End User Software Engineering: CHI'2009 Special Interest Group Meeting

Brad A. Myers
Human Computer Interaction Inst.
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
bam@cs.cmu.edu
http://www.cs.cmu.edu/~bam
see also: http://eusesconsortium.org/

Margaret M. Burnett
School of Elec. Engr. &
Computer Science
Oregon State University
Corvallis, OR 97331
burnett@eecs.oregonstate.edu
http://web.engr.oregonstate.edu/~burnett/

Susan Wiedenbeck
College of Info Science & Technology
Drexel University
Philadelphia, PA 19104
http://www.ischool.drexel.edu/ faculty/wiedenbeck/

Andrew J. Ko
The Information School
University of Washington
Seattle, WA 98195
ajko@u.washington.edu
http://faculty.washington.edu/ajko

Mary Beth Rosson
Info. Sciences & Tech.
Pennsylvania State Univ.
University Park, PA 16802
814-863-2478
mrosson@ist.psu.edu

CHI 2009, April 4 – 9, 2009, Boston, MA, USA
ACM 978-1-60558-247-4/09/04.

Abstract
End users create software whenever they write, for instance, educational simulations, spreadsheets, or dynamic e-business web applications. Researchers are working to bring the benefits of rigorous software engineering methodologies to these end users to try to make their software more reliable. Unfortunately, errors are pervasive in end-user software, and the resulting impact is sometimes enormous. This special interest group meeting will bring together the community of researchers who are addressing this topic with the companies that are creating and using end-user programming tools.

Keywords
End-User Software Engineering (EUSE), End-User Development (EUD), End Users Shaping Effective Software (EUSES), Empirical Studies of Programmers (ESP), Psychology of Programming, Natural Programming

ACM Classification Keywords
D.2.5 Testing and Debugging; H.1.2 User/Machine Systems—Software psychology

Introduction
One way to define "programming" is as the process of transforming a mental plan of desired actions for

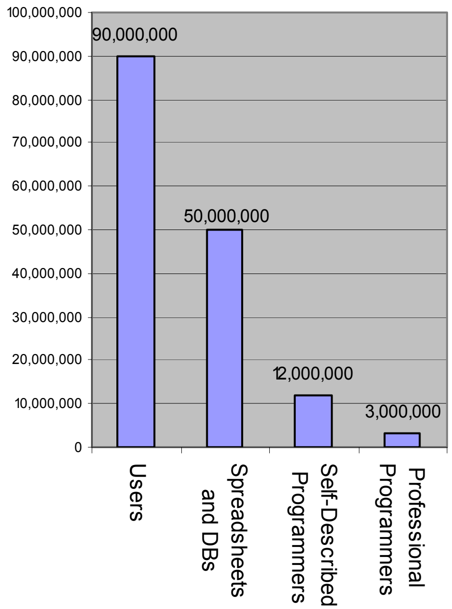

Figure 1: Estimates for the number of people in the US in 2006 who use computers at work, who use spreadsheets at work, who describe themselves as programmers, and who say they are professional programmers [24].

a computer into a representation that can be understood by the computer [10]. Expressed this way, it seems obvious that the study of humans and programming should be a topic of HCI. Indeed, this area of study has a long history, and has appeared under many names, including "Software Psychology" [25], "Psychology of Programming" [7, 9] and "Empirical Studies of Programming" (ESP).

We define "end-user programmers" (EUP) as people who write programs, but *not* as their primary job function — they write programs in support of achieving their main goal, which is something else, such as accounting, designing a web page, doing office work, scientific research, entertainment, etc. End-user programmers generally use special-purpose languages such as spreadsheet languages or web authoring scripts, but some EUPs, such as chemists or other scientists, may learn to use "regular" programming languages such as C or Java to achieve their programming goals.

Two NSF workshops determined that end-user software is in need of serious attention [4]. The reasons are compelling. Our research shows that while there are about 3 million professional programmers in the United States, over 12 million people say they do programming at work, and over 12 million people use spreadsheets and databases, and thus may also be considered to be doing programming [24] (see Figure 1). The NSF reports that there are about 6 million scientists and engineers in the US, most of whom program as part of their jobs [19]. Unfortunately, however, errors are pervasive in software created by end users. When the software that end users create is not dependable, there can be serious consequences for the people whose retirement funds, credit histories, e-business revenues, and even health and safety rely on decisions made based on that software. For example, a Texas oil firm lost millions of dollars in an acquisition deal through spreadsheet errors [22].

Two recent large collaborative efforts, one in the U. S. (the EUSES Consortium http://eusesconsortium.org/), and one in Europe (the Network of Excellence on End-User Development, http://giove.cnuce.cnr.it/eud-net.htm) have produced a number of promising results in this area (see, e.g., [27]). Special Interest Group meetings at CHI'2004 [15], CHI'2005 [16], and CHI'2007 [18], CHI'2008 [17] and the WEUSE series of workshops at ICSE'2005 [8], CHI'2006 [6], Dagstuhl 2007 (see http://www.dagstuhl.de/07081) and ICSE'2008 [1], very successfully brought together researchers and companies interested in this topic.

The special interest group (SIG) meeting at CHI'09 is designed to bring this community back together, as well as to introduce the area to others who are interested in allowing users to create more correct software.

Examples of Current Work

A few End-User Software Engineering (EUSE) projects, some of which have been presented at CHI, are already successful. Here are just a few examples.

The "Natural Programming" project at Carnegie Mellon University has been working for more than 10 years to make programming more "natural", or closer to the way people think. Many studies were performed (e.g., [12, 14, 20]), and new programming languages [21] and environments were created. For example, Figure 2 shows a new "Why"-oriented technique for debugging

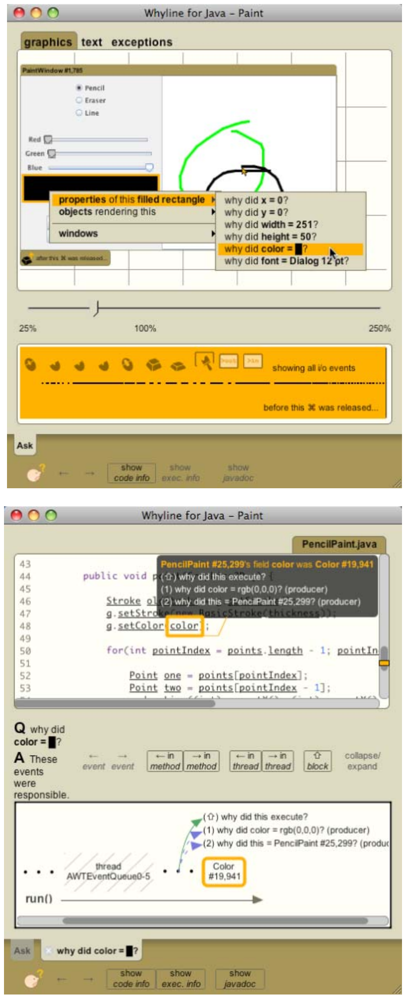

Figure 2: The Whyline for Java [11], showing a user asking about a black rectangle that was supposed to be blue (top), and the answer visualization (bottom).

which lets users ask about program output [11]. New work focuses on how to enable interaction designers to be more creative when exploring interactive behaviors {Myers, 2008 #1587}.

The "End-User Software Engineering" project involving researchers at Oregon State University and University of Washington aims to improve the reliability of software produced by end-user programmers. Some results have included "What You See Is What You Test" (WYSIWYT) integrated with fault localization [5], semi-automated detection of erroneous combinations of units in spreadsheets (Figure 3) [2], and new methods for involving end users in the "debugging" of machine-learned programs (Figure 4) [13]. The work emphasizes research on how to engage users in end-user software engineering practices without detrimentally interrupting their problem-solving efforts.

The Gender HCI Project [3], a collaboration of Oregon State University and Drexel University, has the goal to support both males' and females' problem solving, especially in end-user software development tasks. Our results show that females are less willing than males to try out and adopt software features that support testing and dataflow-oriented debugging, and further that male and female end-user programmers use different strategies when debugging [26]. Current work focuses on support for males and females in designing and reusing applications. Related work at Penn State has been exploring the impacts of up-front design planning in end-user programming, using simple representations like concept maps [23].

Acknowledgements
The authors are supported in part by the National Science Foundation as part of the EUSES Consortium under NSF grants ITR CCR-0324770, CCR-0325273, and CCR-0324844. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

References
[1] Abraham, R., Burnett, M., and Shaw, M. "Fourth Workshop on End-User Software Engineering (WEUSE IV)," in *In conjunction with ICSE 2008.* May 12, 2008. Leipzig, Germany: http://eusesconsortium.org/weuse/.

[2] Abraham, R. and Erwig, M. "Header and unit inference for spreadsheets through spatial analyses," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).* Sept. 27-30, 2004. pp. 165-172.

[3] Beckwith, L., Burnett, M., Grigoreanu, V., and Wiedenbeck, S., "Gender HCI: What about the software." *Computer*, Nov, 2006. **39**(11): pp. 97-101.

[4] Boehm, B. and Basili, V., "Gaining intellectual control of software development." *Computer*, May, 2000. **33**(5): pp. 27-33.

[5] Burnett, M., Cook, C., and Rothermel, G., "End-User Software Engineering." *CACM*, Sept, 2004. **47**(9): pp. 53-58.

[6] Burnett, M.M., Myers, B., Rosson, M.B., and Wiedenbeck, S. "The Next Step: From End-User Programming to End-User Software Engineering (WEUSE II)," in *Extended Abstracts, CHI'2006.* April 22-27, 2006. Montreal, Canada: pp. 1699-1702.

[7] Curtis, B., "Fifteen Years of Psychology in Software Engineering: Individual Differences and Cognitive Science," in *Proceedings of the 7th International Conference on Software Engineering*, 1984, IEEE Computer Society Press. pp. 97-106.

[8] Elbaum, S. and Rothermel, G., *First Workshop on End-User Software Engineering: WEUSE 2005.* In conjunction with ICSE 2005, May 21, 2005. http://www.cse.unl.edu/~grother/weuse/weuse-proceedings.pdf.

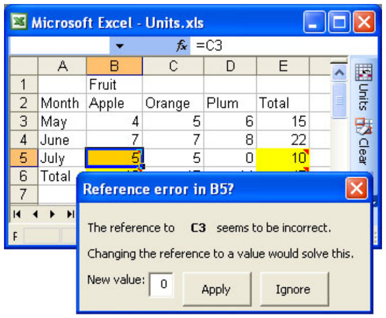

Figure 3: Microsoft Excel spreadsheet augmented by the Ucheck system that tries to help the user find errors [2].

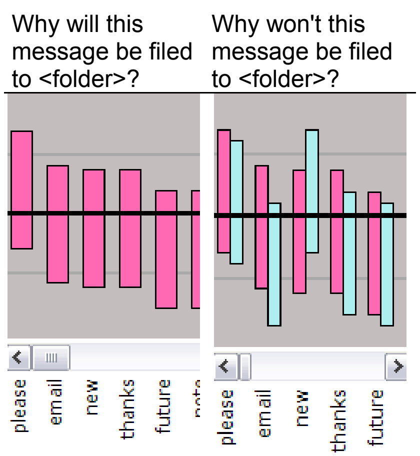

Figure 4: WhyLine approach for debugging machine-learned programs [13].

[9] Hoc, J.-M., Green, T.R.G., Samurçay, R., and Gilmore, D.J., eds. *Psychology of Programming*. 1990, Academic Press: London.

[10] Hoc, J.-M. and Nguyen-Xuan, A., "Language Semantics, Mental Models and Analogy," in *Psychology of Programming*, J.-M. Hoc*, et al.*, Editors. 1990, Academic Press. London. pp. 139-156.

[11] Ko, A.J. and Myers, B.A. "Debugging, Reinvented: Asking and Answering Why and Why Not Questions about Program Behavior," in *ICSE'2008: 30th International Conference on Software Engineering.* 10 - 18 May, 2008. Leipzig, Germany: pp. 301-310.

[12] Ko, A.J., Myers, B.A., Coblenz, M., and Aung, H.H., "An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks." *IEEE Transactions on Software Engineering*, Dec, 2006. **33**(12): pp. 971-987.

[13] Kulesza, T., Wong, W.K., Stumpf, S., Perona, S., White, R., Burnett, M., Oberst, I., and Ko, A.J. "Fixing the Program My Computer Learned:Barriers for End Users, Challenges for the Machine," in *IUI.* 2009. pp. to appear.

[14] LaToza, T.D., Garlan, D., Herbsleb, J.D., and Myers, B.A. "Program comprehension as fact finding," in *ESEC/FSE 2007: ACM SIGSOFT Symposium on the Foundations of Software Engineering.* September 3-7, 2007. Dubrovnik, Croatia: pp. 361-370.

[15] Myers, B.A. and Burnett, M. "End-Users Creating Effective Software (Special Interest Group Meeting Abstract)," in *Extended Abstract CHI'2004: Human Factors in Computing Systems.* April 24-29, 2004. Vienna, Austria: pp. 1592-1593.

[16] Myers, B.A., Burnett, M., and Rosson, M.B. "End Users Creating Effective Software. (Special Interest Group)," in *Extended Abstracts CHI'2005: Human Factors in Computing Systems.* April 2-7, 2005. Portland, OR: pp. 2047-2048.

[17] Myers, B.A., Burnett, M.M., Rosson, M.B., Ko, A.J., and Blackwell, A. "End User Software Engineering: CHI'2008 Special Interest Group Meeting," in *Extended Abstracts, CHI'2008.* April 5-10, 2008. Florence, Italy: pp. 2371-2374.

[18] Myers, B.A., Burnett, M.M., Wiedenbeck, S., and Ko, A.J. "End User Software Engineering: CHI'2007 Special Interest Group Meeting," in *Extended Abstracts CHI'2007.* April 28 - May 3, 2007. San Jose, CA: pp. 2125-2128.

[19] National Science Board, *Science and Engineering Indicators 2006.* National Science Foundation, volume 1: NSB 06-01; volume 2: NSB 06-01A, 2006. Arlington, VA. http://www.nsf.gov/statistics/seind06/.

[20] Pane, J.F. and Myers, B.A. "Tabular and Textual Methods for Selecting Objects from a Group," in *Proceedings of VL 2000: IEEE International Symposium on Visual Languages.* September 10-13, 2000. Seattle, WA: IEEE Computer Society. pp. 157-164.

[21] Pane, J.F. and Myers, B.A. "The Impact of Human-Centered Features on the Usability of a Programming System for Children," in *CHI.* Apr 1-6, 2002. Minneapolis, MN: pp. 684-685. http://www.cs.cmu.edu/~pane/research.html. Extended Abstracts for CHI'2002.

[22] Panko, R., "Finding spreadsheet errors: Most spreadsheet models have design flaws that may lead to long-term miscalculation." *Information Week*, May, 1995. pp. 100.

[23] Rosson, M.B., Sinha, H., Bhattacharya, M., and Zhao, D., "Design planning by end-user web developers." *Journal of Visual Languages and Computing*, 2008. **19**: pp. 468-484.

[24] Scaffidi, C., Shaw, M., and Myers, B. "Estimating the Numbers of End Users and End User Programmers," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05).* 20-24 September, 2005. Dallas, Texas: pp. 207-214.

[25] Shneiderman, B., *Software Psychology: Human Factors in Computer and Information Systems.* 1980, Cambridge, MA: Winthrop Publishers.

[26] Subrahmaniyan, N., Beckwith, L., Grigoreanu, V., Narayanan, V., Wiedenbeck, S., Burnett, M., Bucht, K., Drummond, R., and Fern, X. "Testing vs. Code Inspection vs. What Else? Male and Female End Users' Debugging Strategies," in *CHI'2008.* Apr, 2008. Florence, Italy: pp. 617-626.

[27] Wulf, V., Paterno, F., and Lieberman, H., eds. *End User Development*. 2006, Kluwer Academic Publishers: