End User Software Engineering: CHI'2007 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
Human Computer Interaction Inst.
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
ajko@cs.cmu.edu
http://www.cs.cmu.edu/~ajko

CHI 2007, April 28 – May 3, 2007, San Jose, USA
ACM 1-xxxxxxxxxxxxxxxxxx.

Abstract

Recently, researchers have been working to bring the benefits of rigorous software engineering methodologies to end users who find themselves in programming situations, to try to make their software more reliable. End users create software whenever they write, for instance, educational simulations, spreadsheets, or dynamic e-business web applications. Unfortunately, errors are pervasive in end-user software, and the resulting impact is sometimes enormous. This special interest group meeting has three purposes: to bring the results of a recent (February 2007) week-long "Dagstuhl" meeting on end-user software engineering to interested researchers at CHI; to incorporate attendees' ideas and feedback into an emerging survey of the state of this interesting new subarea; and generally to bring together the community of researchers who are addressing this topic, with the companies that are creating end-user programming tools.

Keywords

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

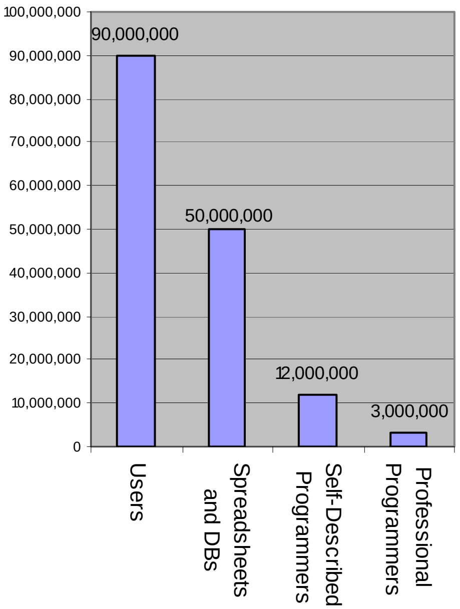


Figure 1: Estimates for the number of people in the US in 2006 who use computers at work, who use spreadsheets at work, who describe themselves as programmers, and who say they are professional programmers [20].

ACM Classification Keywords

Categories & Subject Descriptors: 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 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 gone under many names, including “Software Psychology” [21], “Psychology of Programming” [7, 9] and the “Empirical Studies of Programming” (ESP).

We define “end-user programmers” (EUP) as people who write programs, but *not* as their primary job function. Instead, they must 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 need to 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 [3]. 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 [20] (see Figure 1). 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 [19].

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, which resulted in a new book [22]) have produced a number of promising results in this area. Special Interest Group meetings at CHI’2004 [15] and CHI’2005 [16], and workshops at ICSE’05 [8] and CHI’06 [5] very successfully brought together researchers and companies interested in this topic. In addition, a recent Dagstuhl meeting was held (February 2007), at which about 50 researchers in the areas of HCI, programming languages, and software engineering spent a week studying the state of the art in this area (www.dagstuhl.de/07081).

The special interest group (SIG) meeting at CHI’07 is designed to bring this community back together, and to introduce the area to others who are interested in allowing users to create more correct software. The meeting will present the results of the Dagstuhl meeting to this community, and will solicit attendees’ inputs and feedback on an emerging survey that aims to capture the current state of this active new subarea. We especially want to involve practitioners interested in

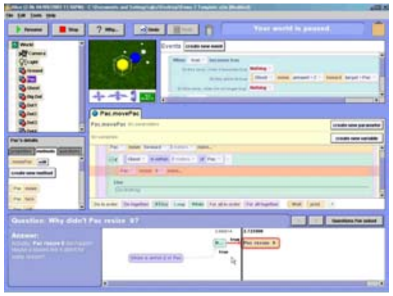
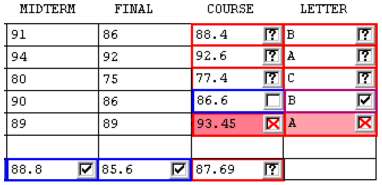


Figure 2: The Whyline [11] after a user asks why an event did not happen, and the resulting answer and visualization.



MIDTERM	FINAL	COURSE	LETTER
91	86	88.4 ?	B ?
94	92	92.6 ?	A ?
80	75	77.4 ?	C ?
90	86	86.6 □	B ☑
89	89	93.45 ☒	A ☒
88.8 ☑	85.6 ☑	87.69 ?	

Figure 3: WYSIWYT supports systematic testing for end users, to help the user test and debug the formulas and values [4].

current and future techniques that can be embodied in tools and development processes.

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., [13, 14, 17]), and new programming languages [18] and environments were created. For example, Figure 2 shows a new technique for debugging called the Whyline [12].

The “End-User Software Engineering” project at Oregon State University aims to improve the reliability of software produced by end-user programmers in general, and by spreadsheet users in particular. Some results have included “What You See Is What You Test” (WYSIWYT) integrated with fault localization (Figure 3) [4], and semi-automated detection of erroneous combinations of units in spreadsheets (Figure 4) [1]. The work emphasizes research on how to interest users in end-user software engineering devices without detrimentally interrupting their problem-solving efforts.

The Gender HCI Project [2], 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 have lower confidence than males in their ability to carry out spreadsheet testing and debugging tasks and are slow to try out and adopt software features that support these tasks. Behaviorally, males are drawn to tinkering, or playful experimentation, more than females, but they have a tendency to tinker to excess which can interfere with effectiveness. Pair collaboration in end-user software development tasks may be a way to improve effectiveness in these tasks [6].

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

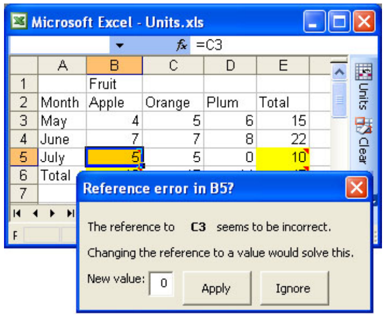
[1] Abraham, R. and Erwig, M. “Header and unit inference for spreadsheets through spatial analyses,” in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. Sept. 27-30, 2004. pp. 165-172.

[2] Beckwith, L., *et al.*, “Gender HCI: What about the software.” *Computer*, Nov, 2006. **39**(11): pp. 97-101.

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

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

[5] Burnett, M.M., *et al.* “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.



=C3

	A	B	C	D	E
1		Fruit			
2	Month	Apple	Orange	Plum	Total
3	May	4	5	6	15
4	June	7	7	8	22
5	July	5	5	0	10
6	Total				
7					

The reference to C3 seems to be incorrect.

Changing the reference to a value would solve this.

New value: 0

Apply Ignore

Figure 4: Microsoft Excel spreadsheet augmented by the Ucheck system that tries to help the user find errors [1].

[6] Chintakovid, T., *et al.* "Pair collaboration in end-user debugging," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).* Sept. 4-8, 2006. pp. 3-10.

[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.

[9] Hoc, J.-M., *et al.*, 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. "Designing the Whyline, A Debugging Interface for Asking Why and Why Not questions about Runtime Failures," in *CHI.* 2004. pp. 151-158.

[12] Ko, A.J. and Myers, B.A. "Designing the Whyline, A Debugging Interface for Asking Why and Why Not questions about Runtime Failures," in *Proceedings CHI'2004: Human Factors in Computing Systems.* April 24-29, 2004. Vienna, Austria: pp. 151-158.

[13] Ko, A.J., Myers, B.A., and Aung, H.H. "Six Learning Barriers in End-User Programming Systems," in *IEEE VL/HCC.* Sep 26-29, 2004. pp. 199-206.

[14] Ko, A.J., *et al.*, "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.

[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] 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.

[18] 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-2.cs.cmu.edu/~pane/research.html. Extended Abstracts for CHI'2002.

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

[20] 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.

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

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

Supplementary Material

Community(ies) to which this SIG would be of interest and why

We hope this SIG will be of interest to all the researchers trying to make programming easier, through various techniques such as Visual Programming, Programming by Example, and new programming environments. We also hope to attract representatives of the businesses that are using end-user programming techniques, such as Microsoft with Excel and Visual Basic Express, Adobe (formerly MacroMedia) with Director and Flash, MathWorks with MATLAB, Google with their new spreadsheet system, and IBM with their existing and emerging end-user programming tools. . The CHI'2004 and CHI'2005 SIGs on this theme attracted quite a large crowd – for each, 40-50 people filled out our sign-in sheet, including not only academics but also representatives from more than 20 different companies. We expect an even larger turnout this time, with the increased attention to this topic (such as the Invited Research Presentation last year by Brad Myers on *End-User Development*).

Assumed attendee background

We hope to get a great mix of people – from established researchers to students working in this area. From companies, we are hoping for a variety of people as well, anywhere from the developers at the companies who have to create these tools, to the HCI specialists, trainers, and technical writers who work on making them better and more understandable.

An informal schedule of discussion topics

We propose to use the 90 minutes as follows. First, we will start with about 40 minutes split among the organizers summarizing the state of the research in this area that we know about, results of the Dagstuhl workshop, and the emerging survey of end-user software engineering. Then, we will have about 50 minutes of discussion and presentations from the other attendees about other research in the area, issues and problems that should be addressed in general and/or in the survey, and opportunities and techniques for applying these ideas to real problems and applications. If the discussions lag, then we can provide more demonstrations and information about the projects that are being developed by the researchers, but based on previous meetings, we don't expect this to be a problem.

Which of the organizers should serve as the primary contact

Brad Myers, bam@cs.cmu.edu