocking Binary Search Tree*. Poster in International Conference on Parallel Processing (ICPP) 2019.
60. Isaac Oscar Gariano, Marco Servetto, Alex Potanin, Hrshikesh Arora. *Iteratively Composing Statically Verified Traits.* Extended Abstract at Seventh International Workshop on Verification and Program Transformation (VPT 2019).
61. Darya Melicher, Yangqingwei Shi, Valerie Zhao, Alex Potanin, and Jonathan Aldrich. *Using Object Capabilities and Effects to Build an Authority-Safe Module System.* Poster in HotSoS 2018.
62. Aaron Craig, Alex Potanin, Lindsay Groves, Jonathan Aldrich. *Capabilities and Effects.* In OCAP 2017.
63. Darya Melicher, Yangqingwei Shi, Valerie Zhao, Alex Potanin, and Jonathan Aldrich. *Using Object Capabilities and Effects to Build an Authority-Safe Module System*. In OCAP 2017.
64. Jonathan Aldrich and Alex Potanin. *Usably Expressing and Enforcing Design in Wyvern.* In NOOL 2017.
65. Jens Dietrich, Kamil Jezek, Shawn Rasheed, Amjed Tahir, Alex Potanin. *EvilPickles: DoS attacks based on Object-Graph Engineering (Artifact).* In DARTS.
66. Darya Melicher, Yangqingwei Shi, Alex Potanin, Jonathan Aldrich. *A Capability-Based Module System for Authority Control (Artifact).* In DARTS.
67. Jonathan Aldrich and Alex Potanin. *Delegation Revisited.* NOOL 2016.
68. Jonathan Aldrich and Alex Potanin. *Naturally Embedded DSLs.* DSLDI 2016.
69. Darya Kurilova, Alex Potanin, and Jonathan Aldrich. *Modules in Wyvern: Advanced Control over Security and Privacy.* Poster in HotSOS 2016. See last page here.
70. Joseph Lee, Jonathan Aldrich, Troy Shaw, and Alex Potanin. *A Theory of Tagged Objects (Artifact).* DARTS, Volume 1, Issue 1, 2015.
71. Cyrus Omar, Darya Kurilova, Ligia Nistor, Benjamin Chung, Alex Potanin, and Jonathan Aldrich. *Safely Composable Type-Specific Languages.* Poster in ECOOP2014.
72. Darya Kurilova, Cyrus Omar, Ligia Nistor, Benjamin Chung, Alex Potanin, Jonathan Aldrich. *Type-Specific Languages to Fight Injection Attacks.* Poster in HotSOS 2014.
73. Cyrus Omar, Benjamin Chung, Darya Kurilova, Ligia Nistor, Alex Potanin, Jonathan Aldrich. *Extensible Type-Driven Parsing for Embedded DSLs in Wyvern.* Parsing@SLE 2013.
74. Cyrus Omar, Benjamin Chung, Darya Kurilova, Ligia Nistor, Alex Potanin, Jonathan Aldrich. *Extensible Type-Driven Parsing for Embedded DSLs in Wyvern.* Poster in SPLASH2013.
75. Jonathan Aldirch, Benjamin Chung, Darya Kurilova, Cyrus Omar, Alex Potanin. *DSL support in Wyvern Language.* Domain Specific Language Design and Implementation (DSLDI) 2013.
76. Jan Larres, Alex Potanin, and Yuichi Hirose. *Performance Variance Evaluation on Mozilla Firefox.* In the proceedings of the NZCSRSC2011.
77. Mairead de Roiste, Hien Tran, and Alex Potanin. *What makes a map?* At Ireland Organisation for Geographic Information Conference (IRLOGI) 2010.

78. Chris Andreae, Donald Gordon, Alex Potanin, James Noble, Robert Biddle. *Terrier: Static Query-Based Debugging in Eclipse.* Poster in Conference on Object-Oriented Programming, Systems, Languages, and Applications, Vancouver, Canada, October 2004.
79. Alex Potanin. *Generic Ownership: Practical Ownership Control in Programming Languages*. In Doctoral Symposium Proceedings of Conference on Object-Oriented Programming, Systems, Languages, and Applications, Vancouver, Canada, October 2004.
80. Alex Potanin. *Practical Ownership Control in Programming Languages*. In Doctoral Symposium Proceedings of European Conference for Object-Oriented Programming, Oslo, Norway, June 2004.
81. Alex Potanin. *A Tool for Ownership and Confinement Analysis of the Java Object Graph*. Conference poster and student research competition entry in Object-Oriented Programming Systems, Languages, and Applications Conference, Seattle, USA, November 2002. The poster itself is available in Microsoft PowerPoint format here. This entry was awarded a second place in the competition in the graduate division, even though I was technically an undergraduate. The results are available here. (**Update:** This entry was accepted into the ACM Student Research Competition Grand Finals and can be found in HTML format here. It was awarded a second place overall in the undergraduate category as can be found here.)

Theses

82. Alex Potanin. *Generic Ownership - A Practical Approach to Ownership and Confinement in OO Programming Languages*. My PhD thesis, awarded in 2007.
83. Alex Potanin. *The Fox - A Tool for Object Graph Analysis*. Honours report which resulted in First Class Honours at Victoria University of Wellington in 2002. My supervisors were James Noble and Robert Biddle. The PDFLaTeX source is available here.