MOBILE COMPUTING PRINCIPLES

Written to address technical concerns that mobile developers face regardless of the platform (J2ME, WAP, Windows CE, etc.), this book explores the differences between mobile and stationary applications and the architectural and software development concepts needed to build a mobile application. Using UML as a tool, Reza B'Far guides the developer through the development process, showing how to document the design and implementation of the application. He focuses on general concepts while using platforms as examples or as possible tools.

After introducing UML, XML, and the derivative tools necessary for developing mobile software applications, B'Far shows how to build user interfaces for mobile applications. He covers location sensitivity, wireless connectivity, mobile agents, data synchronization, security, and push-based technologies and finally discusses the practical issues of mobile application development including the development cycle for mobile applications, testing mobile applications, and architectural concerns. These are illustrated with a case study.

Reza B'Far (Behravanfar) is an executive consultant currently serving as the CTO of Voice Genesis and Acting CTO of Semantic Messaging Systems Inc. His company, Cienecs Inc., has had a variety of engagements in the mobile arena with startups as well as Fortune 500 companies. Early in his career, he worked for Weyerhaueser Company, Parr & Associates Inc., and the National Oceanic Research Department of NASA. He has spent the past ten years working for Noor Electrical Engineering, Virtual Mortgage Network, AdForce Inc., eBuilt Inc., and Data Trace Corporation. He is currently an independent contractor working with a variety of companies as an architect and/or CTO, including some in the mobile arena.

MOBILE COMPUTING PRINCIPLES

DESIGNING AND DEVELOPING MOBILE APPLICATIONS WITH UML AND XML

REZA B'FAR
Cienecs Inc.

Foreword by ROY T. FIELDING

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2005

Printed in the United States of America

Typefaces ITC Berkeley Oldstyle 10.75/13 pt. and ITC Franklin Gothic *System* LaTeX 2ε [TB]

A catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

B'Far, Reza.

Mobile computing principles : designing and developing mobile applications with UML and XML / Reza B'Far ; foreword by Roy T. Fielding.

p. cm.

Includes bibliographical references and index.

ISBN 0-521-81733-1

1. Mobile computing. 2. UML (Computer science) 3. XML (Document markup language) I. Title.

QA76.59.B43 2004
005.268 – dc22 2004045705

ISBN 0 521 81733 1 hardback

Contents

Foreword

Back and forth, back and forth . . .

Four years ago, Reza and I were working together at eBuilt when he first stopped by my office to talk about frameworks for wireless application development. We were in the final months of the so-called "dot-com era," when dreams of a new economy allowed just about anyone to get funding for a network-based application, particularly when it also involved some form of mobile computing device. Those people with ideas (and sometimes funding) would come to our company and ask us to implement their vision. Of course, they would also ask for a few miracles, such as a working prototype within a month and deployment across all devices in six months. Oddly enough, we could actually accomplish implementations like that, if it were not for one problem out of our control: mobile devices had a market lifetime of only about four months.

It was the year 2000, just a couple months after Y2K became a non-issue, and there was so much variance in the types of mobile devices, both in terms of their feature sets and in their application development environments, that an application developed for one device environment would be obsolete by the time it was ready to market. Reza had a solution in mind, which is why he was busy pacing in my office. Back and forth, back and forth, all the while explaining to me why eBuilt needed a device-independent application development environment and how we might sell such an environment to other software organizations.

This was prior to the eventual unification of platforms around base operating systems, such as PalmOS, Symbian, and J2ME, and about the same time that device manufacturers realized the impact of design turnover on device sales: innovation had become so frenzied that most of the application developers simply could not keep up. Unfortunately, eBuilt did not have the resources and necessary alliances with device manufacturers to pursue Reza's vision, aside from one project at a time, but he never gave up on the general idea. That is demonstrated by the enormous amount of information and effort he has put into this book.

The funny thing about "mobile computing" is that mobility is the easy part. What is actually of interest to the consumer, and hence to those who need to sell to the consumer, is computing despite mobility. There is a small segment of the population who will buy a new device purely for the sake of its coolness, but mass appeal does not come until there exists an application that is sufficiently compelling to justify purchasing (and carrying around) a new device.

Like most people whose work involves a lot of travel, I think most about mobile computing when I suffer from the lack of it. While I am writing this, my wife and I are on our first real vacation together: a late honeymoon trip to Italy. Our first day of travel involved 27 hours of planes, trains, and automobiles, in which the limitations of current mobile computing have been readily apparent. We are so close to a world in which all of the information needed is available, when and where we need it, and yet I knowingly embarked on this trip without my cellphone (CDMA doesn't work in Europe), laptop (too heavy, expensive, and tempting of work), or even a PDA. In fact, the only technology we have with us are two wristwatches and a new digital camera.

I used the Web to purchase all of our tickets and accommodations in advance, something that was unthinkable just ten years ago. However, even a well-planned trip is susceptible to change. What is traffic like to the airport? Should we go up the coast or take the freeway? Is our flight on time? What terminal? Do we have time to park in the remote lot? Those are just the basic questions that fill my mind while readying the car. The more complex question is this: can we get better seats on the flight? I wouldn't even have considered such a question a few years ago, but today it is possible to store my itinerary on the airline's Web site, access it from any Web browser, and make use of a visual diagram for discovering what seats are available on each leg of the flight. That is great design, even though it assumes a broadband connection to the Internet and a full-color 1024 × 768 display.

I know there are mobile devices on the market that can answer my questions (i.e., perform my application), if only they had the software to do so. I can buy a five-ounce PDA with built-in 802.11b and bluetooth wireless connectivity, a bluetooth GPS device to provide geographical positioning, card-slot memory for gigabytes of data, and a color TFT display that is just as clear as a laptop LCD screen (if not more so). In addition to the airline's Web site, there are real-time traffic maps available on the Internet for the freeways in Southern California.

All I really need is an application that monitors my itinerary, collects data from the appropriate sources whenever it can do so, and notifies me when conditions change (or at least makes the information continuously available so that I can read it at the push of a button).

Unfortunately, the mobility of software is considerably behind that of hardware devices. An 802.11b interface can automatically detect and switch from one hotspot to another, but the device software will invariably ask the user if they wish to do so each time—it seems that folks haven't considered the option of pre-approving a set of wireless carriers for automatic switch-over. Likewise, applications that expect a network interface to exist tend to drop like flies in the presence of intermittent connectivity, and geographic applications don't understand the concept of a device that is only occasionally within range of a GPS. I

can't really blame this state of affairs on the device manufacturers—after all, they are building devices that are intended to be generic and thus usable for many different applications.

My travel assistant application isn't a particularly novel vision of mobile computing. Whether it be called ubiquitous computing or mobility-aware applications, the desire for continuous information support has been imagined, if not expressed, by countless technologists as they rush to meet their next travel connection for some far-away conference at which techno-visionaries are sure to speak about their latest advances in shrinking hardware into lighter but less useful forms. The hardware, networking, and network-accessible information is already available to support a mobile travel assistant, and yet I felt no compelling need to buy a new device this past year. That is, other than our new digital camera.

Digital Camera?

I already had one digital camera, but my wife wanted something a little smaller. Something inconspicuous, fitting within her purse. In other words, something a little more mobile. What we bought has a four-megapixel CCD, internal clock, high-density TFT display, AV-output port (supporting both NTSC and PAL formats), USB interface, and a CPU with sufficient computing power to obtain, compress, and store a four-MB image in less than a second (or a small-format movie at 24 frames per second). It weighs five ounces, uses a standard flash card for storage (a 256-MB card at the moment), and costs roughly the same as the PDA described above. Sales of digital cameras are pretty hot right now, judging from the digital print services that have cropped up all over the place. Why? Because they are selling an application (personal photography) on a device that provides all of the traditional affordances (user interface controls) of a film-based camera. It just happens to also be a device capable of mobile computing. In fact, the only reason I do not classify our camera as a mobile computing device is that its firmware has no built-in support for communicating directly on a network, even though its USB interface is more than capable of doing so.

Would it make sense to add networking capabilities to the camera? It would be nice to upload pictures directly to our personal Web site. There are, after all, many other noncamera features within the firmware, such as running a slideshow via the AV interface and the ability to postprocess images for special effects. Camera firmware, though, is just as proprietary as the mobile devices of 2000. Eventually, to keep up with requests for new functionality, camera manufacturers will have to move to more modular designs based on common platforms. I can only hope that, in doing so, they do not succumb to the same mistakes as the cellphone and PDA manufacturers: adding low-tech camera lenses as a feature suitable only for toy use.

A truly modular device would consist of a self-contained camera with almost all of the features of our new camera, a self-contained PDA with almost all the features one would expect to find in a PDA, a self-contained GPS unit that tells everything in range where they are, and a self-contained wireless communication device that

services the other devices in much the same way that consumer firewall/gateway devices service computers on a home network. That is, essentially, the way that bluetooth is intended to work. Communication alone, however, is not sufficient: we need platforms that are capable of recognizing such interfaces (even when they are inactive) and flexible enough to select the one that is best used for image capture, the one that is best used for display, the one that can be used for Web-based retrieval, and the several that are available for "storage." A common platform allows application development to mature despite the rapid pace of device evolution, which allows software developers to build interesting applications before their platforms become obsolete, which in turn gives consumers a reason to buy devices that do something useful for them (computing despite their mobility), driving further demand for that platform of devices.

Therefore, while reading this book, I hope that you keep in mind that the above describes not a single technology development, but rather the development of a system that is intended, if successful, to become a self-sustaining feedback loop. Just as the Web has become the preferred platform for successful Internet services, one of the platforms that Reza describes herein will become the basis for future mobile applications. It will be up to you to determine which one, because it is the application developers that drive consumer demand.

Roy T. Fielding
Somewhere between Laguna Beach,
California, and Venice, Italy
January 2004

Acknowledgments

There are so many people who I need to thank. And since this may be my one and only opportunity to thank them in a semipublic forum, then I must take the opportunity to do so. I thank Robert Gottesman for his support throughout my professional endeavors and David Armstrong, Dr. Roy Fielding, and Phillip Lindsay for their mentorship. Many thanks are owed to my loyal clients (Charity Funding Services and Barney Mckinley, Voice Genesis, eBuilt Inc., and a few others) that helped me put food on the table while I was authoring this text. Also, thanks are owed to Dr. Dennis H. Parr for having given me an opportunity when few would and to Abe and Najmeh Khadem who helped me with a monumental problem and showed me the road to Southern California, the best place on earth. Many thanks go to Brenda and Roger Eeds, who have become my lifelong friends with their advice and support, and to Nima Oreizy, Jos Bergmans, Mark Scheele, and Mark Mariott, who have become closer friends during the time of authoring this text.

I thank Susan Boettger, who has recently become my wife, for her encouragement in life and her family, Laura, Jane, and John Boettger, for their friendship. I am certain that her achievements musically as a classical composer and a pianist will dwarf mine as an engineer.

I dedicate this book to my wife; to my mother Shahrnaz Karimzadeh, who fostered a notion of excellence and sent me to the greatest country—the United States of America—that I call home; to my uncle and aunt Amir and Shauna Karimzadeh, who stuck with me all the way; to Bill, Deena, Scott, and Cindy Bernhardt, who made their home my home as I grew from a teenager to an adult; to Durrelle Singleton, who I know always has my back covered; to my brothers Abdollah and Atta, whom I miss; and to my little cousins Nikoo and Natasha, who embody innocence and purity to me.

Finally, I want to thank Jim Turner of SIGS, Lothlorien Homet, formerly of Cambridge University Press, and Lauren Cowles and Katherine Hew, currently of Cambridge University Press, who all had something to do with my deciding to write this book and helping me to carry it out. And I thank Albrecht Muller whose input has been invaluable during the reviews.

To no-look passes by Magic, hook shots by Kareem, dunks by Dr. J, and finger-roles by the Ice Man. To Steve Urkel's laugh, Getting' Jiggy with It, *Revenge of the Nerds*, Trekkers, and comebacks by number 7 in Mile High, which inspire me never to give up. To Terri Tower and Elizabeth Elam: Whoop, there it is, I did it my way! To Beethoven, Gorecki, Arvo Part, Lauridson, Red Hot Chili Peppers, Sting, Durufle, Delerium, and Dire Straits for keeping me company late at nights and to the following: It's all about old school, it'll never be as good as it was, and the older I get the better I was. I only wish I could invent phrases like Dicky V., to whom I say "here is a diaper dandy writer for you baby" and John Madden, to whom I say "I stuck to it for two and a half years!" I wish anyways. To slow-mo, Flo-Jo, go Bo, go. To Andre, who looks a ton better bald than with hair, and to Freddy Mercury, who used to be Farookh Mohammad and taught me all about name changes.