

Dagstuhl Seminar 07081: End-User Software Engineering

Margaret M. Burnett, Gregor Engels, Brad Myers, Gregg Rothermel

1 March, 2007

The number of end users creating software is far larger than the number of professional programmers. These end users are using various languages and programming systems to create software in forms such as spreadsheets, dynamic web applications, and scientific simulations. This software needs to be sufficiently dependable, but substantial evidence suggests that it is not.

Solving these problems involves not just software engineering issues, but also several challenges related to the users that the end user software engineering intends to benefit. End users have very different training and background, and face different motivations and work constraints, than professional programmers. They are not likely to know about such things as quality control mechanisms, formal development processes, system models, language design characteristics, or test adequacy criteria, and are not likely to invest time learning about them.

It is important to find ways to help these users pursue their goals, while also alerting them to dependability problems, and assist them with their explorations into those problems. Further, it is important to work within the contexts with which these users are familiar, which can include programming environments that have not been directly considered by software engineering or programming languages researchers.

These challenges require collaborations by teams of researchers from various computer science subfields, including specialists in end-user-programming (EUP) and end-user development (EUD), researchers expert in software engineering methodologies and programming language design, human-computer interaction experts focusing on end-user programming, and empiricists who can evaluate emerging results and help understand fundamental issues in supporting end-user problem solving. Collaborations with industrial partners must also be established, to help ensure that the real needs of end-user programming environments in industry are met.

This Dagstuhl seminar was organized in order to bring together researchers from these various groups and with the various appropriate backgrounds, along with an appropriate selection of industrial participants. The seminar allowed the participants to work together on the challenges faced in helping end-user programmers create dependable software, and on the opportunities for research addressing these challenges. Our goals were to help these researchers better understand (1) the problems that exist for end-user programmers, (2) the environments, domains and languages in which those programmers create software, (3) the types of computing methodologies (especially in the areas of software engineering and programming language design) that can be brought to bear on these problems and in these domains, and (4) the issues that impact the success of research in this area. In addition, an overarching goal was to

build awareness of the interdisciplinary connections and opportunities that exist for researchers working in the area.

The seminar included several tutorial-style presentations by experts on software engineering, programming languages, human-computer interaction, and empirical studies in relation to end-user software engineering. The program was complemented with brief presentations by some participants on topics of a more specialized nature, grouped into sessions on related topics. We also incorporated system demonstrations of prototypes and environments relevant to the topics. Ample time was allowed for interactive discussion sessions.

Most of the seminar participants provided white papers summarizing their primary interests in the area, including work that they are doing and open problems. These white papers are compiled into the seminar proceedings. Additional contributions to the seminar were provided as slides, and are available on the Dagstuhl website for the seminar.