Open and Original Problems in Software Language Engineering

SANER 2015 Workshop Proposal

http://oopsle.github.io

Anya Helene Bagge, anya@ii.uib.no, http://www.ii.uib.no/~anya
Bergen Language Design Laboratory (BLDL), University of Bergen, Norway
Vadim Zaytsev, vadim@grammarware.net, http://grammarware.net,
Institute of Informatics, Universiteit van Amsterdam, The Netherlands

***Abstract*—Software languages comprise all kinds of artificial languages used in software development. 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 reverse engineers as well as by forward software engineers The SLE field is relatively new has not yet produced a list of acknowledged open problems. This workshop is meant to expose hidden expertise in coping with unsolvable or unsolved problems which commonly remain unexposed in academic publications. OOPSLE aims to serve as a think tank in selecting candidates for the open problem list, as well as other kinds of unconventional questions and definitions that do not necessarily have clear answers or solutions, thus facilitating the exposure of dark data. We also plan to formulate promising language-related challenges to organise in the future.**

Description

The third international workshop on Open and Original Problems in Software Language Engineering (OOPSLE'15) will follow the first two editions held at WCRE 2013 in Koblenz and at CSMR-WCRE 2014 in Antwerp. 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 like SLE, MoDELS, CSMR, WCRE, ICSM and others.

The field covered by the workshop, revolves around "software languages" — all kinds of artificial languages used in software development: for programming, markup, pretty-printing, modelling, data description, formal specification, evolution, etc. Software language engineering is a relatively new 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 reverse engineers as well as by forward 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.

Some research fields have a list of acknowledged open problems that are being slowly addressed by the community: as examples of such lists, we can recall the Hilbert's problems [5], the POPLmark Challenge [9] and a list of open problems in Boolean grammars [8]. However, the field of software language engineering has not yet produced one. This workshop is meant to expose hidden expertise in coping with unsolvable or unsolved problems which commonly remain unexposed in academic publications. OOPSLE aims to serve as a think tank in selecting candidates for the open problem list, as well as other kinds of unconventional questions and definitions that do not necessarily have clear answers or solutions, thus facilitating the exposure of dark data [4]. We also plan to formulate promising language-related challenges to organise in the future. Beside the abovementioned POPLmark Challenge which can also be seen as a collection of benchmarks, there have been many more contests, challenges and competitions related to software language engineering: LDTA Tool Challenge held at the LDTA workshop in 2011 [6], CodeGeneration-affiliated Language Workbench Challenge [3] held yearly since 2011, Transformation Tool Contest held six times since 2007 [10], Rewrite Engines Competition held three times in 2006, 2008 and 2010 at WRLA [2], PLT Games held monthly since December 2012 [7].

An extensive list of topics encouraged for investigation for workshop participants, can be found at our website and the corresponding description of its previous instances [1].

Format

Based on the interest in OOPSLE at CSMR-WCRE'14 and WCRE'13 and the result of its publicity, we expect around 20 participants and would like to hold a whole day workshop to leave room for discussion.

A. Before the workshop

OOPSLE participants are encouraged to submit position papers up to 4 pages in length, sketching an open or original problem, idea or challenge. The submissions are screened by the workshop chairs, who will select papers based on potential for discussion and interest to the community, as well as the clarity of presentation and motivation — *OOPSLE is not a mini-conference*, and therefore it is not necessary for the work to be conclusive yet. The papers will be posted online prior

to the workshop, so the participants have the opportunity to read them in advance.

B. At the workshop

Each accepted paper is presented at the workshop as a brief summary of its main idea and a set of open questions to be discussed with the audience. Presenters will ask for input on how to proceed with experiments, validation or refinement of their ideas, collect opinions on the presented definitions, share similar experience. We expect the participants to be friendly but inquisitive, and ask hard questions back that may lead to deepening the initially presented insights. The workshop is planned to have short presentations and long discussions to stimulate direct collaboration afterwards.

C. After the workshop

All workshop participants will be invited to submit a full paper to a special issue of the Electronic Communications of the EASST, an open access peer-reviewed journal. Journal submissions will undergo peer review by the members of the program committee consisting of researchers in software language engineering and reverse engineering.

Organisers

The following people share equal responsibility in shaping the theme of the workshop and attracting participants.

Anya Helene Bagge

My research interests are in tools and formalisms for manipulating programs, integration of (lightweight) specification and programming, and the design, specification and implementation of programming languages. I did my PhD on design of language constructs for increased flexibility and reliability, and since then I have followed up with work on language description and implementation, and on specification-based testing. My 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. I have recently also explored a novel tree-walking approach to model and program transformation. Long term, my research is focused on the question of how programming languages and tools can help the programmer in producing reliable, flexible, reusable, and maintainable code.

I have recently been co-organiser of OOPSLE (2013–14), NWPT (2012), poster co-chair of SLE (2012), organising co-chair of LDTA (2012), tutorialist of SATToSE (2014), and PC member of WGP (2012–13), SLE (2011), WCRE tool track (2013), and LDTA (2011–12).

Vadim Zaytsev

Last years works mostly with "grammars in a broad sense" as software language definitions and other kinds of commitments to structure. Favourite topics include convergence of such grammars with different technological background, inconsistency management, increasing maintainability, generative and transformational techniques. Fights "dark data" by exposing his own research process with open notebook science and writing extensive reports that include negative research results (cf. *The Grammar Hammer of 2012*, http://arxiv.org/abs/1212.4446). Populates and maintains the biggest grammar repository, Grammar Zoo, http://slps.github.io/zoo, at this day holding over 500 grammars in a broad sense.

Has acted as a workshop co-organiser at OOPSLE (2013–14); Program Chair at WCN (2011–12) and SATToSE (2014); Tool Track Chair at WCRE (2013); Hackathon Chair at SoTeSoLa (2012, reverse engineering and reengineering) and SATToSE (2013, repository evolution); Publicity or Social Media Chair at STAF (2015), MoDELS (2013), SLE (2011), GTTSE (2009–2011); weekly PEM Colloquium organiser (2012–13); PC member for SANER ERA (2015), CSMR-WCRE ERA (2014), DADA (2014), SCAM (2010–14), LDTA (2012), SQM (2012–14), DYLA (2010), WCN (2011–14), ACM SRC (2013), XM (2013–14); as well as an external reviewer for conferences (WCRE, LOPSTR, ESEC/FSE, CSMR, ECMFA, ICPC, MoDELS, SAC/PL, ICSTW, TOOLS, DSL, SCAM, SERP, ICSM), workshops (STSM, ATEM) and journals (SCP, EMSE, SoSym, IET Software).

References

[1] A. H. Bagge and V. Zaytsev, "Open and original problems in software language engineering," Oct. 2013. [Online]. Available: http://oopsle.github.io

[2] F. Durán, M. Roldán, J.-C. Bach, E. Balland, M. van den Brand, J. R. Cordy, S. Eker, L. Engelen, M. de Jonge, and K. T. Kalleberg, "The Third Rewrite Engines Competition," in *Eighth International Workshop on Rewriting Logic and Its Applications (WRLA)*, ser. Lecture Notes in Computer Science, P. C. Ölveczky, Ed., vol. 6381. Springer, 2010, pp. 243–261.

[3] S. Erdweg, T. van der Storm, M. Völter, M. Boersma, R. Bosman, W. R. Cook, A. Gerritsen, A. Hulshout, S. Kelly, A. Loh, G. Konat, P. J. Molina, M. Palatnik, R. Pohjonen, E. Schindler, K. Schindler, R. Solmi, V. Vergu, E. Visser, K. van der Vlist, G. Wachsmuth, and J. van der Woning, "The State of the Art in Language Workbenches. Conclusions from the Language Workbench Challenge," in *Proceedings of the Sixth International Conference on Software Language Engineering (SLE 2013)*, ser. Lecture Notes in Computer Science, M. Erwig, R. F. Paige, and E. Van Wyk, Eds., vol. 8225. Springer International Publishing Switzerland, Oct. 2013, in print.

[4] T. Goetz, "Freeing the Dark Data of Failed Scientific Experiments," *Wired Magazine*, vol. 15, no. 10, 2007.

[5] D. Hilbert, "Mathematical Problems," *Bulletin of the American Mathematical Society*, vol. 33, no. 4, pp. 433–479, 1902.

[6] LDTA 2011, "11th International Workshop on Language Descriptions, Tools and Applications. Tool Challenge," 2011. [Online]. Available: http://ldta.info/tool.html

[7] B. McKenna, "The Programming Language Theory Games: a monthly programming language competition," Dec. 2012. [Online]. Available: http://www.pltgames.com

[8] A. Okhotin, "Conjunctive and Boolean Grammars: The True General Case of the Context-Free Grammars," *Computer Science Review*, vol. 9, pp. 27–59, 2013. [Online]. Available: http://users.utu.fi/aleokh/boolean/nine_open_problems.html

[9] B. C. Pierce, P. Sewell, S. Weirich, and S. Zdancewic, "It Is Time to Mechanize Programming Language Metatheory," in *Verified Software: Theories, Tools, Experiments*, ser. Lecture Notes in Computer Science, B. Meyer and J. Woodcock, Eds. Springer Berlin Heidelberg, 2008, vol. 4171, pp. 26–30.

[10] A. Rensink and P. Van Gorp, "Graph Transformation Tool Contest 2008," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 12, no. 3–4, pp. 171–181, 2010.