Final Scientific Report SNF Project no. 200020-162352
“Agile Software Analysis”

February 20, 2019

1 Summary of results

Software developers actually spend much of their time not producing new code, but analysing the existing code base. Integrated Development Environments (IDEs) however are mostly glorified text editors, and offer only limited support for developers to query and analyse software systems. In this continuation of our SNF project *Agile Software Assessment*, we proceeded to explore new ways to enable developers to efficiently answer detailed questions about the software system under development.

Results

Agile Model Extraction

In this track we explored techniques to rapidly extract software models from unknown source code. The key result was the development of *bounded seas*, an approach to parsing that allows one to incrementally refine a parser that recognizes elements of interest within source code while ignoring others. [KVG$^+$16] [Kur16] [KVG$^+$18]. In contrast to older “island parsers”, bounded seas are not fragile to incremental changes, since the rules that determine what to ignore are derived from the rules describing what to recognize. The PetitParser framework, which served as a testbed for these ideas, is widely available[1], and has been ported to other languages, notable Java[2] and Dart.[3]

In addition to this work, we explored, with the help of several Bachelors and Masters students, several other promising directions, including rapidly building parsers from pre-existing building blocks [Fuc17] testing the limits of island parsing and parsing expression grammars with complex programming languages [Iya16] [Kub16] [Rüf16] and applying machine learning approaches to automatically recognize structural patterns in code [Wal16].

More recently we have started to explore how to exploit executable domain models as part of the software system under development, to keep requirements and implementation in sync [Pat18].

Context-Aware Tooling

In this track we explored ways to close the abstraction gap between code and application domains in IDEs.

We continued to explore the potential for *moldable tools*: software development tools that are designed to be easily tailored to specific application domains [CGK$^+$16b] [Chi16d] [Chi16e] [CGK$^+$17]. In particular, with the help of Bachelors and Masters students we developed a moldable object inspector that supports reproducible developer interactions [Kau18], a moldable context-aware search tool [CGK$^+$16a], and a moldable editor [Syr18]. Many of these innovations have been adopted in the Pharo[4] development environment, and others are part of the Glamorous Toolkit.[5]

[1] https://github.com/moosetechnology/PetitParser

[2] https://github.com/petitparser/java-petitparser

[3] https://pub.dartlang.org/packages/petitparser

[4] https://pharo.org

[5] https://gtoolkit.com

In related work, we have further explored how to support live debugging, particularly of concurrent threads [Ste17] [Cor16] [Les16] [LCN16] [LCN18].

We have also studied how to support software developers in selecting suitable visualizations to present and explore the results of software analyses. Based on the results of a systematic literature review of software visualizations [MGAN18]. We have developed tools to help developers explore and select suitable visualizations [MGN16a] [MGN$^+$16b] [MGN17] as well as dedicated tools to visualize and explore collaboration networks [MSGN16] [Sel16].

More recently we have studied the impact of the medium (*i.e.*, computer screen vs. physical 3D models vs. virtual reality) on the effectiveness of software visualizations [MFB$^+$17] [MGAN17] [Mer18]. We are continuing to explore the potential for augmented reality to enrich the experience of software developers by providing a separate medium to monitor software quality issues [MBN18] [Hes19].

Ecosystem Mining

Here we explored how information mined from the broader ecosystems of related software can be exploited to support analysis of a given software system under development.

The *Ecosystem Monitoring Framework* [Spa16c] [Spa16d] is a proof-of-concept infrastructure that can automate a variety of different kinds of analyses on a given corpus of software projects. The *Object Repository* [SGN16] is an experimental ecosystem analysis tool that extracts code snippets from the ecosystem that can create a variety of instances (*i.e.*, objects) from the application domain. These objects can then be used as input for other analyses, such as determining potential dynamic types.

In related research, we have investigated the effectiveness of numerous lightweight heuristics to infer types for dynamically-typed programming languages [MN16] [MGN17a] [MGN17b] [Mil17c] [Mil17d] and we have also explored the use of data extracted from the inline caches of virtual machines to rank the inferred types [MBGN18] [MBGN16].

In yet other work, we mined software ecosystems to infer the likelihood of the most common type of errors in Java code, namely "null pointer exceptions", which arise when an attempt is made to use an uninitialized variable. By analyzing what proportion of other clients guard against null errors, we can assess with high reliability the likelihood that a given piece of code is in error [Leu17] [LOGN17a] [LOGN17b] [Tra16].

Finally, we have started to explore the use of natural language processing to align developer questions with relevant software analyses [Ran18].

Evolutionary Monitoring

This track is concerned with monitoring of technical debt over the lifetime of a software project.

We have carried out various empirical studies to track the evolution of exception usage in Java projects [OCS$^+$17] [OCC$^+$17] [Hoh18]. We have confirmed that bad programming practices, in which null return values are abused to signal an error instead of raising an exception, are among the leading causes of bugs in Java programs [OLLN16]. We have extensively explored the use of machine learning techniques for bug prediction, and found that feature selection and hyperparameter tuning have a significant impact on the quality and efficiency of the results [Osm16a] [Osm16b] [OGN17b] [OGN17a] [OGNL17] [Osm17] [Bro18].

We have explored ways to improve feedback to developers of software quality issues by reporting such issues "just in time". The resulting tool, called QualityAssistant, is now a standard feature of the Pharo IDE [TGN16] [Tym17c] [TGN17] [Tym17f] [TGN18]. A dedicated 3D visualization reveals how code quality evolves over time [TMGN16] [STSB16].

Finally, we have studied the effectiveness of "code smell" detection rules to expose security concerns in the development of mobile apps [Gad17] [Fri18] [Spr18] [Ytr18] [CSGN18] [GGFN18] [GGN17] [HG18] [Haz18]. The results of this work will be integrated into the Android Studio IDE.

Staff contributions

This project was supervised by Oscar Nierstrasz (Full Professor) and Mohammad Ghafari (Postdoc).

Here we summarize the contributions of the project staff (PhD students) throughout the project. Note that we only report on the staff whose salaries were paid from this project.[6]

- Andrei Chiş worked extensively on the development of "moldable tools" [CGK+16a] [CGK+16b] [Chi16e] [CGK+17] [Les16] [LCN16] [LCN18]. He defended his PhD [Chi16d] in September 2016, and now works for feenk GmbH in Bern. We plan to continue to collaborate with him in the context of the followup SNSF project.
- Claudio Corrodi carried out some promising work on declarative breakpoints for live debugging [Cor16] and on benchmarking tools to detect Android security issues [CSGN18]. He terminated his doctoral studies in 2018, however, to pursue an industrial career.
- Pascal Gadient has been studying effective techniques to detect security issues on mobile apps, particularly related to data leaks [Gad17] [GGFN18] [GGN17]. He is expected to defend his PhD in the Autumn of 2021.
- Mohammadreza Hazhirpasand started his PhD studies in April 2018. He has also been studying security issues in mobile apps [HG18] [Haz18] and is currently carrying out an empirical study to assess the performance of Java developers in using cryptographic APIs. He is scheduled to defend his PhD in the Spring of 2022.
- Manuel Leuenberger has developed techniques to automatically detect the need for null checks in software based on ecosystem analysis [Leu17] [LOGN17a] [LOGN17b]. He is now expanding this work to tackle the broader problem of how to assist developers in migrating code to address API changes. He plans to defend his thesis in early 2021.
- Haidar Osman carried out research in improving the quality of automated bug prediction [OLLN16] [Osm16a] [Osm16b] [OGN17b] [OCS+17] [OGN17a] [OCC+17] [OGNL17]. He successfully defended his dissertation [Osm17] in December 2017. He now works as a Data Scientist for Swisscom in Bern. We are also collaborating with him in the followup SNSF project.
- Nitish Patkar started his PhD research in March 2018. He is studying the problem of how to improve the requirements elicitation process with the help of executable domain models [Pat18]. He will defend his thesis in early 2022.
- Pooja Rani (started January 2018)is studying ways to better support software developers in the IDE by correlating developer questions with the tasks at hand [Ran18]. She will defend her thesis at the end of 2021.
- Boris Spasojević carried out research on ecosystem monitoring [SGN16] [Spa16c]. Boris Spasojević defended his thesis [Spa16d] in December 2016. He now works as a Research Scientist at Oracle Labs in Baden. We are continuing to collaborate with him in the followup SNSF project.
- Yuriy Tymchuk worked on "just in time" quality feedback in the IDE [STSB16] [TMGN16] [TGN16] [Tym17c] [TGN17] [TGN18]. Yuriy Tymchuk defended his PhD [Tym17f] in December 2017. He now works as a Data Scientist at Swisscom in Bern.

Changes to the research plan and Important Events

There were no major deviations from the research plan. Claudio Corrodi interrupted his PhD studies, and has been replaced by Mohammadreza Hazhirpasand.

[6] In addition, Jan Kurš was paid from the predecessor project, ASA1 but from Canton in ASA2. He successfully defended his dissertation [Kur16] in October 2016, and now works for Google Zurich. Leonel Merino was supported from a Chilean doctoral grant. He defended his PhD [Mer18] in July 2018, and is now working at the Institute for Visualisation and Interactive Systems (VIS) at the University of Stuttgart. Nevena Milojković-Lazarević was paid from Canton. She defended her PhD [Mil17d] in June 2016, and now works as a Data Scientist at local.ch.

2 Research output

All reported publications are available electronically from the project's home page:

http://scg.unibe.ch/asa2

We also list selected PhD, Masters and Bachelors theses directly relevant to this project.

Journal papers

[GGFN18] Pascal Gadient, Mohammad Ghafari, Patrick Frischknecht, and Oscar Nierstrasz. Security code smells in Android ICC. *Empirical Software Engineering*, 2018. URL: http://scg.unibe.ch/archive/papers/Gadi18a.pdf, doi:10.1007/s10664-018-9673-y.

[GRPK17] Mohammad Ghafari, Konstantin Rubinov, and Mohammad Mehdi Pourhashem K. Mining unit test cases to synthesize API usage examples. *Journal of Software: Evolution and Process*, pages e1841–n/a, 2017. e1841 smr.1841. URL: http://dx.doi.org/10.1002/smr.1841, doi:10.1002/smr.1841.

[KVG+18] Jan Kurš, Jan Vraný, Mohammad Ghafari, Mircea Lungu, and Oscar Nierstrasz. Efficient parsing with parser combinators. *Science of Computer Programming*, 161:57.88, September 2018. URL: http://scg.unibe.ch/archive/papers/Kurs17a.pdf, doi:10.1016/j.scico.2017.12.001.

[LCN18] Max Leske, Andrei Chiş, and Oscar Nierstrasz. Improving live debugging of concurrent threads through thread histories. *Science of Computer Programming*, 161:122–148, 2018. URL: http://scg.unibe.ch/archive/papers/Lesk17a.pdf, doi:10.1016/j.scico.2017.10.005.

[MBGN18] Nevena Milojković, Clément Béra, Mohammad Ghafari, and Oscar Nierstrasz. Mining inline cache data to order inferred types in dynamic languages. *Science of Computer Programming, Elsevier, Special Issue on Adv. Dynamic Languages*, 161:105–121, 2018. URL: http://scg.unibe.ch/archive/papers/Milo17e.pdf, doi:10.1016/j.scico.2017.11.003.

[MGAN18] Leonel Merino, Mohammad Ghafari, Craig Anslow, and Oscar Nierstrasz. A systematic literature review of software visualization evaluation. *Journal of Systems and Software*, 144:165–180, October 2018. URL: http://scg.unibe.ch/archive/papers/Meri18a.pdf, doi:10.1016/j.jss.2018.06.027.

[MGN17] Leonel Merino, Mohammad Ghafari, and Oscar Nierstrasz. Towards actionable visualization for software developers. *Journal of Software: Evolution and Process*, 30(2):e1923–n/a, 2017. URL: http://scg.unibe.ch/archive/papers/Meri17a.pdf, doi:10.1002/smr.1923.

Conference papers

[CGK+16a] Andrei Chiş, Tudor Gîrba, Juraj Kubelka, Oscar Nierstrasz, Stefan Reichhart, and Aliaksei Syrel. Moldable, context-aware searching with Spotter. In *Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2016, pages 128–144, New York, NY, USA, 2016. ACM. URL: http://scg.unibe.ch/archive/papers/Chis16a-MoldableContextAwareSearchingWithSpotter.pdf, doi:10.1145/2986012.2986023.

[Chi16e] Andrei Chiş. Towards object-aware development tools. In *Companion Proceedings of the 2016 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity*, SPLASH Companion 2016, pages 65–66, New York, NY, USA, 2016. ACM. URL: http://scg.unibe.ch/archive/papers/Chis16e-ObjectAwareDevelopmentTools.pdf, doi:10.1145/2984043.2998546.

[CSGN18] Claudio Corrodi, Timo Spring, Mohammad Ghafari, and Oscar Nierstrasz. Idea: Benchmarking Android data leak detection tools. In Mathias Payer, Awais Rashid, and Jose M. Such, editors, *Engineering Secure Software and Systems*, pages 116–123, Cham, 2018. Springer International Publishing. URL: http://scg.unibe.ch/archive/papers/Corr18a.pdf, doi:10.1007/978-3-319-94496-8_9.

[GGN17] Mohammad Ghafari, Pascal Gadient, and Oscar Nierstrasz. Security smells in Android. In *17th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 121–130, September 2017. URL: http://scg.unibe.ch/archive/papers/Ghaf17c.pdf, doi:10.1109/SCAM.2017.24.

[HG18] Mohammadreza Hazhirpasand and Mohammad Ghafari. One leak is enough to expose them all — from a WebRTC IP leak to web-based network scanning. In *International Symposium on Engineering Secure Software and Systems (ESSoS 2018)*, pages 61–76, 2018. URL: http://scg.unibe.ch/archive/papers/Hazh18a.pdf, doi:10.1007/978-3-319-94496-8_5.

[LOGN17a] Manuel Leuenberger, Haidar Osman, Mohammad Ghafari, and Oscar Nierstrasz. Harvesting the wisdom of the crowd to infer method nullness in Java. In *Proceedings of the 17th International Working Conference on Source Code Analysis and Manipulation*, SCAM 2017. IEEE, 2017. URL: http://scg.unibe.ch/archive/papers/Leue17b-InferMethodNullness.pdf, doi:10.1109/SCAM.2017.22.

[LOGN17b] Manuel Leuenberger, Haidar Osman, Mohammad Ghafari, and Oscar Nierstrasz. KOWALSKI: Collecting API clients in easy mode. In *Proceedings of the 33rd International Conference on Software Maintenance and Evolution*, ICSME 2017. IEEE, 2017. URL: `http://scg.unibe.ch/archive/papers/Leue17c-KowalskiCollectingApi.pdf`, `doi:10.1109/ICSME.2017.78`.

[MBN18] Leonel Merino, Alexandre Bergel, and Oscar Nierstrasz. Overcoming issues of 3d software visualization through immersive augmented reality. In *VISSOFT'18: Proceedings of the 6th IEEE Working Conference on Software Visualization*, pages 54–64. IEEE, 2018. URL: `http://scg.unibe.ch/archive/papers/Meri18c.pdf`, `doi:10.1109/VISSOFT.2018.00014`.

[MFB+17] Leonel Merino, Johannes Fuchs, Michael Blumenschein, Craig Anslow, Mohammad Ghafari, Oscar Nierstrasz, Michael Behrisch, and Daniel Keim. On the impact of the medium in the effectiveness of 3D software visualization. In *VISSOFT'17: Proceedings of the 5th IEEE Working Conference on Software Visualization*, pages 11–21. IEEE, 2017. URL: `http://scg.unibe.ch/archive/papers/Meri17b.pdf`, `doi:10.1109/VISSOFT.2017.17`.

[MGAN17] Leonel Merino, Mohammad Ghafari, Craig Anslow, and Oscar Nierstrasz. CityVR: Gameful software visualization. In *ICSME'17: Proceedings of the 33rd IEEE International Conference on Software Maintenance and Evolution (TD Track)*, pages 633–637. IEEE, 2017. URL: `http://scg.unibe.ch/archive/papers/Meri17c.pdf`, `doi:10.1109/ICSME.2017.70`.

[MGN16a] Leonel Merino, Mohammad Ghafari, and Oscar Nierstrasz. Towards actionable visualisation in software development. In *VISSOFT'16: Proceedings of the 4th IEEE Working Conference on Software Visualization*. IEEE, 2016. URL: `http://scg.unibe.ch/archive/papers/Meri16a.pdf`, `doi:10.1109/VISSOFT.2016.10`.

[MGN+16b] Leonel Merino, Mohammad Ghafari, Oscar Nierstrasz, Alexandre Bergel, and Juraj Kubelka. MetaVis: Exploring actionable visualization. In *VISSOFT'16: Proceedings of the 4th IEEE Working Conference on Software Visualization*. IEEE, 2016. URL: `http://scg.unibe.ch/archive/papers/Meri16c.pdf`, `doi:10.1109/VISSOFT.2016.19`.

[MGN17a] Nevena Milojković, Mohammad Ghafari, and Oscar Nierstrasz. Exploiting type hints in method argument names to improve lightweight type inference. In *25th IEEE International Conference on Program Comprehension*, 2017. URL: `http://scg.unibe.ch/archive/papers/Milo17a.pdf`, `doi:10.1109/ICPC.2017.33`.

[MGN17b] Nevena Milojković, Mohammad Ghafari, and Oscar Nierstrasz. It's duck (typing) season! In *25th IEEE International Conference on Program Comprehension (ERA Track)*, 2017. URL: `http://scg.unibe.ch/archive/papers/Milo17b.pdf`, `doi:10.1109/ICPC.2017.10`.

[MN16] Nevena Milojković and Oscar Nierstrasz. Exploring cheap type inference heuristics in dynamically typed languages. In *Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward! 2016*, pages 43–56, New York, NY, USA, 2016. ACM. URL: `http://scg.unibe.ch/archive/papers/Milo16b.pdf`, `doi:10.1145/2986012.2986017`.

[OCC+17] Haidar Osman, Andrei Chiş, Claudio Corrodi, Mohammad Ghafari, and Oscar Nierstrasz. Exception evolution in long-lived Java systems. In *Proceedings of the 14th International Conference on Mining Software Repositories*, MSR '17, 2017. URL: `http://scg.unibe.ch/archive/papers/Osma17d-exception-evolution.pdf`, `doi:10.1109/MSR.2017.21`.

[OCS+17] Haidar Osman, Andrei Chiş, Jakob Schaerer, Mohammad Ghafari, and Oscar Nierstrasz. On the evolution of exception usage in Java projects. In *Proceedings of the 24rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 422–426, February 2017. URL: `http://scg.unibe.ch/archive/papers/Osma17b-exception-usage.pdf`, `doi:10.1109/SANER.2017.7884646`.

[OGNL17] Haidar Osman, Mohammad Ghafari, Oscar Nierstrasz, and Mircea Lungu. An extensive analysis of efficient bug prediction configurations. In *Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering*, PROMISE, pages 107–116, New York, NY, USA, 2017. ACM. URL: `http://scg.unibe.ch/archive/papers/Osma17f-EfficientBugPrediction.pdf`, `doi:10.1145/3127005.3127017`.

[OLLN16] Haidar Osman, Manuel Leuenberger, Mircea Lungu, and Oscar Nierstrasz. Tracking null checks in open-source Java systems. In *Proceedings of the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, March 2016. URL: `http://scg.unibe.ch/archive/papers/Osma16a.pdf`, `doi:10.1109/SANER.2016.57`.

[STSB16] Teseo Schneider, Yuriy Tymchuk, Ronie Salgado, and Alexandre Bergel. CuboidMatrix: Exploring dynamic structural connections in software components using space-time cube. In *VISSOFT'16: Proceedings of the 4th IEEE Working Conference on Software Visualization*, pages 116–125. IEEE, 2016. URL: `http://scg.unibe.ch/archive/papers/Schn16a.pdf`, `doi:10.1109/VISSOFT.2016.17`.

[TGN18] Yuriy Tymchuk, Mohammad Ghafari, and Oscar Nierstrasz. JIT feedback — what experienced developers like about static analysis. In *26th IEEE International Conference on Program Comprehension (ICPC 2018)*, pages 64–73, 2018. URL: `http://scg.unibe.ch/archive/papers/Tymc18a.pdf`, `doi:10.1145/3196321.3196327`.

[TMGN16] Yuriy Tymchuk, Leonel Merino, Mohammad Ghafari, and Oscar Nierstrasz. Walls, pillars and beams: A 3d decomposition of quality anomalies. In *VISSOFT'16: Proceedings of the 4th IEEE Working Conference on Software Visualization*, pages 126–135. IEEE, 2016. URL: `http://scg.unibe.ch/archive/papers/Tymc16a.pdf`, `doi:10.1109/VISSOFT.2016.9`.

[VGDB18] Sten Vercammen, Mohammad Ghafari, Serge Demeyer, and Markus Borg. Goal-oriented mutation testing with focal methods. In *Proceedings of the 9th ACM SIGSOFT International Workshop on Automating TEST Case Design, Selection, and Evaluation*, A-TEST 2018, pages 23–30, New York, NY, USA, 2018. ACM. URL: `http://doi.acm.org/10.1145/3278186.3278190`, `doi:10.1145/3278186.3278190`.

International Workshop papers

[CGK+16b] Andrei Chiş, Tudor Gîrba, Juraj Kubelka, Oscar Nierstrasz, Stefan Reichhart, and Aliaksei Syrel. Exemplifying moldable development. In *Proceedings of the Programming Experience 2016 (PX/16) Workshop*, PX/16, pages 33–42, New York, NY, USA, 2016. ACM. URL: `http://scg.unibe.ch/archive/papers/Chis16b-ExemplifyingMoldableDevelopment.pdf`, `doi:10.1145/2984380.2984385`.

[Cor16] Claudio Corrodi. Towards efficient object-centric debugging with declarative breakpoints. In *Post-proceedings of the 9th Seminar on Advanced Techniques and Tools for Software Evolution (SATToSE 2016)*, volume 1791. CEUR, July 2016. URL: `http://scg.unibe.ch/archive/papers/Corr16b-DeclarativeBreakpoints.pdf`.

[Haz18] Mohammadreza Hazhirpasand. Mheye: A hybrid android security assessment tool for ordinary users. In *SATTOSE*, SATToSE '18. SATToSE, 2018. URL: `http://scg.unibe.ch/archive/papers/Hazh18b.pdf`.

[KVG+16] Jan Kurš, Jan Vraný, Mohammad Ghafari, Mircea Lungu, and Oscar Nierstrasz. Optimizing parser combinators. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2016)*, pages 1:1–1:13, 2016. URL: `http://scg.unibe.ch/archive/papers/Kurs16a-Compiler.pdf`, `doi:10.1145/2991041.2991042`.

[LCN16] Max Leske, Andrei Chiş, and Oscar Nierstrasz. A promising approach for debugging remote promises. In *Proceedings of the International Workshop on Smalltalk Technologies*, IWST' 16, pages 18:1–18:9, 2016. URL: `http://scg.unibe.ch/archive/papers/Lesk16b.pdf`, `doi:10.1145/2991041.2991059`.

[MBGN16] Nevena Milojković, Clément Béra, Mohammad Ghafari, and Oscar Nierstrasz. Inferring types by mining class usage frequency from inline caches. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2016)*, pages 6:1–6:11, 2016. URL: `http://scg.unibe.ch/archive/papers/Milo16a.pdf`, `doi:10.1145/2991041.2991047`.

[Mil17c] Nevena Milojković. Improving the precision of type inference algorithms with lightweight heuristics. In *SATToSE'17: Pre-Proceedings of the 10th International Seminar Series on Advanced Techniques & Tools for Software Evolution*, June 2017. URL: `http://scg.unibe.ch/archive/papers/Milo17c.pdf`.

[MSGN16] Leonel Merino, Dominik Seliner, Mohammad Ghafari, and Oscar Nierstrasz. Communityexplorer: A framework for visualizing collaboration networks. In *Proceedings of International Workshop on Smalltalk Technologies (IWST 2016)*, pages 2:1–2:9, 2016. URL: `http://scg.unibe.ch/archive/papers/Meri16b.pdf`, `doi:10.1145/2991041.2991043`.

[OGN17a] Haidar Osman, Mohammad Ghafari, and Oscar Nierstrasz. Automatic feature selection by regularization to improve bug prediction accuracy. In *1st International Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE 2017)*, pages 27–32, February 2017. URL: `http://scg.unibe.ch/archive/papers/Osma17c-bugPredictionRegularization.pdf`, `doi:10.1109/MALTESQUE.2017.7882013`.

[OGN17b] Haidar Osman, Mohammad Ghafari, and Oscar Nierstrasz. Hyperparameter optimization to improve bug prediction accuracy. In *1st International Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE 2017)*, pages 33–38, February 2017. URL: `http://scg.unibe.ch/archive/papers/Osma17a.pdf`, `doi:10.1109/MALTESQUE.2017.7882014`.

[Osm16a] Haidar Osman. Against the mainstream in bug prediction. In *Extended Abstracts of the Ninth Seminar on Advanced Techniques and Tools for Software Evolution (SATToSE 2016)*, July 2016. http://sattose.wdfiles.com/local–files/2016:alltalks/SATTOSE2016_paper_2.pdf. URL: `http://scg.unibe.ch/archive/papers/Osma16b.pdf`.

[Osm16b] Haidar Osman. On the non-generalizability in bug prediction. In *Post Proceedings of the Ninth Seminar on Advanced Techniques and Tools for Software Evolution (SATToSE 2016)*, 2016. URL: `http://ceur-ws.org/Vol-1791/paper-03.pdf`.

[Pat18] Nitish Patkar. Towards executable domain models. In *Seminar Series on Advanced Techniques & Tools for Software Evolution (SATToSE)*, 2018. URL: `http://scg.unibe.ch/archive/papers/Patk18b.pdf`.

[Ran18] Pooja Rani. Software analysis using natural language queries. In *Seminar Series on Advanced Techniques & Tools for Software Evolution (SATToSE)*, 2018. URL: `http://scg.unibe.ch/archive/papers/Rani18a.pdf`.

[SGN16] Boris Spasojević, Mohammad Ghafari, and Oscar Nierstrasz. The Object Repository, pulling objects out of the ecosystem. In *Proceedings of the 11th Edition of the International Workshop on Smalltalk Technologies*, IWST'16, pages 7:1–7:10, New York, NY, USA, 2016. ACM. URL: `http://scg.unibe.ch/archive/papers/Spas16b.pdf`, `doi:10.1145/2991041.2991048`.

[Spa16c] Boris Spasojević. Building ecosystem-aware tools using the ecosystem monitoring framework. In *Post-proceedings of the 9th Seminar on Advanced Techniques and Tools for Software Evolution (SATToSE 2016)*, volume 1791. CEUR, July 2016. URL: `http://scg.unibe.ch/archive/papers/Spas16c.pdf`.

[TGN16] Yuriy Tymchuk, Mohammad Ghafari, and Oscar Nierstrasz. When QualityAssistant meets Pharo: Enforced code critiques motivate more valuable rules. In *IWST '16: Proceedings of International Workshop on Smalltalk Technologies*, pages 5:1–5:6, 2016. URL: `http://scg.unibe.ch/archive/papers/Tymc16b.pdf`, `doi:10.1145/2991041.2991046`.

[TGN17] Yuriy Tymchuk, Mohammad Ghafari, and Oscar Nierstrasz. Renraku — the one static analysis model to rule them all. In *IWST'17: Proceedings of International Workshop on Smalltalk Technologies*, 2017. URL: `http://scg.unibe.ch/archive/papers/Tymc17d.pdf`, `doi:10.1145/3139903.3139919`.

[Tym17c] Yuriy Tymchuk. The false false positives of static analysis. In *SATToSE'17: Pre-Proceedings of the 10th International Seminar Series on Advanced Techniques & Tools for Software Evolution*, June 2017. URL: `http://scg.unibe.ch/archive/papers/Tymc17c.pdf`.

Book chapters

[CGK$^+$17] Andrei Chiş, Tudor Gîrba, Juraj Kubelka, Oscar Nierstrasz, Stefan Reichhart, and Aliaksei Syrel. Moldable tools for object-oriented development. In Bertrand Meyer Manuel Mazzara, editor, *PAUSE: Present And Ulterior Software Engineering*, pages 77–101. Springer, Cham, 2017. URL: `http://scg.unibe.ch/archive/papers/Chis17a-MoldableToolsPAUSE.pdf`, `doi:10.1007/978-3-319-67425-4_6`.

Selected PhD, MSc and Bachelors Theses

[Bro18] Sébastien O. Broggi. Bug prediction with neural nets using regression- and classification-based approaches. Bachelor's thesis, University of Bern, January 2018. URL: `http://scg.unibe.ch/archive/projects/Brog18a.pdf`.

[Chi16d] Andrei Chiş. *Moldable Tools*. PhD thesis, University of Bern, September 2016. URL: `http://scg.unibe.ch/archive/phd/chis-phd.pdf`.

[Cur18] Simon Curty. Issue report assessment — assessment of issue report quality and class through natural language processing. Bachelor's thesis, University of Bern, February 2018. URL: `http://scg.unibe.ch/archive/projects/Curt18a.pdf`.

[Egg18] Markus Eggimann. Software testing in industry — assessing unit testing in an industrial software project. Bachelor's thesis, University of Bern, September 2018. URL: `http://scg.unibe.ch/archive/projects/Eggi18a.pdf`.

[Fri18] Patrick Frischknecht. Security in Android ICC. Bachelor's thesis, University of Bern, June 2018. URL: `http://scg.unibe.ch/archive/projects/Fris18a.pdf`.

[Fuc17] Mathias Fuchs. Recognising structural patterns in code — a parser based approach. Bachelor's thesis, University of Bern, May 2017. URL: `http://scg.unibe.ch/archive/projects/Fuch17a.pdf`.

[Gad17] Pascal Gadient. Security in Android applications. Masters thesis, University of Bern, August 2017. URL: `http://scg.unibe.ch/archive/projects/Gadi17.pdf`.

[Hes19] Mario Hess. Pervasive visualization in immersive augmented reality for software performance monitoring. Bachelor's thesis, University of Bern, February 2019. URL: `http://scg.unibe.ch/archive/projects/Hess19a.pdf`.

[Hoh18] Andreas Hohler. Big commit analysis — towards an infrastructure for commit analysis. Masters thesis, University of Bern, January 2018. URL: `http://scg.unibe.ch/archive/masters/Hohl18a.pdf`.

[Iya16] Rathesan Iyadurai. Parsing Ruby with an island parser. Bachelor's thesis, University of Bern, April 2016. URL: `http://scg.unibe.ch/archive/projects/Iyad16a-ParsingRuby.pdf`.

[Kau18] Mario Kaufmann. Reproducible moldable interactions. Masters thesis, University of Bern, April 2018. URL: `http://scg.unibe.ch/archive/masters/Kauf18a.pdf`.

[Kub16] Milan Kubicek. F# parsing expression grammar. Bachelor's thesis, University of Bern, August 2016. URL: `http://scg.unibe.ch/archive/projects/Kubi16a.pdf`.

[Kur16] Jan Kurš. *Parsing For Agile Modeling*. PhD thesis, University of Bern, October 2016. URL: `http://scg.unibe.ch/archive/phd/kurs-phd.pdf`.

[Les16] Max Leske. Improving live debugging of concurrent threads. Masters thesis, University of Bern, August 2016. URL: `http://scg.unibe.ch/archive/masters/Lesk16a.pdf`.

[Leu17] Manuel Leuenberger. Nullable method detection — inferring method nullability from API usage. Masters thesis, University of Bern, February 2017. URL: `http://scg.unibe.ch/archive/masters/Leue17a.pdf`.

[Mer18] Leonel Merino. *The Medium of Visualization for Software Comprehension*. PhD thesis, University of Bern, June 2018. URL: `http://scg.unibe.ch/archive/phd/merino-phd.pdf`.

[Mil17d] Nevena Milojković. *Augmenting Type Inference with Lightweight Heuristics*. PhD thesis, University of Bern, June 2017. URL: `http://scg.unibe.ch/archive/phd/lazarevic-phd.pdf`.

[Osm17] Haidar Osman. *Empirically-Grounded Construction of Bug Prediction and Detection Tools*. PhD thesis, University of Bern, December 2017. URL: `http://scg.unibe.ch/archive/phd/osman-phd.pdf`.

[Rüf16] Michael Rüfenacht. Error handling in PEG parsers — local error recovery for PetitParser. Masters thesis, University of Bern, August 2016. URL: `http://scg.unibe.ch/archive/masters/Ruef16a.pdf`.

[Sel16] Dominik Seliner. EggShell — a workbench for modeling scientific communities. Bachelor's thesis, University of Bern, August 2016. URL: `http://scg.unibe.ch/archive/projects/Seli16a.pdf`.

[Spa16d] Boris Spasojević. *Developing Ecosystem-aware Tools*. PhD thesis, University of Bern, December 2016. URL: `http://scg.unibe.ch/archive/phd/spasojevic-phd.pdf`.

[Spr18] Timo Spring. Benchmarking android data leak detection tools. Bachelor's thesis, University of Bern, August 2018. URL: `http://scg.unibe.ch/archive/projects/Spri18a.pdf`.

[Ste17] Roger Stebler. An empirical investigation into the usage of a live debugger. Masters thesis, University of Bern, June 2017. URL: `http://scg.unibe.ch/archive/masters/Stebl17a.pdf`.

[Syr18] Aliaksei Syrel. The moldable editor. Bachelor's thesis, University of Bern, February 2018. URL: `http://scg.unibe.ch/archive/projects/Syre18a.pdf`.

[Tra16] Lina Tran. NullSpy – an approach to pinpoint the origin location of a null reference. Bachelor's thesis, University of Bern, December 2016. URL: `http://scg.unibe.ch/archive/projects/Tran16a-NullSpy.pdf`.

[Tym17f] Yuriy Tymchuk. *Quality-Aware Tooling*. PhD thesis, University of Bern, December 2017. URL: `http://scg.unibe.ch/archive/phd/tymchuk-phd.pdf`.

[Wal16] Cédric Walker. Recognising structural patterns in code — a k-means clustering approach. Bachelor's thesis, University of Bern, August 2016. URL: `http://scg.unibe.ch/archive/projects/Walk16a.pdf`.

[Ytr18] Astrid Ytrehorn. Smelly APIs in Android ICC analysis of source code and relevant metadata. Bachelor's thesis, University of Bern, September 2018. URL: `http://scg.unibe.ch/archive/projects/Ytre18a.pdf`.