Preface

At a workshop I attended not long ago, my colleague Matthew Chalmers made the observation that computer science is based entirely on philosophy of the pre-1930s. Computer-science in practice involves reducing high-level behaviors to low-level, mechanical explanations, formalizing them through pure scientific rationality; in this, computer science reveals its history as part of a positivist, reductionist tradition. Similarly, much of contemporary cognitive science is based on a rigorous Cartesian separation between mind and matter, cognition and action. These are philosophical positions of long standing, dating from the nineteenth century or earlier. However, they have been under continual assault since around the 1930s, when philosophers such as Martin Heidegger and Ludwig Wittgenstein began to articulate radically new positions on cognition, language, and meaning. This new approach abandoned the idea of disembodied rationality and replaced it with a model of situated agents, at large in the world, and acting and interacting within it. Practical action and everyday experience replaced abstract reasoning and objective meaning as the foundations of a philosophical psychology.

Why should any of this matter? Surely computer science has nothing to do with philosophy? It is an engineering discipline, and the testaments to its success are all around us, including the laptop I am using to write this preface. Arguments between philosophers scarcely seem to have held back the course of technological progress, have they?

This is all true, of course, but it hides a deeper truth. The development and application of computational technologies is an engineering discipline, and one that has been spectacularly successful over the past fifty or sixty years.
However, it is most certainly also a philosophical enterprise. It is philosophical in the way it represents the world, in the way it creates and manipulates models of reality, of people, and of action. Every piece of software reflects an uncountable number of philosophical commitments and perspectives without which it could never be created. Software depends inevitably on our ideas about representation and reality. Phil Agre comments, “Technology at present is covert philosophy; the point is to make it openly philosophical” (Agre 1997:240). Agre’s primary focus was the area of Artificial Intelligence (AI), but the philosophical underpinnings of computer science are every bit as significant in the area of Human-Computer Interaction (HCI) that I will be addressing here. While any software system introduces some kind of formalization of the world, HCI (like AI) deals with formalizations of human cognition and activity. These are the issues that have lain at the heart of philosophical debate for centuries. In some ways, it would be hard to imagine a more philosophical enterprise.

Philosophy tends to get short shrift in technical circles. The word itself is often used to denote fuzzy, muddled, and fundamentally irrelevant reasoning. Debates over philosophical foundations seem irrelevant. However, if our technical practice is built on those foundations, then the arguments are deeply relevant, because they determine the limits of what can be done and the chances for success of our efforts to have people and computers work effectively together. This argues that we need to uncover the philosophical assumptions that run throughout both the theory and practice of computer system design, and understand what kind of intellectual commitments are being made.

My goal here, though, is more modest. Reexamining the philosophical foundations of computer science is a task for another person or another time. Instead, I will focus on one particular way in which these philosophical questions have lately arisen in the area of HCl. Recent research activities, focusing on what I call “embodied interaction,” reflect the situated, embodied perspective of Heidegger, Wittgenstein, and other post-1930s philosophers. In the pages that follow, I will explore this perspective and show how it applies to these new research programs, and how, by looking at the philosophical background, we can begin to understand the foundational underpinnings of these new approaches. The goal is to develop an understanding that explains the relationship between the various elements of the
embodied interaction approach, and that begins to suggest how, when, and why embodied interaction works.

So, this book is about “where the action is,” in two ways. First, it is about a perspective that places the action of embodied agents center stage. Rather than take action to be generated from or subservient to abstract reasoning, the perspective I will explore here sees embodied practical action in the world as the foundation for our conscious experience. Second, this approach is “where the action is” in the sense that it provides a way to understand the contributions and opportunities emerging from dynamic new forms of technological practice.

Acknowledgments

No book is ever the work of one person alone, whatever the cover says. That is especially true of one that, as this does, tries to draw together material from many different places. I owe a great deal to the various people who helped me see connections, guided me as I ventured into new areas, and supported the whole process.

This book was written while I was a member of the research staff in the Computer Science Lab at Xerox’s Palo Alto Research Center (PARC). My area manager, Karin Petersen, and lab manager, Richard Bruce, were endlessly supportive of my efforts, and the time I spent working on the manuscript, despite the fact that it took much longer than I think any of us had anticipated. PARC is a wonderfully interdisciplinary environment, and while working both there and at its European sister laboratory, I had the opportunity to work with and learn from many people whose insights, one way or another, are reflected here (although perhaps not in ways that they would support). These include Annette Adler, Bob Anderson, Victoria Bellotti, Graham Button, Gregor Kiczales, Steve Harrison, Austin Henderson, John Lamping, Wendy Maclay, Brian Cantwell Smith, and Lucy Suchman.

Without Mark Weiser’s initial encouragement as I first attempted to articulate these ideas, the book would probably never have been written. He was an endless source of insight and optimism; his enthusiasm, energy, and irreverence made him an inspiring colleague. Sadly, Mark passed away while this book was being written. He is hugely missed.
Preface

The work reported here also has its roots in research conducted in the Discourse Architecture Laboratory at Apple Computer during 1996 and 1997, and the many useful and insightful interactions I had with my colleagues there, including David Curbow, Allen Cypher, Nicholas Damiris, Tom Erickson, Jed Harris, Austin Henderson (again), and Don Norman.

Beyond these, a network of sustaining friends and colleagues have provided both intellectual and emotional sustenance and valuable feedback on earlier presentations of this material. I am particularly indebted to Matthew Chalmers, Judith Donath, Geraldine Fitzpatrick, Beki Grinter, Kristina Höök, Hiroshi Ishii, Natalie Jeremijenko, and Beth Mynatt for discussions about the material here and comments on earlier drafts. Bob Prior at MIT Press was also a great supporter and deftly helped steer this book through the whole process. Deborah Cantor-Adams smoothly managed the production of the book itself, while Danyel Fisher, Wayne Lutters, Jack Muramatsu, Madhu Reddy, and Suzanne Schaefer provided last-minute assistance reading and correcting proofs.

Most of all, my thanks to Melinda Stelzer, my love and inspiration, who has endured my being absent and preoccupied for much longer than either of us anticipated. And also to Elwood, of course, for his own special contributions.