Christian Newman

PhD Candidate

Department of Computer Science
Kent State University
Kent, OH 44240

Cell: -
Email: cnewman@kent.edu
URL: www.cs.kent.edu/~cnewman

Education

PhD. Computer Science	Kent State University	Summer 2017 (expected)
M.S. Computer Science	Kent State University	Fall 2013
B.S. Computer Science	Kent State University	Fall 2010

Academic Experience

- **Graduate Research Assistantship,** Department of Computer Science, Kent State University, Kent, Ohio. 01/16 – present, Funded by the National Science Foundation, CNS 13-05292/05217.
- **Graduate Research Assistantship,** Department of Computer Science, Kent State University, Kent, Ohio. 05/15 – 08/15, Funded by the National Science Foundation, CNS 13-05292/05217.
- **Teaching Assistantship**, Department of Computer Science, Kent State University, Kent, Ohio. (08/12-05/15).
- **Graduate Research Assistantship**, Department of Computer Science, Kent State University, Kent, Ohio. 05/12 – 08/12, Funded by ABB inc.
- **Graduate Research Assistantship**, Department of Computer Science, Kent State University, Kent, Ohio. 07/10 – 05/12, Funded by the National Science Foundation MRI-R2 CNS 09-59924.

Non-Academic Experience

- ABB Engineering and Research intern; wrote a wrapper around the srcML framework in C# as well as a web-based query builder for communicating between arbitrary database REST APIs and a UI. Additionally, I wrote a wrapper around srcML's c++ library to adapt srcML for us in C# for use at ABB. The project can be found here: https://github.com/abb-iss/SrcML.NET. Employment dates: Aug 17th 2015 – Feb 17th 2016
- Kent State University IS as a Student Technician. Computer hardware and software end-user support, group policy management, software distribution. Employment dates: Aug. 2008 - Jun 2010

Advisor

Dr. Jonathan I. Maletic (2010 – Present)

Research Interests & Statement

Software engineering, maintenance and evolution; specifically, program transformation, static analysis, program slicing, and program comprehension

Program Transformation

The work on Normalizing-Restructurings [2] simplifies the task of writing transformation scripts by significantly reducing the number of syntax isomorphisms (via pre-processing transformation step). Further research and development [1] are currently aimed at developing a language to support transformation tasks using srcML; an empirical study on usage of transformation technology in industry and in research; and methods for automatic generation of transformation scripts.

Program Comprehension
Published research [3, 6] investigates the use and meaning of identifiers. It is focused on discovering, statically, the role of an identifier in the context of the body of code it appears in and correlating this role with its name. This research will be continued by further investigating the patterns presented in [3] via user-study, and applying the patterns to solve problems in program comprehension, naming consistency/recommendation, and more. As it stands, the technique involves re-documentation of srcML archives with parts-of-speech-like markers. Currently, a framework is in development to extend this functionality to English parts-of-speech as well, to assist other researchers in program comprehension and to help compare/contrast the technique given in [3] with English parts of speech.

Program Slicing
srcSlice implements a static, forward slicing technique and is written on top of the srcML framework. Currently, it is still being developed and is open-sourced for free use in other projects. Application of the srcSlice tool to helping find higher-order relationships between patterns presented in [3] is planned.

Static Analysis
Several projects feed into transformation and comprehension, but stand well on their own.

Software differences are used to investigate automatically detecting adaptive maintenance commits within software repositories. The goal is to find examples of adaptive maintenance changes to use as a golden set for future work as well as support program transformation research and possible detection of other forms of maintenance.

The TreeMiner algorithm, along with srcML archives, is used to investigate API usage and evolution patterns. This can help with comprehension tasks as well as to inform API developers of how people are using their API.

A framework was constructed to help decrease the time needed to construct tools for analyzing srcML's syntax tree, meaning many types of research requiring static program information are available. Additionally, this framework, and many of the tools that use it are open source; other researchers are free to use it.

Awards, Funding, and Other Support
- **ABB Stipend** – Travel support to ICSME 2015 (~1600$)
- **Best Presentation Award** - 30th Annual Graduate Research Symposium 2015
- **NSF Travel Grant** - Travel support to ICSM '11 (750$)
- **NSF REU** - Research Experience for Undergraduates (5000$)
- **NSF S-Stem Scholarship** – Undergrad scholarship for science, technology, engineering and mathematics (5000$) – 2009-2010

Publications and Scholarly Work
Works in Progress
- Newman, C.D., Collard, M.L., Maletic, J.I. "***Why Do Developers Avoid Automated Refactorings*?**", to be submitted.
- Decker, M.J., Newman, C.D., NA Kraft, Collard, M.L., Maletic, J.I. "***How Do Source Code Stereotypes Evolve*?**", to be submitted.
- AlAbed AlHaq, A., Bartman, B., Newman, C.D., Decker, M.J., Collard, M.L., Maletic, J.I. "***Automatically Detecting API Usage Patterns with TreeMiner***", to be submitted.
- Meqdadi, O., Alhindawi, N., Newman, C.D., Decker, M.J., Collard, M.L., Maletic, J.I. "***Empirically Mining Software Repositories for Adaptive Change Commits***", to be submitted
- Newman, C.D., Decker, M.J., Collard, M.L., Maletic, J.I. "***Using Differences to Automatically Detect Adaptive Maintenance Commits***", to be submitted.

Research Publications

1. Bartman, B., Newman, C. D., Collard, M.L., Maletic, J.I. " **srcQL: A Syntax-Aware Query Language for Source Code**", in the *Proceedings of 24th IEEE International Conference on Software Analysis, Evoluation, and Reengineering* (SANER '17), (Tool Demonstrations Track) Klagenfurt, Austria, Feb. 20-24, 2017, 5 pages.
2. Newman, C.D., Bartman, B., Collard, M.L., Maletic, J.I., "**Simplifying the Construction of Source Code Transformations via Automatic Syntactic Restructurings**", *Journal of Software Evolution and Process*, Vol. ?, No. ?, Accepted Sept. 6, 2016, 28 pages, (to appear)
3. Newman, C. D., Newman, Alsuhaibani, R., Collard, M.L., Maletic, J.I., **"Lexical Categories for Source Code Identifiers**", *in the Proceedings of the 24th IEEE International Conference on Software Analysis, Evolution, and Reengineering* (SANER'17), Klagenfurt, Austria, Feb. 20-24, 2017, 12 pages
4. Newman, C.D., Michael L. Collard, and Jonathan I. Maletic. 2016. "**srcType: A Tool for Efficient Static Type Resolution**", in *Proceedings of the 32nd International Conference on Software Maintenance and Evolution* (ICSME '16). IEEE, Raleigh, NC, USA. 2 pages.
5. Newman, C.D., Tessandra Sage, Michael L. Collard, Hakam W. Alomari, and Jonathan I. Maletic. 2016. "**srcSlice: a tool for efficient static forward slicing**", in *Proceedings of the 38th International Conference on Software Engineering Companion* (ICSE '16). ACM, New York, NY, USA, 621-624.
6. R. S. Alsuhaibani, C. D. Newman, M. L. Collard and J. I. Maletic, "**Heuristic-based part-of-speech tagging of source code identifiers and comments",** *Mining Unstructured Data (MUD), 2015 IEEE 5th Workshop on*, Bremen, 2015, pp. 1-6.
7. Alali, A., Bartman, B., Newman, C.D., Maletic, J.I., "**A Preliminary Investigation of Using Age and Distance Measures in the Detection of Evolutionary Couplings**" in the Proceedings of the ACM International Working Conference on Mining Software Repositories (MSR'13), San Francisco, California, May 18-19, 2013, pp. 169-172.
8. Maletic, J.I., Mosora, D.J., Newman, C.D., Collard, M.L., Sutton, A., Robinson, B.P., (2011), "**MosaiCode: Visualizing Large Scale Software: A Tool Demonstration**", in the Proceedings of the IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT'11), Williamsburg, VA, USA, Sept 31 – Oct 1, pp.

Online Publications

1. C.D Newman., M.J.Decker. Feb. 12th, 2013. srcML (Wikipedia Page) [Online]. http://en.wikipedia.org/wiki/SrcML

Software Systems Developed

- srcSlice – A program slicer https://github.com/srcML/srcSlice
- srctype – A tool for static type resolution https://github.com/srcML/srcType
- srcTL – A domain-specific language for program transformation (currently under development)
- Static analysis tool which tags identifiers with lexical category as described in *Lexical Categories for Source Code Identifiers* (not yet open-sourced)
- Event-Driven dispatcher framework to assist in the construction of srcML tools -- https://github.com/srcML/srcSAXEventDispatch

Teaching & Mentoring

Course Title/Duties	Terms/Dates	Institution
Computer Science 2 (data structures) Laboratory instructor	Fall 2012 – Spring 2015	Kent State University
Intro to Databases Grader	Spring 2014	Kent State University

Undergraduate Mentoring

- Tessandra Sage, Kent State University, Fall 2014.
- David Carlyn, Kent State University, Fall 2016 and Spring 2017
- Vlas Zyrianov, Kent State University, Fall 2016 and Spring 2017

Professional Activities

Conferences Attended

- International Conference on Software Maintenance (ICSM ‘11)
- Working Conference on Software Visualization (VISSOFT ‘11)
- International Conference on Software Maintenance and Evolution (ICSME ‘15)
- Mining Unstructured Documents (MUD ‘15)
- International Conference on Software Maintenance and Evolution (ICSME ‘16)
- International Conference on Software Analysis, Evolution, and Reengineering (SANER ‘17)

Additional Reviewer

- IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER’16)
- IEEE 31st International Conference on Software Maintenance & Evolution (ICSME’15) – *ERA Track*
- ACM/IEEE 37th International Conference on Software Engineering (ICSE’15)
- IEEE 23rd International Conference on Program Comprehension (ICPC’15)
- ACM 8th International Symposium on Software and Systems Traceability (SST’15)
- IEEE International Working Conference on Software Visualization (VISSOFT’15)
- ACM/IEEE 36th International Conference on Software Engineering (ICSE’14)
- IEEE International Working Conference on Software Visualization (VISSOFT’14)
- IEEE 30th International Conference on Software Maintenance & Evolution (ICSME’14) – *ERA Track*
- IEEE CSMR-WCRE 2014, the European Conference on Software Maintenance (CSMR’14)
- IEEE 21st International Working Conference on Reverse Engineering (WCRE’14)
- IEEE 29th International Conference on Software Maintenance (ICSM’13)
- IEEE 21st International Conference on Program Comprehension (ICPC’13)
- ACM International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE’13)
- IEEE International Working Conference on Software Visualization (VISSOFT’13)
- IEEE 20th Working Conference on Reverse Engineering (WCRE’13)
- IEEE Transactions on Software Engineering (TSE)
- Journal of Software: Evolution and Process (Formerly Journal of Software: Maintenance and Evolution) (JSEP)

Programming Languages

Primary languages include C++, C, C#, and Python. I have some previous experience with Javascript, Haskell, and Java.