Context in Software Development (CSD)

Kelly Blincoe (Main Contact)
University of Victoria
Victoria, BC, Canada
kblincoe@acm.org

Daniela Damian
University of Victoria
Victoria, BC, Canada
danielad@uvic.ca

Giuseppe Valetto
Fondazione Bruno Kessler
Trento, Italy
valetto@fbk.eu

Gail Murphy
University of British Columbia
Vancouver, BC, Canada
murphy@cs.ubc.ca

ABSTRACT

The goal of this one-day workshop is to bring together researchers interested in techniques and tools that leverage context information around development activities. There is little research on defining what context is, how we can model it, and how we can use those models in software development at large. This workshop aims at identifying and modeling context in software development, as well as discussing its application.

Categories and Subject Descriptors

D.2.6 [**Software Engineering**]: Programming Environments – *integrated environments, interactive environments*.

General Terms

Management, Human Factors, Theory.

Keywords

Context, Software Development.

1. WORKSHOP OVERVIEW

This workshop addresses the general topic of context around software development activities. In this section, we describe the theme, goals and topics of the workshop, as well as a motivation of its relevance and timeliness.

1.1 Theme, Goals and Topics

The goal of the workshop is to bring together researchers interested in techniques and tools that leverage *context information around development activities* (for brevity, referred to as simply *context* or *development context* henceforth) for better support of software development activities.

In large and complex software projects, developers must manage a large amount of context to complete their development tasks, including the relevant software artifacts and their change history, dependent tasks, concurrent work, discussions and knowledge exchanges about those tasks and artifacts, and more. In fact, context in software development is multifaceted, and what information is relevant as context for a developer working on a given task is not fully understood. Still, developers must make use of knowledge gleaned from all this context to make decisions, coordinate their work, understand the purpose behind their tasks, and understand how their tasks fit with the rest of the project.

Tools exist to help software developers manage and leverage the copious amount of context around their development tasks. For example, Mylyn (formerly Mylar) [7][8], transforms an individual developer's Integrated Development Environment (IDE) into a task-centric view to make context switching between tasks easier. Mylyn records all developer interactions within the IDE as they occur. These events are stored as context data for the task in focus, and represent a specific way to model development context However, that model captures only which software artifacts are related to a given development task, and thus represent a subset of all context information related to that task.

Developers can obtain contextual information from a variety of different sources. For example, Git Blame (http://git-scm.com/docs/git-blame) gives a context for who last changed a line in a file in the Git version control system. Additional context can be gleaned from discussions and knowledge exchanges found within task reports in issue trackers like Bugzilla (http://www.bugzilla.org) or Jira (https://www.atlassian.com/software/jira).

In general, traces of context can be extracted from many tools, and used for empirical studies to better understand software development. For example, the context data available from Mylyn has been used to better understand coordination needs of developers [2][3], optimize task scheduling [6], and provide context around developer microblogging activities [5]. However, that is a limited use of development context. We, as scholars, could benefit from having context recorded, modeled and applied more systematically for all types of software development activities.

There is little research on defining what context is, how we can model it, and how we can use those models in software development at large. This workshop aims at identifying and modeling context in software development, as well as discussing its application. Papers may address issues along the broad topics, including but not limited to the following:

- Types of context in software development
- Modeling context in software development
- Application of context in software development
- Using context to support implicit coordination
- Context in requirements engineering
- Context in testing and maintenance,
- Context in collaborative activities such as bug triaging
- Context in agile processes

1.2 Motivation

Identifying and modeling context in software development will lay the foundation for future techniques and tools that leverage development context for better support of software developers as they manage and make use of the copious amounts of context

Conference'10, Month 1–2, 2010, City, State, Country.

around their development tasks. In addition, more in-depth studies of context and models of context can help researchers develop a better understanding of many facets of software development, including cognitive, behavioral, social and teamwork issues.

2. WORKSHOP FORMAT

Context in Software Development will be a one-day workshop. We will invite two kinds of submissions: 5-page papers and 2-page position papers. A selection of papers will be presented at the workshop. Papers selected for presentation will be allotted 10 minutes plus 10 minutes of discussion. Other accepted papers will be presented via a poster session.

Accepted papers will be distributed to all participants at least two weeks prior to the workshop, and participants will be encouraged to review the papers ahead of time to identify topics of interest for discussion. Each presenter will be asked to pose three questions to the workshop participants on their final slide to stimulate discussion. Questions that receive the most interest from the workshop participants will be selected for further round-table discussions during the second part of the workshop. Further, the moderator for each session will be prepared with questions and discussion topics for the papers in their session. We expect to accept up to 10 papers and 10 notes.

Preliminary agenda:

9:00am	Start, Welcome and Icebreaker
9:30am	Keynote Speaker
10:00am	Paper Presentations (Session I)
11:00am	Coffee Break
11:30am	Paper Presentations (Session II)
1:00pm	Lunch break and Posters
2:30pm	Round Table Discussion
4:00pm	Coffee break
4:30pm	Round Table Discussion
5:30pm	Closing

3. PAPER SELECTION PROCEDURE

Each submission will be reviewed by at least two members of the Program Committee. Submissions will be evaluated on the basis of originality, importance of contribution, soundness, evaluation, quality of presentation and appropriate comparison to related work. The program committee as a whole will make final decisions about which submissions to accept for presentation at the workshop.

Submissions will be accepted electronically through EasyChair at https://www.easychair.org/conferences/?conf=csd14.

Papers are intended as positions for discussion and not for archival publication, so they will not appear in the ACM Digital Library.

Important Dates:

Call for papers	May 9, 2014
Paper submissions due	July 7, 2014
Notifications	August 1, 2014
Camera Ready deadline	August 15, 2014

Proposed Program Committee Members:

- Andre van der Hoek, University of California, Irvine
- Andrew Begel, Microsoft Research
- Walid Maalej, University of Hamburg
- Romain Robbes, University of Chile
- Martin Robillard, McGill University
- Annie Ying, McGill University
- Mik Kersten, TaskTop Technologies
- Cleidson de Souza, Federal University of Para and Vale Institute of Technology, Brazil
- Rashina Hoda, University of Auckland
- Philippe Kruchten, University of British Columbia
- Tore Dybå, University of Oslo and SINTEF, Trondheim, Norway
- Daniela S. Cruzes, NTNU, Trondheim, Norway
- Kate Ehrlich, IBM
- Patrick Wagstrom, IBM
- Jim Herbsleb, Carnegie Mellon University
- Laura Dabbish, Carnegie Mellon University
- Bruno Antunes, University of Coimbra
- Paulo Gomes, University of Coimbra
- John Grundy, Swinburne University of Technology

Our preliminary workshop website and call for papers can be accessed at http://csd-ws.github.io.

4. PUBLICITY PLANS

Based on the reception from the proposed Program Committee members and past participation in previous FSE workshops, we expect approximately 30 participants.

The workshop call for papers will be advertised through social media including Twitter and Facebook. Targeted emails will be sent to researchers who have recently published in this research area.

We will encourage all participants to submit at least a 2-page position paper, but the workshop will be open; all attendees will be asked to participate in discussions.

5. FOLLOW-UP PLANS

We are currently negotiating a post-workshop special issue of Journal of Software and Systems for extended versions of the best papers.

6. BIOGRAPHIES

Kelly Blincoe received a BE in Computer Engineering from Villanova University in 2004, an MS in Information Science from Pennsylvania State University in 2008, an MS in Computer Science from Drexel University in 2011, and a Ph.D in Computer Science from Drexel University in 2014. She is currently a Postdoctoral Fellow at the University of Victoria. She previously worked at Lockheed Martin as a Proposal Manager and Software Engineer. Her research interests lie in collaborative software engineering and computer-supported cooperative work.

Daniela Damian is an Associate Professor in University of Victoria's Department of Computer Science, where she leads research in the Software Engineering Global interAction Laboratory (SEGAL, thesegalgroup.org). Her research interests include Software Engineering, Requirements Engineering, Computer-Supported Cooperative Work and Empirical Software Engineering. Her recent work has studied the interplay of the social and technical aspects of developers' coordination in large, geographically distributed software projects. Daniela has served on the program committee boards of several software engineering conferences, and was (co-)organizer on many workshops at ICSE, FSE and RE conferences. The Global Software Engineering Workshop she organized at ICSE in early 2000 was very successful and became the current International Conference on Global Software engineering (ICGSE) – of which first edition in 2006 she served as the program co-chair. She is currently serving on the editorial boards of the Journals of Transactions on Software Engineering, Requirements Engineering, Empirical Software Engineering, and Software and Systems.

Giuseppe Valetto received a Laurea degree in Electronic Engineering from Politecnico di Torino, Turin, Italy in 1992, an MS in Computer Science from Columbia University, New York, NY, USA in 1994, and a Ph.D. in Computer Science, again from Columbia University, in 2004. In his career as a researcher, he has mostly worked on collaborative software engineering and distributed and self-adaptive systems. He has held positions as Xerox Research in Grenoble, France, CEFRIEL- Politecnico di Milano, Milan, Italy, Telecom Italia Lab, Turin, Italy, IBM T.J. Watson Research Center, Hawthorne, NY, USA, Drexel University, Philadelphia, PA, USA. He is currently a researcher in the Service Oriented Applications unit at Fondazione Bruno Kessler, Trento, Italy. He was the general chair of the IEEE Self-Adaptive and Self-Organizing Systems (SASO) Conference in 2013.

Gail C. Murphy is a Professor in the Department of Computer Science and Associate Dean (Research and Graduate Studies) in the Faculty of Science at the University of British Columbia. She is also a co-founder and currently Chief Scientist at Tasktop Technologies. Her research interests are in improving the productivity of knowledge workers, especially software developers. She has been a co-organizer of workshops at ICSE and OOPSLA. She has served as a Program Chair for FSE 2008, a Co-Program Chair for ICSE 2012, General Chair of AOSD 2004 and a co-local organizer of ICSE 2013.

7. REFERENCES

[1] Antunes, Bruno, Joel Cordeiro, and Paulo Gomes. "An Approach to Context Modeling in Software Development." In *Software and Data Technologies*, pp. 188-202. Springer Berlin Heidelberg, 2013.

[2] Blincoe, K, Valetto, G. and Damian, D., "Do all task dependencies require coordination? The role of task properties in identifying critical coordination needs in software projects," *Proc. 9th Joint Meeting on Foundations of Software Engineering* (ESEC/FSE 13), ACM, 2013, pp. 213-223.

[3] Blincoe, K. Valetto, G. and Goggins, S., "Proximity: a Measure to Quantify the Need for Developers Coordination," *Proc. Conf. Computer Supported Cooperative Work* (CSCW 12), ACM, 2012, pp.1351-1360.

[4] Dybå, Tore, Dag IK Sjøberg, and Daniela S. Cruzes. "What Works for Whom, Where, When, and Why?: On the Role of Context in Empirical Software Engineering." In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, pp. 19-28. ACM, 2012.

[5] Guzzi, A. Pinzger, M. and van Deursen, A., "Combining Micro-Blogging and IDE Interactions to Support Developers in their Quests," *Proc. Int'l Conf. Software Maintenance (ICSM 10),* IEEE CS, 2010, pp. 1–5.

[6] Kasi, B.K. and Sarma, A., "Cassandra: Proactive Conflict Minimization through Optimized Task Scheduling," *Proc. 35th Int'l Conf. Software Eng. (ICSE 13)*, IEEE Press, 2013, pp. 732-741.

[7] Kersten, M. and Murphy, G.C., "Mylar: a Degree-of-Interest Model for IDEs," *Proc. 4th Int'l Conf. Aspect-Oriented Software Development (AOSE 05)*, ACM, 2005, pp. 159-168.

[8] Kersten, M. and Murphy, G.C., "Using Task Context to Improve Programmer Productivity," *Proc. 14th Int'l Symposium Foundations of Software Engineering (FSE 06)*, ACM, 2006, pp. 1-11.