

1 HUMAN-CENTERED SOFTWARE ENGINEERING: SOFTWARE ENGINEERING ARCHITECTURES, PATTERNS, AND MODELS FOR HUMAN COMPUTER INTERACTION

Ahmed Seffah*,
Jean Vanderdonckt**, and Michel C. Desmarais***

*Human-Centered Software Engineering Group, Concordia University, Canada

**Université catholique de Louvain, Belgium

***École Polytechnique de Montréal, Canada

1.1 SCOPE

The Computer-Human Interaction and Software Engineering (CHISE) series of edited volumes originated from a number of workshops and discussions over the latest research and developments in the field of Human Computer Interaction (HCI) and Software Engineering (SE) integration, convergence and cross-pollination. A first volume in this series (CHISE Volume I – Human-Centered Software Engineering: Integrating Usability in the Development Lifecycle) aims at bridging the gap between the field of SE and HCI, and addresses specifically the concerns of integrating usability and user-centered systems design methods and tools into the software development life-cycle and practices. This has been done by defining techniques, tools and practices

that can fit into the entire software engineering lifecycle as well as by defining ways of addressing the knowledge and skills needed, and the attitudes and basic values that a user-centered development methodology requires. The first volume has been edited as Vol. 8 in the Springer HCI Series (Seffah, Gulliksen and Desmarais, 2005).

1.2 SPECIFIC OBJECTIVES OF THE CHISE VOLUME II

In CHISE Volume I, we looked at means of integrating HCI and usability engineering techniques into the well-established software engineering lifecycles which, many of us claim, often lack the human-centered quality required for the development of highly interactive systems. In this volume, we look at the converse side of this issue. A development process may be dominated by HCI concerns and we may fail to integrate SE method into it. We also look at means of bringing more formal and systematic methods, and better tools for the HCI design and development of interactive systems.

A good example of how SE methods can have a positive impact in the design and development of interactive systems is in the transformation of the Web development process, from a print paradigm of online brochureware, to complex and highly interactive Web-based systems. Web user interfaces (UIs) and HCI designers have to consider new challenges such as content management, security versus usability, information architecture and navigation support, etc. Besides Web engineering, there has also been significant progress in terms of development approaches including component-based engineering, separation of software concerns, model-driven architecture, formal specification methods, and reengineering techniques. Such methods are certainly useful and needed by HCI researchers and practitioners. They help design the user interaction with the system and build more usable interfaces.

However, these methods are not sufficient yet. New methods and ideas are needed to address the particular issues of interaction engineering. They will borrow many of the software engineering's fundamental concepts and principles of SE emphasizing the same technical and management activities.

1.3 OVERVIEW

This second volume aim to establish a meaningful dialog between the HCI community and SE practitioners and researchers on the results (both good and bad), obstacles, and lessons learned associated with applying software development practices in the field of UI. The book will provide accounts of the application of SE practices (which may be principles, techniques, tools, methods, processes, etc.) to a specific domain or to the development of a significant interactive system.

This second volume includes 16 chapters organized in four sections. They provide a roadmap for the future of UI SE especially how emerging SE techniques can help HCI. The emphasis is put on the following topics:

- Software engineering models, methods and tools for challenging HCI issues such as adaptability, universal usability and accessibility
- Software architectures for HCI
- Model-driven architectures and engineering (MDA, MDE)