OOPSLE 2020: Open and Original Problems in Software Language Engineering*

http://oopsle.github.io/2020

Vadim Zaytsev[1,2] and Anya Helene Bagge[3]

[1] vadim@grammarware.net, http://grammarware.net
Raincode Labs, Brussels, Belgium
[2] Universiteit Twente, Enschede, The Netherlands

[3] anya@ii.uib.no, http://www.ii.uib.no/~anya
Bergen Language Design Laboratory, University of Bergen, Bergen, Norway

Workshop Description

Software languages are any artificial languages used in software development: for programming, markup, pretty-printing, modelling, transformation, data description, formal specification, evolution, requirements, etc. Software language engineering (SLE) is a research domain of systematic, disciplined and measurable approaches of development, evolution and maintenance of such languages. Many concerns of software language engineering are acknowledged by both forward and reverse software engineers: robust parsing of language cocktails, fact extraction from heterogeneous codebases, tool interfaces and interoperability, renovation of legacy systems, static and dynamic code analysis, language feature usage analysis, mining repositories and chrestomathies, library versioning and wrapping, etc. The SLE field is relatively new (its flagship conference existing since 2011) and has not yet produced a list of acknowledged open problems, like the Hilbert's problems [8] or the POPLmark Challenge [9]. This workshop is meant to expose hidden expertise in coping with unsolvable or unsolved problems which commonly remain unexposed in academic publications. The main focus of the workshop lies in identifying and formulating challenges in the software language engineering field — these challenges could be addressed later at venues of SPLASH, STAF, MoDELS, SANER, ICSME, ICSE, ESEC/FSE and others. It is by design a discussion platform, not a mini-conference.

The *fifth* international workshop on Open and Original Problems in Software Language Engineering (OOPSLE'20) followed the first four editions held at WCRE 2013 [1], CSMR-WCRE 2014 [2], SANER 2015 [3] and SANER 2016. Since OOPSLE also aims to serve as a think tank in contributing to the SLE Body of Knowledge (SLEBoK, http://slebok.github.io) [26], it can be seen related to and/or continuing the tradition of the SLEBoK workshops at GTTSE 2009 [6], SoTeSoLa 2012 [5], SLE 2012 [7] and SPLASH 2018 [25] and the initiatives planned at the Dagstuhl seminars 17342 [4] and 20343 [21].

The core contributors of the 2020 edition of OOPSLE were:

- **Friedrich Steimann** (Fernuniversität Hagen, Germany)
- **Alfonso Pierantonio** (Università degli Studi dell'Aquila, Italy)
- **Federico Tomassetti** (Strumenta, Italy)
- **Mikhail Barash** (University of Bergen, Norway)
- **Eric Van Wyk** (University of Minnesota, USA)
- **Eleni Constantinou** (TU/Eindhoven, The Netherlands)
- **Bernd Fischer** (Stellenbosch University, South Africa)
- **Jurriaan Hage** (Universiteit Utrecht, The Netherlands)

The workshop was held at the Zoom platform (creating a whole-day event there generously supported by Raincode Labs), with about 30 participants throughout the day, peaking at 35 and dropping to 20 around planned lunch breaks and coffee breaks. There was a mix of researchers, practitioners, educators and students.

Programme Committee

The following people served as reviewers for the post-proceedings submissions:

- **Mikhail Barash** (University of Bergen, Norway)
- **Jurriaan Hage** (Universiteit Utrecht, The Netherlands)
- **Friedrich Steimann** (Fernuniversität Hagen, Germany)
- **Federico Tomassetti** (Strumenta, Italy)
- **Markus Völter** (independent, Germany)
- **Vadim Zaytsev** (Universiteit Twente, The Netherlands)

Each submission was reviewed by three PC members. The discussion panel paper was assigned to participants who were the most active and vocal during the discussion.

Biographies

Vadim Zaytsev (Workshop Co-Chair)

http://grammarware.net

Associate Professor in software evolution at UTwente, previously a Chief Science Officer at Raincode Labs (the largest independent compiler company in the world), a Lecturer of software engineering at UvA, and a Researcher on grammars in a broad sense, model transformation and megamodelling at CWI, VU Amsterdam and Universität Koblenz. Favourite topics include convergence of grammars with different technological background, inconsistency management, improving maintainability of legacy systems, generative and transformational techniques, various forms of modelling. Founder and active contributor of SLPS, Grammar Zoo, BibSLEIGH, etc. Principal investigator of INTiMALS and CodeDiffNG.

Has acted as an editor in chief of SLEBoK (2017–2020), member of IFIP TC 2 WG 2.11 (since 2020), workshop co-organiser at OOPSLE/SLEBoK (2013–20) at WCRE/SANER/SPLASH, CoCoDo (2016–2020) at ⟨Programming⟩, MM-MDE (2015) at MoDELS; Chair of PC or AEC at ICPC (2020), SLE (2016), SATToSE (2014), WCRE (2013), WCN (2011–12); awardee at ICSME (2016), SCAM (2009, 2016, 2018), WCRE (2013), WLM (2011); a member of various committees for ASE, BX, DADA, DYLA, ECMFA, FlexMDE, GPCE, GTTSE, ICMT, ICPC, ICSE, ICSME, ITSLE, LDTA, MoDELS, MSR, PAME, SAC, SANER, SATToSE, SCAM, SLE, SQM, SRC, TechDebt, WCN, XM; keynote speaker at MLE (2019), BENEVOL (2019), SATToSE (2018), ICSME (2016); and a hackathon / publicity / social media chair at a number of events.

Anya Helene Bagge (Workshop Co-Chair)

https://www.uib.no/en/persons/Anya.Helene.Bagge

Research interests covering tools and formalisms for manipulating programs, integrating (lightweight) specification with programming, design, specification and implementation of programming languages. Her PhD on design of language constructs for increased flexibility and reliability, was followed with work on language description and implementation, and on specification-based testing. Current efforts are concentrated on developing the Magnolia programming language, on specification and composition of reusable program components, and on integrated programming environments, both for programming in new languages, and to support development of the languages themselves. She has recently also explored a novel tree-walking approach to model transformation (Nuthatch).

She has recently been co-organiser of OOPSLE (2013–16), NWPT (2012), LDTA (2012), program chair of SATToSE (2015), publicity chair of SLE (2015), poster co-chair of SLE (2012), tutorialist of SATToSE (2014), and PC member of WGP (2012–13), SLE (2011,15), WCRE (2013), LDTA (2011–12), etc.

Friedrich Steimann (Keynote Speaker)

https://www.fernuni-hagen.de/ps/team/friedrich.steimann.shtml

Full Professor in Fernuniversität Hagen, a well-known expert in programming languages, object-oriented programming, software modelling and programming systems. Graduated in 1991 in Universität Karlsruhe, earned a doctoral degree in 1995 at Technische Universität Wien, habilitated in 2000 at Universität Hannover. Worked as an external professor and substitute professor at Universität Hannover and Heinrich-Heine-Universität Düsseldorf, since 2004 at his present position of a PL/OOP Professor in Hagen.

A long standing contributor to the SLE community, a steering committee member of the SLE conference, he is known not only for his seminal works on aspect-oriented programming [13, 14], roles in object-oriented and conceptual modelling [11, 12, 15, 22], and refactoring [10, 16, 17, 23, 24], but also for having one of the best naming senses in the community, with titles like *Giving ACID to Programmers* [20], *Coding for the Code* [19] and *Fatal Abstraction* [18].

References

1. A. H. Bagge and V. Zaytsev. Open and Original Problems in Software Language Engineering. http://oopsle.github.io, Since 2013.
2. A. H. Bagge and V. Zaytsev. International Workshop on Open and Original Problems in Software Language Engineering (OOPSLE 2014). In S. Demeyer, D. Binkley, and F. Ricca, editors, *Proceedings of the Software Evolution Week (IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering), Workshop Descriptions (CSMR-WCRE 2014)*, page 478. IEEE, Feb. 2014. https://doi.org/10.1109/CSMR-WCRE.2014.6747223
3. A. H. Bagge and V. Zaytsev. Open and Original Problems in Software Language Engineering 2015 Workshop Report. *SIGSOFT Software Engineering Notes*, 40:32–37, May 2015. https://doi.org/10.1145/2757308.2757313
4. B. Combemale, R. Lämmel, and E. Van Wyk. SLEBOK: The Software Language Engineering Body of Knowledge. https://www.dagstuhl.de/17342, 2017.
5. J.-M. Favre. Research 2.0 Working Group at SoTeSoLa. https://github.com/SATToSE/SoTeSoLa2012/wiki/Research, 2012.
6. J.-M. Favre and D. Avrilionis. Research 2.0 and Software Engineering 2.0: How Community Engineering will Change our Worlds. http://gttse.wikidot.com/2009:research-2-0, 2009.
7. J.-M. Favre and J. Vinju. SL(E)BOK 2.0 @ SLE2012. http://www.sleconf.org/2012/SLEBOK_SLE2012.html, 2012.
8. D. Hilbert. Mathematical Problems. *Bulletin of the American Mathematical Society*, 33(4):433–479, 1902.
9. B. C. Pierce, P. Sewell, S. Weirich, and S. Zdancewic. It Is Time to Mechanize Programming Language Metatheory. In B. Meyer and J. Woodcock, editors, *Verified Software: Theories, Tools, Experiments*, volume 4171 of *LNCS*, pages 26–30. Springer Berlin Heidelberg, 2008. https://doi.org/10.1007/978-3-540-69149-5_3
10. J. von Pilgrim, B. Ulke, A. Thies, and F. Steimann. Model/Code Co-refactoring: An MDE Approach. In *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 682–687. IEEE, 2013. https://doi.org/10.1109/ASE.2013.6693133
11. F. Steimann. A Radical Revision of UML's Role Concept. In A. Evans, S. Kent, and B. Selic, editors, *Proceedings of the Third International Conference on the Unified Modeling Language: Advancing the Standard (UML)*, volume 1939 of *LNCS*, pages 194–209. Springer-Verlag, 2000. https://doi.org/10.1007/3-540-40011-7_14
12. F. Steimann. On the Representation of Roles in Object-Oriented and Conceptual Modelling. *Data & Knowledge Engineering*, 35(1):83–106, 2000. https://doi.org/10.1016/S0169-023X(00)00023-9
13. F. Steimann. Domain Models are Aspect Free. In L. C. Briand and C. Williams, editors, *Proceedings of the Eighth International Conference on Model Driven Engineering Languages and Systems (MoDELS)*, volume 3713 of *LNCS*, pages 171–185. Springer International Publishing, 2005. https://doi.org/10.1007/11557432_13
14. F. Steimann. The Paradoxical Success of Aspect-Oriented Programming. In P. L. Tarr and W. R. Cook, editors, *Proceedings of the 21th Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 481–497. ACM, 2006. https://doi.org/10.1145/1167473.1167514
15. F. Steimann. The Role Data Model Revisited. *Applied Ontology*, 2(2):89–103, 2007. http://content.iospress.com/articles/applied-ontology/ao029

16. F. Steimann. Refactoring Tools and Their Kin. In *Tutorial Lectures of the Fifth International Summer School on Grand Timely Topics in Software Engineering (GTTSE)*, pages 179–214. Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-60074-1_8
17. F. Steimann. Constraint-based refactoring. *ACM Transactions on Programming Languages and Systems*, 40(1):2:1–2:40, 2018. https://doi.org/10.1145/3156016
18. F. Steimann. Fatal Abstraction. In E. G. Boix and R. P. Gabriel, editors, *Proceedings of the SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, pages 125–130. ACM, 2018. https://doi.org/10.1145/3276954.3276966
19. F. Steimann and T. Kühne. Coding for the Code. *ACM Queue*, 3(10):44–51, 2005. https://doi.org/10.1145/1113322.1113336
20. F. Steimann and N. Kurowsky. Transactional Editing: Giving ACID to Programmers. In *Proceedings of the 12th International Conference on Software Language Engineering (SLE)*, pages 202–215. ACM, 2019. https://doi.org/10.1145/3357766.3359536
21. F. Steimann, R. Lämmel, and V. Zaytsev. Software Language Engineering Body of Knowledge (SLEBoK) Recap. `https://www.dagstuhl.de/20343`, 2020.
22. F. Steimann and F. U. Stolz. Refactoring to Role Objects. In R. N. Taylor, H. Gall, and N. Medvidović, editors, *Proceedings of the 33rd International Conference on Software Engineering (ICSE)*, pages 441–450. ACM, 2011. https://doi.org/10.1145/1985793.1985854
23. F. Steimann and J. von Pilgrim. Constraint-Based Refactoring with Foresight. In J. Noble, editor, *Proceedings of the 26th European Conference on Object-Oriented Programming (ECOOP)*, volume 7313 of *LNCS*, pages 535–559. Springer International Publishing, 2012. https://doi.org/10.1007/978-3-642-31057-7_24
24. F. Steimann and J. von Pilgrim. Refactorings without Names. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 290–293. ACM, 2012. https://doi.org/10.1145/2351676.2351726
25. E. Van Wyk and V. Zaytsev. Software Language Engineering Body of Knowledge Workshop. `https://2018.splashcon.org/track/slebok-2018`, 2018.
26. Zaytsev, V. (Ed.). Software Language Engineering Body of Knowledge. `http://slebok.github.io`, 2009–2020.