
The Next Challenge: from Easy-to-Use to Easy-to-Develop. Are You Ready?

Joerg Beringer

VP of NextGen User Experience
SAP Labs, LLC
3410 Hillview Avenue
Palo Alto, CA 94304 USA
joerg.beringer@sap.com

Gerhard Fischer

Center for LifeLong Learning
and Design.
University of Colorado, Boulder,
gerhard@colorado.edu

Piero Mussio

Dip. Di Informatica e
Comunicazione.
Università degli Studi di Milano.
Via Comelico 39
20135 Milano, Italy
mussio@dico.unimi.it

Brad Myers

Human Computer Interaction Inst.
Carnegie Mellon University
Pittsburgh, PA 15213-3891
bam@cs.cmu.edu
<http://www.cs.cmu.edu/~bam>

Fabio Paternò

ISTI-CNR
Via Moruzzi 1
Pisa, 56124 Italy
fabio.paterno@isti.cnr.it
<http://giove.isti.cnr.it/~fabio>

Boris de Ruyter

Philips Research Europe
High Tech Campus 34 -WB-5.27
5656 AE Eindhoven, The
Netherlands
boris.de.ruyter@philips.com

Abstract

The main challenge of next years is to allow users of software systems, who are non-professional software developers, to create, modify or extend software artefacts. In this panel we want to discuss with the CHI community the key aspects in the area of End User Development and an associated research agenda, which should be then proposed to the main research agencies, such as NSF and EU ICT.

Keywords

End User Development, End-User Software Engineering, Psychology of Programming, Programming by Example, Macros, Tailoring, Customization

ACM Classification Keywords

H.1.2 User/Machine Systems—Software psychology
H.5.3. Information interfaces and presentation (HCI).

Introduction

We think that over the next few years, the goal of human-computer interaction will evolve from just making systems easy to use (even though that goal has not yet been completely achieved) to making systems that are easy to develop. By now, most people have become familiar with the basic functionality and interfaces of computers. However, developing new or modified applications that effectively support users' goals still requires considerable expertise in programming that

Copyright is held by the author/owner(s).

CHI 2008, April 5 - April 10, 2008, Florence, Italy

ACM 978-1-60558-012-8/08/04.

cannot be expected from most people. Thus, one fundamental challenge for the coming years is to develop environments that allow users who do not have background in programming to develop or modify their own applications. Today, some forms of EUD have found widespread use in commercial software with some success: recording macros in word processors, setting up spreadsheets for calculations and defining e-mail-filters. While these applications only realize a fraction of EUD's potential and still suffer from several flaws, they illustrate why EUD is an important contribution to letting users become active citizens of the Information Society.

End User Development concerns activities that imply some modification, aiming at creating from scratch or modifying an existing software artefact. Examples of these approaches are: Programming by Example (also called Programming by Demonstration), Visual Programming, Macros, and Scripting Languages. Research projects that have started to address such complex issues have been activated both in Europe (for example EUD-Net, <http://giove.isti.cnr.it/eud-net.htm> and EUDISMES, <http://www.eudismes.de>) and USA (for example EUSES, <http://eusesconsortium.org/>). This panel aims to bring together researchers (some from industrial organizations) from both sides of the Atlantic, with various experience in the area, which should stimulate the discussion with the CHI audience. The goal is to identify the key aspects in the area and an associated research agenda, which should then be proposed to the main research agencies, such as NSF and EU ICT. Some key issues for the discussion are the identification of the more relevant application domains and promising research approach, how to exploit current technological trends (e.g., the availability in the mass market of various types of interaction platforms,

such as mobile devices, digital TVs, ...), what new business models will be enabled by EUD.

Position Statements

Joerg Beringer

Joerg Beringer's article "Reducing Expertise Tension" [2] defines End User Development (EUD) as the design problem to enable subject matter experts to create or modify executable software components. This design problem is not limited to end users who want to adapt software to reflect their own work practices but to process analysts or power users in organizations, who want to configure content and processes for end user communities without having deep IT knowledge. EUD is not limited to end users, but to the design challenge per se to enable user to become creative on a level which typically is beyond their level of expertise.

Gerhard Fischer

The industrial information economy [1] has been focused on creating finished goods such as complete software systems, movies, and information repositories. The emerging networked information economy is democratizing the design and evolution of rich collaboratively constructed information environments by creating socio-technical environments supporting active contributors instead of passive consumers [5]. These fundamental changes create new challenges for interactive systems by breaking down the barriers and distinctions between designers and users and open up opportunities for social creativity. Meta-design [6] is focused on "design for designers". It creates open systems at design time that can be modified by their users and evolved, requiring and supporting more complex interactions at use time. Open systems allow significant modifications when the need arises. The

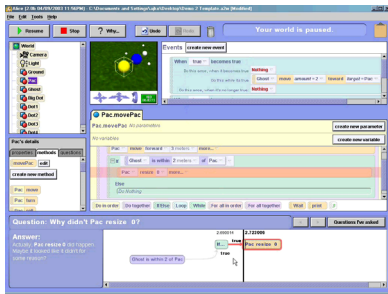


Figure 1: The Whyline [8] after a user asks why an event did not happen, and the resulting answer and visualization.

successes of open source software systems and open content environments have demonstrated that given the right conditions, design through the collaboration of many can create new kinds of systems. Meta-design tries to reduce the gap in the world of computing between a population of elite scribes who can act as designers and a much larger population of intellectually disenfranchised knowledge workers who are forced into a consumer role. To achieve these objectives, it addresses the following three necessities for socio-technical environments: (1) they must be flexible and evolve because they cannot be completely designed prior to use; (2) they must evolve at the hands of the users; and (3) they must be designed for evolution.

Piero Mussio

In a recent paper [7], Piero Mussio and co-authors present the current status of their research on End User Development, which focuses on people who are experts in a specific discipline (e.g. mechanical engineering, medicine, geology, etc.) but by no means experts in computer science, nor willing to be so, and who use computer systems for their daily work activities and are responsible for possible errors and mistakes, even those generated by wrong or inappropriate use of computer systems. Such domain experts usually lack any development skill and do not want to be constrained by formalisms that are alien to their culture. They want software environments to be easily accessible, that they can develop as environments shaped to their needs, tasks and habits without being aware of programming. To this aim, the participatory, meta-design Software Shaping Workshop methodology has been developed, which provides end users with software environments (called workshops in analogy to artisans' workshops) allowing them to perform development activities even at

use time. These environments provide interaction languages that reflect the end user mental models of the situations to be managed. Using these languages, end users become designers of their tools, and environments, being able of solving their problems and shape their software tools following their style of reasoning without being forced to think as programmers.

Brad Myers

Brad Myers's "Natural Programming" project [9] at Carnegie Mellon University has been working for more than 10 years to make programming more "natural", or closer to the way people think. The strategy is to first perform studies that reveal people's needs and strategies. This motivates novel tools, which are then evaluated to verify their impact. For example, studies revealed that debugging usually starts with "why" and "why not" questions [7], which motivated the design of the Whyline tool (see Figure 1) which studies showed decreased debugging time by a factor of 8 [8].

Fabio Paternò

People should be able to work through familiar and immediately understandable representations that allow them to easily express and manipulate relevant concepts, and thereby create or modify interactive software artefacts. In contrast, since a software artefact needs to be precisely specified in order to be implemented, there will still be the need for environments supporting transformations from intuitive representations into more precise, but more difficult to develop, descriptions. The main motivation for model-based approaches to user interface design has been to support development through the use of meaningful abstractions to avoid dealing with low-level details. Despite such potential benefits, their adoption has

mainly been limited to professional designers, but new solutions [3] are recently emerging that are able to extend such approaches in order to achieve natural development by enabling end users to develop or modify interactive applications still using conceptual models, but with support that facilitates their development, analysis, and use.

Boris de Ruyter

With technologies moving from being deployed in passive and functional applications towards their application in reactive and intelligent solutions that integrate with our daily life, it becomes increasingly difficult to develop system functionality that will suit all end users. As part of the challenge to design such Ambient Intelligent systems that are acceptable for end users, we are faced with the quest for developing tools that enable end-users to create their own functionality. Research challenges in end-user development relate to interaction metaphors, tangible interaction tools and system platforms that are forward compatible. The following challenges for end-user programming can be formulated: 1) Keep it simple; Each small increase in complexity significantly reduces the number of potential users of the programming function. A solution should be found that allows users to adapt the system in a straightforward way. 2) Proper feedback; It is important that the user knows what he has programmed so far. The user should have immediate feedback on what is currently being programmed. 3) Make it fun; The user programming method should invite the user to tailor his system or environment to his specific needs and wishes. The user should feel like a creative composer whose attention can be focused on the desired composition itself rather than the process.

Citations

- [1] Benkler, Y. (2006) *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press, New Haven.
- [2] Beringer, Jörg (2004) Reducing Expertise Tension., End User Development, Communications of the ACM, Sept. 2004, Volume 47, Number 9, 39-40.
- [3] Berti, S., Paternò, F., Santoro, C. Natural Development of Nomadic Interfaces Based on Conceptual Descriptions, in *End User Development*, pp.143-160, Springer Verlag, 2006.
- [4] Costabile, M. F.; Fogli, D.; Mussio, P.; Piccinno, A., Visual Interactive Systems for End-User Development: A Model-Based Design Methodology, IEEE Trans. on Systems, Man and Cybernetics, part A- Systems and Humans, Vol. 37, Nov. 2007. pp.:1029 – 1046.
- [5] Fischer, G. (2002) *Beyond 'Couch Potatoes': From Consumers to Designers and Active Contributors, in FirstMonday (Peer-Reviewed Journal)*, Available at http://firstmonday.org/issues/issue7_12/fischer/.
- [6] Fischer, G., & Giaccardi, E. (2006) "Meta-Design: A Framework for the Future of End User Development." In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End User Development*, Springer Verlag, pp. 427-457.
- [7] Ko, A.J. and Myers, B.A., "A Framework and Methodology for Studying the Causes of Software Errors in Programming Systems." *Journal of Visual Languages and Comput.*, Feb, 2005. 16(1): pp. 41-84.
- [8] 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. April 24-29, 2004. Vienna: pp. 151-158.
- [9] Myers, B. A., Pane, J. F. and Ko, A. Natural Programming Languages and Environments. *Communications of the ACM*, Sept. 2004, 47, 9, 47-52.