– **Workshop proposal for ESEC/FSE 2013** –

ARCS 2013: 1st International Workshop on Aligning Research on Code Smells

https://sites.google.com/site/warcs2013/

Leon Moonen,[*] Aiko Yamashita
Simula Research Laboratory
Lysaker, Norway
leon.moonen@computer.org, aiko@simula.no

Tracy Hall, Steve Counsell
Brunel University
Uxbridge, U.K.
{tracy.hall, steve.counsell}@brunel.ac.uk

ABSTRACT

Code smells are a metaphor for issues with source code that negatively affect software comprehensibility and maintainability. As such, code smells can help to monitor the health of a system throughout its evolution, and to guide efforts to sustain comprehensibility and maintainability. Much effort by researchers and tool vendors has focused on the detection of code smells. Recently, studies have started to investigate the impact of smells on software maintenance and evolution. However, there is a long road to go before code smell analysis can drive exactly those changes that improve comprehensibility and maintainability of a given system. We have identified four challenges that need to be addressed: (1) Lack of a common vocabulary to build a consistent knowledge base, (2) Lack of an ontological framework to compare/synthesize results across studies, (3) Lack of an evaluation framework to compare/evaluate the quality of detection approaches, and (4) Lack of an agenda to make the knowledge/tools accessible in industrial contexts. The ARCS workshop aims at establishing a shared roadmap to address these challenges and mature code smell research.

1. THEME, GOALS & TOPICS

Code smells are indicators of issues with source code that are associated with bad program design and bad programming practices and which negatively affect software comprehensibility and maintainability. An advantage of code smells over other more "traditional" software measures is that smells are associated with an explicit set of refactoring strategies to improve the existing design. As such, code smell analysis is a promising approach to address both the assessment and the improvement of comprehensibility and maintainability. Most code smell related research in the last decade has focused on the definition and detection of code smells, leading to a series of techniques and approaches that have found their way into various professional software development and assessment tools. Unfortunately few of these definitions and detection approaches are consistent with each other and a common understanding of what the definition of a bad smell is and how it should be detected is urgently needed. In recent years, there has also been an increase in empirical studies investigating the impact of code smells on software maintenance and evolution. We believe that the next step is to work towards a better alignment of the different work conducted on code smells.

The goal of this workshop is to lay the foundations for: (1) establishing an initial "domain vocabulary" for the code smell research community. (2) consolidating the current empirical evidence on the effects of code smells and building an ontological framework to compare empirical evidence, (3) establishing a framework for evaluating and comparing code smell detection approaches, and To this end, we bring together practitioners, researchers, academics, and students working in the area of code smells, to share experiences, concur on terminologies and evaluation guidelines, to ultimately build a common research agenda for the community.

Motivation: Despite of the progress achieved during the last decade, there is a long road to go before code smell analysis can cost-efficiently drive the changes that will improve the comprehensibility and maintainability of a given system. One of the big challenges is the need to better integrate the different research work conducted on code smells.

From the current body of knowledge in code smells, it is possible to see that code smells are not harmful to the same extent across different studies, indicating that their effects are potentially contingent on moderator variables and interaction effects. And despite the plethora of detection methods and analysis tools, very little is reported on how various code smell detection tools actually perform when conducting maintainability assessments, and how they compare to each other. Studies reporting tools seldom address the applicability of code smells in industrial, real-life contexts or validate essential aspects such as their descriptive richness or their capability to assess different maintainability factors.

The above-mentioned situations make the interpretation of code smells or the selection of smell detection tools very challenging for practitioners. We postulate that several challenges need to be tackled in order to align the current body of knowledge on code smells:

[*]Primary contact, phone +47 92662474. All organizers can be contacted via email to arcs2013-organizers@googlegroups.com

- Lack of a common vocabulary (conceptual framework) to build a consistent and robust knowledge base of the field.
- Lack of an ontological framework to compare and synthesize results of empirical studies on code smells.
- Lack of an evaluation framework to compare and evaluate the quality of various smell detection approaches.
- Lack of a clear agenda to make the knowledge/tools accessible to industrial contexts.

A common vocabulary – Since definitions of code smells are based on textual, informal descriptions, this could lead to multiple interpretations and sometimes even incompatible definitions and detection approaches. The lack of consensus on a "common vocabulary" hinders further efforts on building a more mature research community, as different researchers utilize different terminologies and constructs. This problem goes beyond a mere "conceptualization" issue, as interpretations of a given smell may influence to a great extent, the implementations for its detection. The great diversity of smell detection approaches represents a challenge to compare results from studies using different tools and detection approaches.

An ontological framework – The context of the studies within the current body of knowledge varies with respect to: the domain of the system and its size, the type of task and its size, the characteristics of the developers, and the code smell detection procedure. Thus, it is necessary to build an ontological framework that encompasses critical moderator factors, in order to synthesize our body of knowledge, to facilitate the interpretation of empirical results, and to understand why some studies appear to display contradictory evidence. Also, studies use rather different operationalization of quality constructs, which also makes comparability difficult. In many cases, surrogates are used whiles there is no enough evidence on the usefulness/meaningfulness of those surrogates. We believe that more consistent operationalization of relevant quality constructs (e.g., Maintainability) within a common framework to evaluate empirical studies would facilitate the comparability of results across studies on code smells.

An evaluation framework – The lack of conciseness and openness on the smell detection approaches used makes comparing across studies using different detection strategies and tools very difficult. Moreover, there are no clear guidelines on how to evaluate the quality or usefulness of the detection strategies and tools in realistic settings.

An agenda to reach-out the practitioners – We believe that more in-vivo studies are required, involving professionals, realistic maintenance tasks, and industry-relevant systems, to ease the transfer of results from a study to the software industry. Realistic study contexts, in spite of being more difficult to attain, are likely to enable higher confidence in the results and lead to more practical insights to both academia and industry. Last but not least, we believe more focus should be given to study replications, and comparative studies that can challenge, extend or improve current work on code smell detection and interpretation.

2. WORKSHOP FORMAT

The workshop is scheduled to take one day. The preferred date of this workshop is before the main ESEC/FSE conference, and not overlapping with IWPSE because because we share a considerable part of the community, i.e., the ideal workshop date would be August 18. In order to build a research agenda to tackle the previously mentioned challenges, the workshop will have a strong focus on discussion and interaction,

2.1 Workshop Activities

To address the first goal, we invite participants to provide input on how the different terminologies and definitions of code smells compare to each other, as well as their relationship to concepts such as anti-patterns and design principles. In order to gather the participant's viewpoints, the "card sorting" technique[2] will be used. Card sorting is a popular technique for conceptualization tasks and is used in HCI for designing information spaces. We will use it to map the collective knowledge about the domain and identify potential gaps and mismatches During the workshop, participants will be asked to fill in index cards with terminologies and concepts of interest according to the different topics addressed. These will be collected and during the "card sorting" session, participants will be asked to collectively organize the cards into piles such that cards representing similar concepts are in the same pile. The participants are allowed to indicate piles with looser levels of similarity, merge piles or move cards, and will be finally asked to come up with concepts or names to describe the piles. A picture will be taken for each pile, and used as material for the 'wrap-up' session. The contents of the piles will also be archived on the workshop website.

To address the second goal, the participants will discuss how the different detection approaches compare to each other, and how the different "vocabulary" or concepts used by different researchers affect the implementation choices. Starting points for discussion are the parts from Zhang et al., [4] literature review that focus on software design research and earlier work by Murphy-Hill [1] to build an evaluation framework that can facilitate the evaluation and comparison of smell detection tools.

To address the third goal, we invite participants to reflect upon the current state of art on empirical studies on code smells. Relevant questions include: what is needed beyond "more" empirical studies, and what characteristics should those studies have? How much do the smell-related concepts differ across studies'? What are the potential reasons studies appear to display contradictory results on the same smells? What is needed to enhance comparability and thus, more robust cross-study results? Starting points for the discussion include the literature review by Zhang et al., [4] and the theoretical background chapter in Yamashita's dissertation [3].

In order to drive the discussion on the second and third goals, the open "Fishbowl" technique will be used. This technique is a variant of a panel discussion, where the audience sits in concentric circles around a set of four or five chairs for the panelists in the innermost circle (the fishbowl). One of the panelist seats is initially kept empty. After the moderator introduces the topic, the panelists start discussing. The audience listens in on the discussion but does not participate. However any member of the audience can, at any time, join the fishbowl by taking the empty chair. When that occurs, one of the current panelists must leave the fishbowl so that there is again a free chair. This process repeats until the time runs out, after which the moderator summarizes the discussion.

Table 1: Preliminary Schedule for the ARCS Workshop

	Morning session:		Afternoon session:
9:00	Welcome and overview of plans	13:30	Working session "definitions and terminology" with card sorting (60 minutes)
9:15	Keynote speaker (TBC)	14:40	Working session "empirical evaluations" (45 mins)
10:00	Presentations and discussion 1 (45 mins)	15:15	Coffee break
10:45	Coffee break	15:30	Working session "detection approaches" (45 mins)
11:00	Presentations and discussion 2 (45 mins)	16:15	Short break
11:45	Presentations and discussion 3 (45 mins)	16:30	Wrap up: lessons learned, resolved and open issues, next steps. (60 minutes)
12:30	Lunch	17:30	Joint dinner/social session

2.2 Schedule

The workshop will start with a keynote address, lasting 45 minutes. Subsequently, we will have thee focused sessions that each contain 4 short presentations ("lightning talks") based on accepted position papers. The lightning talks will each consist of a 5 minutes presentation, plus 1-2 minutes for clarification questions. At the end of every session there will be a longer (20 minute) discussion to synthesize results and identify the "next steps" for the common topic. To minimize the overhead of changing speakers, all presentation will be given from one laptop. A preliminary overview of the workshop schedule is in Table 1.

To stimulate and drive discussion during these sessions, we plan to collect sheets with questions/comments from the presenters as well as the audience during the individual presentations. These sheets also serve as role a "paper trail" to keep track of all topics discussed and will be scanned and put on the workshop web-page.

Based on feedback gained from the morning sessions, we aim to have three working sessions in the afternoon to work towards our intended results: (1) an initial "sketch" of the domain vocabulary of the community, and a strategy to build an integrated terminology dictionary, (2) an overview of the achievements on empirical studies on code smells, and a research agenda to improve comparability across studies, and (3) an evaluation framework to compare and evaluate smell detection techniques. For the first point, we will use the card-sorting technique to drive the process; for the last two points, we plan to use the above-mentioned "fishbowl" technique. The first session will take 60 minutes, and the last two sessions will each take 45 minutes.

The workshop will close with a 'wrap-up' session where we will concretize on the most important points to build a research agenda for the code smell community. Finally, since we feel that an important part of a workshop is to bring the community together through an informal discussion space, we plan to invite all participants to a joint dinner or social event (at own expense) in the evening after the workshop.

All accepted papers will be published in the ESEC/FSE workshop proceedings and made available to registered participants so they can read them prior to the workshop. In addition, we have reached agreement that a few of the best papers from the workshop will be invited to submit extended versions to the Journal of Information and Software Technology (IST), which has accepted the workshop organizers' proposal to guest-edit a special issue on Bad Code Smells.

3. PAPER SELECTION PROCEDURE

Prospective participants are invited to submit a position paper of 2-4 pages describing work or ideas related to the workshop topic, or an abstract describing the theme/vision of the enlightening talk, and its relevance to the workshop. Each submitted paper will be reviewed by at least three members of the program committee. Key decision criteria for paper/talk acceptance include relevance to the workshop's topics, originality and suitability for triggering discussion. EasyChair will be used to manage the submission review process. The accepted position papers will be published in the ACM Digital Library as part of the ESEC/FSE 2013 Workshop Proceedings.

We specifically solicit papers that directly address the workshop's goals by:

- proposing a framework/ontology to compare/synthesize results from empirical studies,
- describing and discussing potential moderator factors affecting results in empirical studies,
- proposing new designs for empirical studies to support comparability across studies,
- proposing methods for evaluation/comparison of detection approaches,
- proposing an evaluative framework for detection tools,
- describing conceptual and technical differences between detection approaches,
- cross-comparing the terminology used in the current body of knowledge

Important dates for submission and notification (preliminary, can be synced with other ESEC/FSE workshops):

Submission deadline:	May 20, 2013
Author notifications:	June 17, 2013
Camera-ready copies:	July 1, 2013 2013

The program committee will consist of renowned experts in the field of code smells and the final list will be announced later. The list of confirmed members of the PC include: Francesca Arcelli, Yann-Gaël Guéhéneuc, Michele Lanza, Isela Macia Bertran, Mika Mäntylä, Naouel Moha, Emmerson Murphy-Hill, Steffen Olbrich, Raed Shatnawri, Nikolaos Tsantalis, Andy Zaidman, and the workshop organizers.

4. PUBLICITY PLANS

We will publicize the workshop via the known software engineering, software reverse engineering and software evolution emailing lists, Facebook groups and LinkedIn groups. In addition, the organizers will use their own networks to publicize the workshop and we will ask our program committee to do the same. The ARCS2013 organizers can be contacted via email: arcs2013-organizers@googlegroups.com.

We maintain a workshop website where we will collect material in preparation of the workshop, and archive the material produced and collected during the workshop for future reference. This website is available at:

https://sites.google.com/site/warcs2013/

In the interest of keeping enough time for discussion, we aim to accept a maximum of 12 position papers. We estimate that the ARCS workshop will attract approximately 30 participants based on the community overlap that we see with the ESEC/FSE and IWPSE communities (and possible some from SBSE). Participation is primarily based on authorship of accepted position papers and on invitation. We will open participation for other interested parties and select on basis of a short motivation letter. We are open to additional "walk-in" participation during the workshop itself, but initially aim at a more committed audience via the submission of position papers or invitation requests via motivation letters.

5. FOLLOW-UP PLANS

The materials produced and collected during the preparation and operation of the workshop will be archived on the workshops webpage. The organizers will synthesize the results in a workshop report that discusses the challenges brought forward, the progress made on the roadmap, and the open issues that still need to be addressed. This report will be available from the workshop website and will be submitted to a suitable venue, such as ACM SIGSOFT Software Engineering Notes.

These archives will act as a starting point for follow-up editions of the ARCS workshop that we intend to organize, as we do not expect that all challenges in aligning research on code smells can be addressed in a single workshop.

6. BIOGRAPHIES

Leon Moonen: is a senior research scientist at Simula Research Laboratory. His research is aimed at devising and improving techniques and tools for the exploration, assessment and evolution of large industrial software systems. This combines several subfields of software engineering, such as program comprehension, reverse engineering, program analysis, software visualization and empirical software engineering. Dr. Moonen has published over 80 scientific papers (2029 citations, h-index: 25) and recently received the 10-year Most Influential Paper award for his work on the first automated code smell detector and it's use in software quality assessments. He serves on steering- and program committees of international conferences on software maintenance (ICSM), reverse engineering (WCRE), program understanding (ICPC), and source code analysis (SCAM) and has (co-)organized various workshops in these areas. Dr. Moonen received his MSc (cum laude, Computer Science, 1996) and PhD (Computer Science, 2002) from the University of Amsterdam. He is a member of ACM and IEEE Computer Society.

Aiko Yamashita: received the BSc. degree from Costa Rica Institute of Technology in 2004 and the MSc. degree from University of Gothenburg in 2007. She recently finalized her doctoral degree at Simula Research Laboratory and The Department of Informatics, University of Oslo. She has worked five years as a software engineer and consultant in Costa Rica, USA, Sweden and Norway within diverse organizations. Her research interests include empirical software engineering, source code analysis, software quality, psychology of programming, HCI and agile methods.

Tracy Hall: is a Reader in Software Engineering at Brunel University. Previously she was Head of the Systems & Software Research Group at the University of Hertfordshire and Adjunct Professor of Industrial Systems at the University of Oslo. Dr Hall's expertise is in Empirical Software Engineering research. Over the last 15 years she has conducted many empirical software engineering studies with a variety of industrial collaborators. Dr Hall has reported on the efficacy of various software engineering techniques and is currently investigating the use of program slicing metrics in relation to fault proneness and maintainability. She has particular interest in the human aspects of software engineering and has successfully led previous projects investigating the motivations of software engineers. She has published nearly 100 international peer reviewed journal and conference papers. Dr Hall was the co-chair of PASE in 1996 and an organising committee member of PROMISE 2010.

Steve Counsell: is a Reader in the Department of Information Systems and Computing at Brunel University. He received his PhD from Birkbeck, University of London in 2002 and his research interests relate to empirical software engineering; in particular, refactoring, software metrics and the study of software evolution. He worked as an industrial developer before his PhD. He has served on the program committee of numerous conferences and workshops and is currently on the organising committee of IEEE SEFM 2013. He was co-chair of the AMOST workshop at ISSTA in 2008. As part of recent government-funded projects 2009-2012, he has been chair a number of one-day Workshops in London, Berlin and Malmo. He has been a PC member of WETSOM since its inception.

7. EXTRA

The organisers would like to use the following equipment:

- data projector
- overhead projector, blank transparency sheets and sheet markers (multiple colors)
- one or two flip charts and markers (multiple colors)

As mentioned, the preferred workshop date is before the main ESEC/FSE conference, and not overlapping with IWPSE because because we share a considerable part of the community, i.e., the ideal workshop date would be August 18.

8. REFERENCES

[1] E. Murphy-Hill. Scalable, expressive, and context-sensitive code smell display. In *Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications*, OOPSLA Companion '08, pages 771–772. ACM, 2008.

[2] J. Nielsen. *Usability Engineering*. Academic Press, 1994.

[3] A. Yamashita. *Assessing the Capability of Code Smells to Support Software Maintainability Assessments : Empirical Inquiry and Methodological Approach*. PhD thesis, University of Oslo, 2012.

[4] M. Zhang, T. Hall, and N. Baddoo. Code Bad Smells: a review of current knowledge. *J. Softw. Maint. and Evolution: Research and Practice*, 23(3):179–202, 2011.