Welcome to MUD 2015, the 5th Workshop on Mining Unstructured Data. The workshop is co-located with the 31st International Conference on Software Maintenance and Evolution (ICSME 2015) and is taking place in Bremen, Germany.

To analyze, comprehend, and reverse engineer software projects and their software development processes, we rely on various sources of information. Bug reports, execution logs, mailing lists, code review reports, change logs, requirements documents, and the actual source code contain implicit developer knowledge about the project and past development efforts. Most of this knowledge is captured as unstructured information: natural language text used to exchange information among people. Researchers in the Information Retrieval, Data Mining, and Natural Language Processing fields have experimented with various techniques and ad-hoc approaches to enable the mining of unstructured data. However, these techniques were not designed to work with the complexities and peculiarities of unstructured software engineering data. Mining unstructured software engineering data poses new and unique challenges, and addressing them is the goal of this workshop.

The 5th Workshop on Mining Unstructured Data (MUD 2015) aims to stimulate discussion between researchers in the field, to encourage cross-fertilization from different research domains, including Natural Language Processing, Information Retrieval, and Machine Learning, and to document and advance the state of the art of MUD in software engineering. More specifically, the intended goals of this workshop are to:

1) Facilitate knowledge exchange in the field of mining unstructured software engineering data through the presentation of short papers.

2) Offer a common framework for showcasing techniques, methodologies, and tools for mining unstructured data, thus enabling researchers and practitioners to find the tools that meet their particular mining needs.

3) Identify open problems and challenges for mining unstructured data in software engineering, thus providing the basis for a roadmap of future research opportunities in the field.

4) Share, discuss, and advance the state-of-the-art in mining unstructured data.

MUD 2015 is a full-day workshop and includes a keynote by Dr. Denys Poshyvanyk (College of William and Mary, USA), two hands-on tutorials by Dr. Massimiliano di Penta (University of Sannio, Italy) and Dr. Jonathan Maletic (Kent State University, USA), paper presentations, and group discussion.

Sonia Haiduc and Gabriele Bavota

MUD'15 Co-Chairs
Workshop Overview

Introduction from the organizers (S. Haiduc and G. Bavota)

Keynote: "Mining Unstructured Data for Mobile Software Development: Challenges and Opportunities", Denys Poshyvanyk

Tutorial One: "Put your hands in the mud: what technique, why, and how?", Massimiliano Di Penta

Paper presentations and group discussion

Tutorial Two: "Exploration, Analysis, and Manipulation of Source Code with srcML", Jonathan Maletic
Organizing Committee

Sonia Haiduc, Florida State University – USA
Gabriele Bavota, Free University of Bolzano-Bozen – Italy

Technical Program Committee

Bogdan Dit, Boise State University – USA
Emily Hill, Drew University – USA
Abram Hindle, University of Alberta – Canada
Nicholas Kraft, ABB Corporate Research – USA
David Lo, Singapore Management University – Singapore
Collin McMillan, University of Notre Dame - USA
Annibale Panichella, Delft University of Technology – The Netherlands
David Shepherd, ABB Corporate Research – USA

Steering Committee

Bram Adams, École Polytechnique de Montréal - Canada
Alberto Bacchelli, Delft University of Technology – The Netherlands
Nicolas Bettenburg, Queen’s University – Canada
Keynote: Mining Unstructured Data for Mobile Software Development: Challenges and Opportunities

Denys Poshyvanyk, College of William and Mary, USA

The mobile handset industry has been growing at an unprecedented rate. This global app economy that encompasses millions of apps and developers, supporting billions of devices and users, has been a tremendous success. Many of these mobile apps have features that rival their desktop counterparts and span several domain categories from games to medical apps. Yet, app developers and testers face new, emerging challenges such as rapid platform/library evolution resulting in API instability, platform fragmentation, continuous market pressure for frequent releases, and countless unstructured user reviews that need to be analyzed to find bugs and improve apps. Moreover, mobile platforms enable user interaction via touchscreens and sensors presenting new challenges for software testing.

In this talk I will overview recent work aimed at solving some of these challenges via mining of structured and unstructured software development data in mobile apps. First, I will show the results of a study conducted on 5,848 (free) Android apps aimed at analyzing how the ratings that apps received correlated with the fault- and change-proneness of the APIs upon which those apps relied. This study highlights many research opportunities that exist for building practical tools to support developers in dealing with API updates that can potentially (and inadvertently) impact their apps with breaking changes and bugs. To highlight one such opportunity, I will outline a solution (as well as the challenges) for tracing unstructured user reviews onto source code changes, and for monitoring the extent to which developers accommodate crowd requests and follow-up user reactions as reflected in their ratings. Second, I will present an approach for mining Android app usages for the purpose of generating actionable GUI-based execution scenarios that can be used in mobile app testing. The approach aims to mine models capable of generating feasible and fully replayable scenarios reflecting either natural user behavior or uncommon usages (e.g., corner cases) for a given app. Third, I will present an approach aimed at supporting the auto-completion of bug reports for mobile apps (with the purpose of making them more reproducible) that relies upon program artifacts extracted through static and dynamic analyses. Finally, I will highlight the opportunities, in terms of open technical problems and the potential benefits of solving these problems, for mining (un)structured data to support mobile app development, maintenance, and testing.

BIOGRAPHY: Denys Poshyvanyk is an Associate Professor in the Department of Computer Science at the College of William and Mary in Virginia, USA where he leads SEMERU research group. His research interests are in the area of software engineering, evolution, maintenance and program comprehension. His recent research projects span topics such as repository mining, traceability, mobile app (Android) development and testing, energy consumption, and reuse. His papers received several Best Paper Awards at ICPC’06, ICPC’07, ICSM’10, SCAM’10, ICSM’13 and ACM SIGSOFT Distinguished Paper Awards at ASE’13 and ICSE’15. He is a recipient of the NSF CAREER award (2013). Denys currently serves on the steering committee of the IEEE International Conference on Software Maintenance and Evolution (ICSME) and IEEE International Conference on Program Comprehension (ICPC). He also serves on the editorial board of the Empirical Software Engineering Journal (Springer). He served as a PC Co-Chair for ICPC’13, WCRE’12, and WCRE’11. He will serve as a PC Co-chair for IEEE ICSME’16. He received his Ph.D. degree in Computer Science from Wayne State University, USA in 2008. He also obtained his M.S. and M.A. degrees in Computer Science from the National University of Kyiv-Mohyla Academy, Ukraine and Wayne State University, USA in 2003 and 2006, respectively.
Tutorial One: Put your hands in the mud: what technique, why, and how?

Massimiliano Di Penta, University of Sannio, Italy

Mining unstructured data is a crucial and often underestimated challenge for software engineering researchers wanting to develop different kinds of recommender systems. To date, there are different kinds of approaches available, such as pattern matching, various kinds of information retrieval algebraic models, and, last but not least, natural language parsing. However, when applying such techniques to software artifacts, it is crucial to carefully ponder two aspects. The first is choosing the most suitable technique for a given problem and for certain kinds of artifacts, as very often one may be tempted to choose an overly-sophisticated technique that, while working well on some problems, may not work so well on others. The second is customizing approaches and tools that, as they have been conceived, they might not work properly on software-related artifacts.

This tutorial provides guidelines and hands-on instructions on pondering, selecting, customizing and applying various kinds of techniques to process software-related artifacts and solve problems relevant for software engineers.

BIOGRAPHY: Massimiliano Di Penta is to date associate professor at the University of Sannio, Italy. His research interests include software maintenance and evolution, mining software repositories, empirical software engineering, search-based software engineering, and testing. He is author of over 200 papers appeared in international journals, conferences and workshops. He serves and has served in the organizing and program committees of over 100 conferences such as ICSE, FSE, ASE, ICSM, ICPC, GECCO, MSR WCRE, and others. He is currently member of the steering committee of ICSME, MSR, SSBSE, and PROMISE. Previously, he has been steering committee member of other conferences, including ICPC, SCAM, and WCRE. He is in the editorial board of IEEE Transactions on Software Engineering, the Empirical Software Engineering Journal edited by Springer, and of the Journal of Software: Evolution and Processes edited by Wiley. More at http://www.ing.unisannio.it/mdipenta.
Tutorial Two: Exploration, Analysis, and Manipulation of Source Code with srcML
Jonathan Maletic, Kent State University, USA

This tutorial is intended for those interested in constructing custom software exploration, analysis, and manipulation tools to support research or commercial applications. srcML (srcML.org) is an infrastructure consisting of an XML representation for C/C++/C#/Java source code along with efficient parsing technology to convert source code to-and-from the srcML format. The tutorial describes srcML, the toolkit, and the application of XPath to query source code. Additionally, a hands-on tutorial of how to use srcML and XML tools to construct custom exploration tool for mining source code will be conducted.

BIOGRAPHY: Jonathan I. Maletic is Professor in the Department of Computer Science at Kent State University. He received the Ph.D. and M.S., both in Computer Science, from Wayne State University in 1995 and 1989 respectively. He received the B.S. in Computer Science in 1986 from The University of Michigan-Flint. His research interests are centered on software evolution, with a focus on the comprehension, analysis, manipulation, transformation, reverse engineering, traceability, and visualization of large-scale software systems. Prof. Maletic has authored over 110 refereed publications and is regularly funded by the US National Science Foundation. He has graduated 14 doctoral students, 12 of which currently hold academic positions.