New Perspectives in End-User Development

Fabio Paternò · Volker Wulf
Editors

New Perspectives in End-User Development

Springer

Editors
Fabio Paternò
ISTI - C.N.R
Pisa
Italy

Volker Wulf
Universität Siegen
Siegen
Germany

ISBN 978-3-319-60290-5 ISBN 978-3-319-60291-2 (eBook)
DOI 10.1007/978-3-319-60291-2

Library of Congress Control Number: 2017951239

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface – New Perspectives in End-User Development: Elaborating Upon a New Research Paradigm

More than a decade ago, we postulated that the design goals of human–computer interaction will evolve from *easy-to-use* to *easy-to-develop* applications (Lieberman et al., 2006). Recent trends show that this challenge is more important than ever. We need to design environments that allow users who do not necessarily have a background in programming to develop or modify their applications, with the ultimate aim of empowering them to flexibly employ digital services.

Since the origins of computing, hardware and software architectures have become more sophisticated, higher-level programming languages have been invented, and computer programming has evolved into a profession whose practices diverged from those of end users. The emergence of a (global) software industry was based on some interesting related aspects:

(a) development of ever more powerful hardware – following Moore's law for long time
(b) abstracting programs from the particularities of the hardware they ran on
(c) creating layered software architectures with well-defined interfaces to build upon each other
(d) abstracting software design from the particularities of specific work practices that the Information Technology (IT) artefacts were supposed to support

These developments contributed to the emergence of affordable and widely applied digital services. While in the beginning computing was restricted to rather few work domains in natural science, engineering, and accounting, we are now in the situation that IT is penetrating all aspects of life for a steadily increasing part of the world's population. So, the design of computer applications interacts with social practices in a vast variety of domains (Wulf et al., 2017). However, the software industry is still based on a division of labour between program creation at design time and use at run time.

With the increased spread of digital services and their resulting deep interaction with social practices, the traditional division of labour has become problematic, mainly for two reasons:

(1) Modern societies have become more and more differentiated in their patterns of life. Therefore, requirements for IT artefacts are very diverse and specific to individual application domains. The differentiated nature of software requirements is difficult to fully extract at design time for reasons of (a) problems in understanding the detailed social practices in all application domains, (b) lacking manpower in terms of software developers, and (c) economic efficiency.
(2) Social practices change rather dynamically. This is due to increasingly more dynamic environments in which organizations and individuals act. Moreover, the appropriation of digital services artefacts may impact social practices, individual qualifications and preferences, and may once again lead to new requirements for them.

The field of end-user development (EUD) emerged as an approach to overcoming these issues. In challenging the existing division of labour, EUD enables domain experts to (re-)design their applications in use – often at run time. In line with the Lieberman et al. (2006) definition, we suggest that:

> End-User Development should be understood as a set of methods, techniques, and tools that allow users who are acting as non-professional software developers of a specific application environment at some point to create, modify or extend an IT artefact.

The "development" concept in the term EUD has sociotechnical implications and thus indicates an important interdependency: on the one hand, it requires design environments enabling end users to modify their digital services; on the other hand, by appropriating the design environment, end users can potentially further develop their skills and practices. The immaterial property of software offers high-level technical flexibility in the sense that its functionality can be modified at any time – at least in theory.

EUD strives to change the traditional labour organization in the software industry by adding tools for end users to modify existing and to develop new applications. Existing software development cycles are still too slow to quickly respond to rapidly changing user needs of variegated categories of users, and professional developers often lack the needed domain knowledge to address such requirements, especially in pervasive modern applications (Ghiani et al., 2017). End users are generally neither skilled nor interested in adapting their applications at the same level as software professionals. So, EUD tools need to be appropriately crafted at application design time to anticipate technical flexibility that will be needed during their use.

Moreover, given new trends in hardware production, EUD does not even need to be understood as restricted to software. Technologies such as 3D printing or laser cutting enable end users to even modify aspects of hardware artefacts in use – thus also starting to challenge the traditional division of labour in hardware production (Ludwig et al., 2017).

The second aspect of the interdependency regards an emancipatory perspective on the development of human actors' capabilities and social practices. Engaging in adapting IT artefacts involves learning on the users' part and may lead to their personal and collective development. For an explicit investigation into this relationship see Dittrich et al. (in this volume), who discuss two cases of EUD in the context of organizations that depend on an IT infrastructure to provide their services. In both cases, EUD was used not only to personalize technical support but also to maintain and evolve the organizations' infrastructure. Thus, EUD was in both cases a constituent part of the innovation capability of the organizations. Therefore, EUD also has a societal dimension since it enables the codesign of work places as well as the full participation of citizens in the emerging Information Society (Fischer et al. in this volume).

Possibilities for EUD need to be intentionally designed into the application environment. Since EUD environments do not typically allow users to fundamentally redesign software architectures, EUD requires foreseen in which aspects of an applications' functionality will remain stable over time (Stiemerling et al., 1997; Stevens et al., 2006). Cabitza and Simone (in this volume) suggest a layered perspective on the architectural design of malleable applications. Wulf et al. (2008) explored opportunities for such architectures in the context of a component-oriented software paradigm.

One important issue is how to design the tools to support application personalization, specifically the level of complexity to offer from a user's point of view. The literature provides different classification schema of the technical means by which end users could be enabled to modify their IT artefacts (e.g. Henderson & Kyng, 1991; Morch et al., 2004). Lieberman et al. (2006) distinguish broadly between parameterization and customization as well as program modification and creation.

It is generally assumed that an EUD-friendly design environment should enable a seamless move from the usage mode of interaction towards an adaptation mode (Wulf & Golombek, 2001). Additionally, the different levels of adaptations should be designed in a way that the transition towards higher levels of complexity is supported. MacLean et al. (1990) suggested the design metaphor of a "gentle slope of complexity." AI techniques, for example, adaptivity may play a role in enabling the different transitions and support certain EUD activities. They typically result in mixed forms of interactions where adaptive features can support interaction but users can still take the initiative in the development process and may provide interesting results.

Since EUD is a sociotechnical activity, it requires analysing how to empower development in its interdependent sociotechnical aspects. Blackwell et al. (in this volume) aim to categorize the differences among end-user developers from a psychological perspective. Future work will address the design implications of such investigations.

There is definitively a collaborative dimension in EUD activities (Mørch & Mehandjiev, 2000; Wulf, 1999; Kahler, 2001). Actors learn from each other and cooperate when conducting EUD. A routinization of such collaborative patterns can lead to a division of labour among end users in conducting adaptations and sharing tailored artefacts of different types and levels of complexity. Supporting

collaborative patterns in EUD is also an interesting theme in design research. This line of research includes recommendations and awareness mechanisms for finding suitable EUD expertise as well as reusable artefacts. EUD-related communities allow end users to share EUD-related knowledge and artefacts with their peers (Costabile et al., 2003; Pipek & Kahler, 2006; Draxler & Stevens, 2011).

There are other approaches to differentiate the division of labour in the software industry by involving users more intensively into the design process; examples of such approaches are Open Source Development, Software Ecosystems, Prototyping, Participatory Design, Agile methods (see, for instance, Diaz et al. in this volume). However, they all focus on design time activities.

EUD is an activity with sociocultural implications, depending on place, time, and people involved. This is particularly true with the explosion of mobile technologies, which has made it possible for people to access their applications from a variety of contexts of use that differ in terms of available devices, things, and services, and that require specific actions when various types of events occur. Differences in EUD practice are likely to develop for different application scenarios, cultures, and languages. These differences may relate to who is in control of EUD activities, the relation between individual and collaborative EUD, and how communities of end-user developers are organized.

At the same time, theory-oriented research in EUD has a long history and may also contribute to the community's efforts towards engineering and reengineering software applications. For example, deSouza (in this volume) discusses the use of Semiotic Engineering to stimulate design-oriented EUD research from a specific conceptual perspective. The chapter by Burnett et al. (in this volume) discusses how theoretical foundations may facilitate the transferability of insights beyond individual tools to the creation of generally applicable methods and principles for other researchers to draw upon.

Comparing the current technological scenario with the state of the art when the first EUD book was published in 2006, the most important technological revolution has been the advent of the Internet of Things. Our life is now characterized by the presence of a multitude of sensors, objects, and devices. This technological trend has posed new challenges for EUD as well. Paternò and Santoro (in this volume) discuss a framework that provides opportunities to identify important aspects to be considered when analysing EUD in Internet of Things domains. In this area, Diaz et al. (in this volume) discuss tools to support the ideation, design, and early prototyping of augmented experience.

This book also presents examples of how EUD research has expanded into specifically interesting and emerging domains: Menestrina and De Angeli discuss how computer games can benefit from a EUD approach, in particular those games designed for a purpose other than entertainment; Valtolina and Barriccelli report on their experience with an EUD framework to support the "quantified self" concept during sport activities; Morch et al. speak about their experience concerning EUD and learning in the 3D virtual world Second Life; Reuter et al. discuss how EUD can support the gathering and assessment process of data from social networks in emergency situations.

From a technological perspective, the Web is the most diffuse and penetrating technological infrastructure. Various mashup environments have been proposed to support the development of new applications starting with components of existing ones. Ardito et al. show how they can be exploited within a three-layer meta-design model. In this area, Aldalur et al. provide a review of Web Augmentation technologies (aimed at improving existing Web applications) as tools and techniques for EUD.

New application domains and emerging new technologies drive innovations in EUD. A key question is how to evaluate these innovations. Tetteroo and Markopulos (in this volume) and Ludwig et al. (in this volume) suggest that innovative EUD solutions need to be explored in practice. While laboratory evaluations or short-term rollouts can be found rather frequently in the literature, these methods do not provide a sufficient understanding regarding the appropriation of EUD technologies in social practices and how these technologies should be improved to encourage such practices (Wulf et al., 2017). Tetteroo and Markopulos discuss challenges pertaining to field deployments based on their experiences in the healthcare sector, coming up with some possible guidelines for the evaluation of EUD technologies.

Overall, we can see that in the last ten years there have been considerable research efforts to establishing the new EUD paradigm in all its different methodological aspects and application domains. Several chapters of this book report on long-term research strategies conducted by individual groups. For example, Myers et al. (in this volume) report on their efforts aiming to better understand how end users think about their tasks, and how to support them to express those tasks in ways closer to the way they think.

While we have better understood certain concepts and design implications of the EUD paradigm, we also realize that generally applicable solutions are (still) missing, important new application domains are materializing (e.g. customizing robot behaviour, personalizing ambient-assisted living, adapting smart home objects), and further research is required to identify how to exploit the potentialities of the EUD paradigm. In this context, we will have to better understand how to apply given insights to new problem domains.

So, we hope you will join us in this fascinating research endeavour!

References

Costabile, M.F., Fogli, D., et al. (2003). Building environments for end-user development and tailoring. In *IEEE symposia on human centric computing languages and environments*, Auckland.

Draxler, S., & Stevens, G. (2011). Supporting the collaborative appropriation of an open software ecosystem. *Computer supported cooperative work (CSCW), 20*(4–5), 403–448.

Ghiani, G., Manca, M., Paternò, F., Santoro, C. (2017). Personalization of context-dependent applications through trigger-action rules. *ACM Transactions*

on Computer-Human Interaction, 24(2), Article N.14, April 2017.80). Programs, Life Cycles, and Laws of Software Evolution. IEEE 68.

Henderson, A., & Kyng, M. (1991). There's no place like home: continuing design in use. In J. Greenbaum, M. King, *Design at work - cooperative design of computer systems* (pp. 219–240). NJ, USA: L. Erlbaum Associates Inc. Hillsdale.

Kahler, H. (2001). *Supporting Collaborative Tailoring*. Roskilde: Department of Communication, Journalism and Computer Science, Roskilde University.

Lieberman, H., Paternó, F., Wulf, V. (Eds.). (2006). End user development. London: Springer.

Lieberman, H., Paternò, F., Klann, M., Wulf, V. (2006). End-user development: an emerging paradigm. In Lieberman, H., et al. (Eds.), *End user development* (pp. 1–8). London: Springer.

Ludwig, T., Boden, A., Pipek, V. (2017). 3D printers as sociable technologies: taking appropriation infrastructures to the internet of things. *Transactions on CHI, 24*(2), 17:1–17:28.

MacLean, A., Carter, K., Lövstrand, L., Moran, T. (1990, April 1–5). User-tailorable systems: pressing the issue with buttons. In *Proceedings of the conference on computer human interaction (CHI '90)* (pp. 175–182). Seattle: ACM-Press.

Mørch, A.I., & Mehandjiev, N.D. (2000). Tailoring as collaboration: the mediating role of multiple representations and application units. *Computer Supported Cooperative Work, 9*(1), 75–100.

Mørch, A. I., Stevens, G., Won, M., Klann, M., Dittrich, Y., Wulf, V. (2004). Component-based technologies for end-user development. *Communications of the ACM*, *47*(9), 59–62.

Pipek, V., & Kahler, H. (2006). Supporting collaborative tailoring. In H. Lieberman, et al. (Eds.), *End user development* (pp. 315–354). London: Springer.

Stevens, G., Quaisser, G., Klann, M. (2006). Breaking it up: an industrial case study of component-based tailorable software design. In H.Lieberman, F. Paternò, V. Wulf, *End user development* (pp. 269–294). London: Springer.

Stiemerling, O., Kahler, H., Wulf, V. (1997). How to make software softer - designing tailorable applications. In *Proceedings of the ACM symposium on designing interactive systems (DIS 97), 18. - 20.8.1997* (pp. 365–376). Amsterdam, NL: ACM-Press.

Wulf, V. (1999). "Let's see your Search-Tool!" - collaborative use of tailored artifacts in groupware. In *Proceedings of GROUP '99* (pp. 50–60). New York: ACM-Press.

Wulf, V., & Golombek, B. (2001). Direct activation: a concept to encourage tailoring activities. *Behaviour & Information Technology, 20*(4), 249–263.

Wulf, V., Pipek, V., Randall, D., Rohde, M., Schmidt, K., Stevens, G. (Eds.). (2017). *Socio informatics – a practice-based perspective on the design and use of IT artefacts*. Oxford: Oxford University Press.

Wulf, V., Pipek, V., Won, M. (2008). Component-based tailorability: Enabling highly flexible software applications. *International Journal of Human-Computer Studies*, *66*(1), 1–22.

Contents

Contributors

Ignacio Aedo Universidad Carlos III de Madrid, Madrid, Spain

Iñigo Aldalur University of the Basque Country (UPV/EHU), San Sebastián, Spain

Antonella De Angeli University of Lincoln, Lincoln, United Kingdom

Carmelo Ardito Università degli Studi di Bari Aldo Moro, Bari, Italy

Barbara R. Barricelli Università degli Studi di Milano, Milan, Italy

Laura Beckwith Configit, Atlanta, GA, United States

Andrea Bellucci Universidad Carlos III de Madrid, Madrid, Spain

Alan F. Blackwell University of Cambridge, Cambridge, United Kingdom

Johan Bolmsten World Maritime University, Malmö, Sweden

Margaret Burnett Oregon State University, Corvallis, OR, United States

Federico Cabitza University of Milano Bicocca, Milano, Italy

Jill Cao Oregon State University, Corvallis, OR, United States

Valentina Caruso Swiss Federal Institute for Vocational Education and Training, Zollikofen, Switzerland

Kerry Chang IBM, Armonk, NY, United States

Maria Francesca Costabile Università degli Studi di Bari Aldo Moro, Bari, Italy

Julian Dax University of Siegen, Siegen, Germany

Giuseppe Desolda Università degli Studi di Bari Aldo Moro, Bari, Italy

Oscar Díaz University of the Basque Country (UPV/EHU), San Sebastián, Spain

Paloma Díaz Universidad Carlos III de Madrid, Madrid, Spain

Yvonne Dittrich IT University, Copenhagen, Denmark

Jeanette Eriksson Malmö University, Malmö, Sweden

Shannon Ernst Oregon State University, Corvallis, OR, United States

Gerhard Fischer University of Colorado, Boulder, CO, United States

Daniela Fogli University of Brescia, Brescia, Italy

Valentina Grigoreanu Microsoft, Washington, DC, United States

Melissa M. Hartley West Virginia University, Morgantown, WV, United States

William Jernigan Oregon State University, Corvallis, OR, United States

Marc-André Kaufhold University of Siegen, Siegen, Germany

Mary Beth Kery Carnegie Mellon University, Pittsburgh, PA, United States

Andrew J. Ko University of Washington, Seattle, WA, United States

Todd Kulesza Microsoft, Washington, DC, United States

Toby Jia-Jun Li Carnegie Mellon University, Pittsburgh, PA, United States

Thomas Ludwig University of Siegen, Siegen, Germany

Panos Markopoulos Eindhoven University of Technology, Eindhoven, Netherlands

Maristella Matera Politecnico di Milano, Piazza Leonardo da Vinci, Milano, Italy

Zeno Menestrina University of Trento, Povo, Italy

Anders I. Mørch University of Oslo, Oslo, Norway

Brad A. Myers Carnegie Mellon University, Pittsburgh, PA, United States

Alannah Oleson Oregon State University, Corvallis, OR, United States

Stephen Oney University of Michigan, Ann Arbor, MI, United States

Philippe Palanque University of Toulouse, Toulouse, France

Fabio Paternò CNR-ISTI, HIIS Laboratory, Pisa, Italy

Antonio Piccinno University of Bari “Aldo Moro”, Bari, Italy

Volkmar Pipek University of Siegen, Siegen, Germany

Christian Reuter University of Siegen, Siegen, Germany

Carmen Santoro CNR-ISTI, HIIS Laboratory, Pisa, Italy

Chris Scaffidi Oregon State University, Corvallis, OR, United States

Carla Simone University of Siegen, Siegen, Germany

Clarisse Sieckenius de Souza Semiotic Engineering Research Group (SERG), PUC-Rio, Rio de Janeiro, Brasil

Daniel Tetteroo Eindhoven University of Technology, Eindhoven, Netherlands

Stefano Valtolina Università degli Studi di Milano, Milan, Italy

Marco Winckler University of Toulouse, Toulouse, France

Volker Wulf University of Siegen, Siegen, Germany

YoungSeok Yoon Google, Mountain View, CA, United States