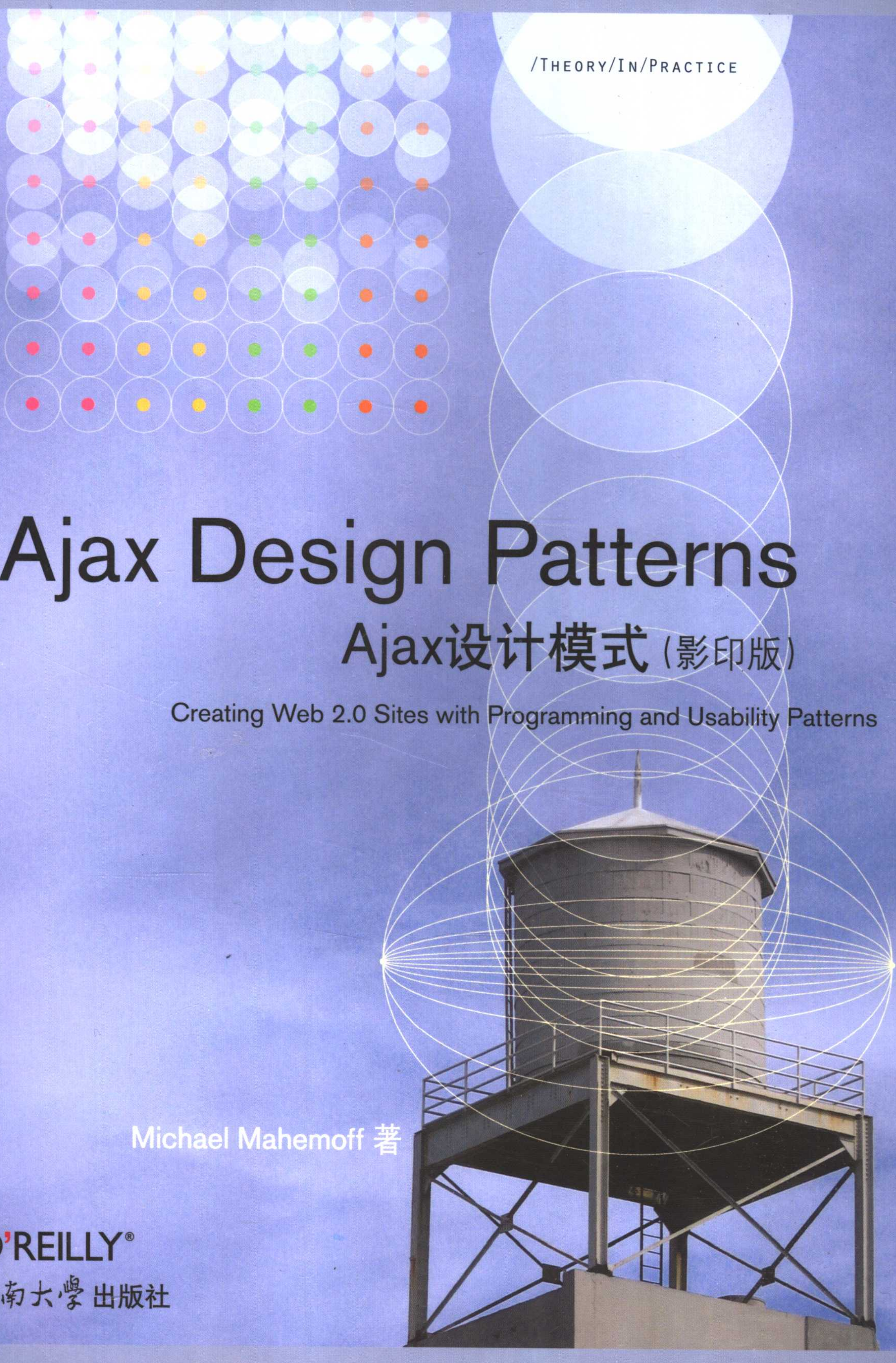
/THEORY/IN/PRACTICE



Ajax Design Patterns

Ajax设计模式（影印版）

Creating Web 2.0 Sites with Programming and Usability Patterns

Michael Mahemoff 著

O'REILLY®

東南大學 出版社

Ajax Design Patterns

Ajax设计模式（影印版）

Michael Mahemoff

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

Ajax设计模式 /（美）麦赫马夫（Mahemaff, M.）著．—影印本．—南京：东南大学出版社，2006.11
书名原文：Ajax Design Patterns
ISBN 7-5641-0571-2

Ⅰ．A… Ⅱ．麦… Ⅲ．计算机网络－程序设计－英文
Ⅳ．TP393.09

中国版本图书馆CIP数据核字（2006）第115483号

江苏省版权局著作权合同登记
图字：10-2006-260号

书　　名／ Ajax设计模式（影印版）
书　　号／ ISBN 7-5641-0571-2
责任编辑／ 张烨
封面设计／ Mike Kohnke，张健
出版发行／ 东南大学出版社（press.seu.edu.cn）
地　　址／ 南京四牌楼2号（邮政编码210096）
印　　刷／ 扬中市印刷有限公司
开　　本／ 787毫米×980毫米　16开本　41印张
版　　次／ 2006年11月第1版　2006年11月第1次印刷
印　　数／ 0001-2000册
定　　价／ 72.00元（册）

O'Reilly Media, Inc.介绍

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商 —— 每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体 —— 他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用，人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而，计算机领域的技术更新速度之快也是众所周知的，为了帮助国内技术人员在第一时间了解国外最新的技术，东南大学出版社和美国O'Reilly Meida, Inc.达成协议，将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作，以影印版或者简体中文版的形式呈献给读者。其中，影印版书籍力求与国外图书“同步”出版，并且“原汁原味”展现给读者。

我们真诚地希望，所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助，对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的一批影印版图书，包括：

- 《深入浅出 Ajax》（影印版）
- 《Ajax Hacks》（影印版）
- 《深入理解 Linux 网络内幕》（影印版）
- 《Web 设计技术手册　第三版》（影印版）
- 《软件预构艺术》（影印版）
- 《Ruby on Rails: Up and Running》（影印版）
- 《Ruby Cookbook》（影印版）
- 《Python 编程　第三版》（影印版）
- 《Python 技术手册　第二版》（影印版）
- 《Ajax 设计模式》（影印版）
- 《实用软件项目管理》（影印版）
- 《用户界面设计模式》（影印版）

PREFACE

A*JAX DESIGN PATTERNS* IS A REFERENCE FOR DEVELOPERS, DESIGNERS, AND MANAGERS WHO WANT TO know how Ajax is being used in the real world. Ajax is a new label to describe rich, desktop-like Internet applications that run in standard web browsers and do not require any special plugins. Its popularity stems from high-profile Ajax applications like Gmail, Google Maps, and 37signals' Basecamp, and it is becoming popular in the enterprise as well.

The patterns in this book fall into four categories. Foundational Technology patterns overview the raw technologies required for Ajax development, such as the `XMLHttpRequest` object. Programming patterns expose techniques developers have been discovering to ensure their Ajax applications are maintainable and performant. Functionality and Usability patterns are about the kinds of user-interfaces you'll come across in Ajax applications and the new types of functionality that Ajax makes possible. Finally, Development patterns explain the processes being used to monitor, debug, and test their Ajax applications.

Who Should Read This Book?

You should read this book if you want to:

- Learn what's possible—and what's not—with Ajax, and see how Ajax is being used in the real world.
- Get up to speed with core Ajax technologies such as XMLHttpRequest, the DOM, and JSON.
- Discover the patterns developers are using to produce high-quality Ajax architectures, streamline performance, and improve usability.

Because of the reference-like nature of the patterns, the book is accessible to people from different backgrounds. Developers who want to ramp up on Ajax will be able to begin with the tutorial chapter and the foundational technologies. Those who already have some experience with Ajax will probably gain the most from the Programming and Development patterns. People in less technical roles will be able to take a high-level perspective, looking especially at the Functionality and Usability patterns and the Real-World Examples therein to see what's possible with Ajax.

To follow the technical discussion and code examples, programmers should have some experience with the basics of web development—HTML, form submission, server-side scripting, and so on. Ideally, you should know some JavaScript too, as this book isn't intended to teach you the language, but the writing does take into account that many readers will only have basic familiarity with JavaScript.

Ajax is mostly about what happens in the browser, so the book doesn't assume you know any particular server-side environment. On those occasions where server-side code is involved, the examples are PHP-based and always explained in language-neutral terms.

Who Should Not Read This Book?

If you haven't performed any web development work, you're probably better off looking for an introduction to the basic concepts before jumping into Ajax and these patterns.

Ajax development involves working with a broad range of technologies, including HTML, XML, CSS, JavaScript, and server-side development. This book will help you understand how Ajax relates to each of these and how they are often combined together, but if you're looking to learn any one of these in depth or you're seeking a reference on browser specifics, I'd recommend consulting a specialized text.

About the Examples

All the examples in this book—the tutorial code as well as the Ajax Patterns code examples—can be downloaded from *http://ajaxify.com/run*. They've been developed to be compatible with recent versions of Firefox and Internet Explorer (IE), and have been tested on

Firefox 1.5 and IE 6.0.* Most of the server-side code requires PHP 5 or later and was tested on PHP 5.0.4. The PHP code should run on any standard web server capable of running PHP scripts; Apache 1.3.33 was used for testing. The Wiki Demo requires MySQL and was tested on version 4.1.14-max. See Appendix B for information on installing the demos.

Browser Specifics

True to the aims of Ajax, all of the Ajax Patterns are implementable in any modern, standard browser, and the discussions focus more on issues of general architecture and usability instead of browser specifics. However, portability issues do arise in some cases and are addressed where they are critical to implementing the pattern (as in some of the topics in Part II, *Foundational Technology Patterns*). To maintain a high-level focus, most of these discussion are still limited to IE and Firefox; it's beyond the scope of the patterns to outline the idiosyncracies of each browser and version, for which a specialized reference is advisable.

Organization of This Book

Part I, *Introduction*

The first few chapters are a prelude to the patterns. Chapter 1, *Introducing Ajax*, overviews the Ajax phenomenon and discusses current trends. Chapter 2, *A Pattern-Led Tutorial*, is a quick-start tutorial on Ajax and the Ajax Patterns. The design concepts behind the Ajax Patterns are discussed in Chapter 3, *Ajax Design: Principles and Patterns*, along with an introduction to the patterns themselves.

Part II, *Foundational Technology Patterns*

The Foundational Technology patterns outline the "building blocks" at the heart of any Ajax application. Ajax itself is a pattern, hence the first pattern, Ajax App (Chapter 4), which acts as the root for all the Ajax Patterns. The next few chapters look at the three core technologies involved in creating an Ajax App. Display Manipulation (Chapter 5) patterns are about repainting the user interface, and Web Remoting (Chapter 6) patterns document several alternatives for communicating with the server. The Dynamic Behavior (Chapter 7) patterns are about events—those derived by users and those based on timing. Finally, Extended Technologies (Chapter 8) explain how you can extend an Ajax App with nonstandard technologies.

Part III, *Programming Patterns*

The Programming patterns focus on technical qualities of software, in particular maintainability, robustness, and performance. The first chapter in this part, Chapter 9, provides several alternative strategies for designing Web Services. Also related to Web Remoting, the Browser-Server Dialogue (Chapter 10) patterns look at the flow of information between browser and server; e.g., synchronization techniques. The DOM Population (Chapter 11) patterns cover several strategies for handling DOM population

* It would have been nice to ensure full compatibility for all major browsers (e.g., Safari, Opera), and that's certainly advisable for production systems, but portability was considered orthogonal to the individual demos, each aiming to expose information about a particular Ajax concept.

following a server response. Code Generation and Reuse (Chapter 12) contains a couple of general programming patterns for maintainability and portability. Finally, the Performance Optimization (Chapter 13) patterns are about improving not only speed of updates and communication, but also optimizing the user experience in the face of inevitable delays.

Part IV, *Functionality and Usability Patterns*

The Functionality and Usability patterns are focused on usability of Ajax applications. Chapter 14 introduces a number of widgets that are being woven into many Ajax interfaces. A higher-level perspective is taken by the Page Architecture (Chapter 15) patterns, where the focus is on page layout, content breakdown, and techniques for exposing server-side content. With the popularity of Ajax, the Web is undergoing a rise in visual effects somewhat unprecedented on conventional desktop systems, and the most common effects are described in the Visual Effects (Chapter 16) patterns. The patterns in the final chapter of this part, Functionality (Chapter 17) patterns, are about new kinds of functionality that Ajax makes possible.

Part V, *Development Patterns*

The Development patterns are not "things" in the same sense as the patterns in previous patterns, but "processes" you can use to aid development. Diagnosis Patterns (Chapter 18) helps with troubleshooting and monitoring the health of an Ajax App. Testing Patterns (Chapter 19) helps with testing at various levels and is based on the agile practice of testing as the application evolves.

Part VI, *Appendixes*

There are four appendix sections. Appendix A, the largest appendix, is a listing of Ajax-related frameworks and libraries, many of which are a great aid in implementing some of the patterns described in this book. Appendix B is a set of installation notes for the code examples. Appendix C places the work here in the context of general design pattern theory. Finally, Appendix D lists texts referenced throughout the book.

Conventions Used in This Book

Italics

Used for names of patterns

`Constant width`

Used for code examples and fragments

`Constant width bold`

Used for important or significant lines of code

The term "the browser" is shorthand for the entire system at the browser end—not only the browser application (e.g., Firefox), but also the web application running inside it (e.g., Google Maps). When I'm talking about the browser application, I'll usually call it a "web browser"—it's fairly obvious from the context anyway. Likewise, "the server" or "the server side" refers to everything on the server—the operating system (e.g., Linux), the web server (e.g., Apache), as well as the application-specific scripts running within.

The terms "ECMAScript" and "JScript" don't appear—"JavaScript" is meant to cover these terms.

The term "class" is used to describe JavaScript functions that are used in a manner similar to classes in object-oriented systems.

The first figure that appears in each pattern's section is an illustrated overview of that particular pattern.

Some common acronyms used throughout the book are:

- CSS: Cascading Style Sheets
- DHTML: Dynamic HTML
- DOM: Document Object Model
- HTML: Hypertext Markup Language
- IE: Microsoft Internet Explorer
- XML: eXtensible Markup Language

Conventions in Code Examples

Design diagrams are based on Universal Modelling Language (UML) notation, though less formal in most cases.

In the code examples, a `.phtml` suffix is used for PHP files that output HTML; all other PHP files (such as business logic modules) end in `.php`.

As illustrated in Chapter 2, `$( )` is aliased to `document.getElementById( )` to reduce code clutter (inspired by the prototype library—see *http://prototype.conio.net/*).

Some code examples have been reformatted for the sake of clarity.

Safari® Enabled

When you see a Safari® Enabled icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *http://safari.oreilly.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

http://www.oreilly.com/catalog/ajaxdp/

There's also web page for this book, containing full draft text for all patterns and links to an increasing collection of audio podcasts about the patterns. In addition, you'll find information on various Ajax resources and errata for this book. The web page is located at:

http://ajaxpatterns.org

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

http://www.oreilly.com

Acknowledgments

Writing this book online means I received lots of excellent feedback throughout the process and have many people to thank. Which is another way of saying it's highly probable I'll leave someone out! If that's the case, please mail me any corrections.

My editor, Simon St.Laurent, has always been quick to respond to my queries and dealt admirably with a book being written in a somewhat unusual manner. Thanks also to proofreader Mary Brady for leaving no stone unturned, and illustrator Rob Romano for helping to make the content accessible at a glance. I also want to express my appreciation to O'Reilly for taking on the Ajax Patterns and especially for letting me write the book online, and blog and podcast about the content without restriction.

The book grew from a blog post on Ajax Patterns (*http://softwareas.com/ajax-patterns*), and it was the insight of Thomas Baekdal to cover Ajax usability principles that inspired to the initial patterns post (*http://www.baekdal.com/articles/Usability/XMLHttpRequest-guidelines*). I also have Brent Ashley to thank for his discussing the patterns at the initial Ajax summit and in his blog, apparently the trigger for the initial ripple of online interest in the project.

The ideas in this book also owe a great deal to Jesse-James Garrett and his seminal Ajax article, without which you would not be reading these words.

All the book reviewers provided valuable feedback on the online version throughout the writing process as well as offering many detailed comments on the book's draft version. The reviewers were: Cameron Shorter (ADI Limited, Mapbuilder) and Kevin Vinsen (ADI Limited), whose employer, ADI Limited, provided time for them both to conduct the review; Jep Castelein (Backbase), Daniel Czarnecki (Zoltak Consulting), Tony Hill (Thomson Corporation), and Alex Kirk (Blummy.com, WizLite.com). In addition, a big thanks to members of the Software Architecture Group (SAG) at the University of Illinois at Urbana-Champaign for conducting several rich discussions on the patterns and making the audio publicly available (*http://www.softwareas.com/sag-ajax-patterns-review-1*)—the comments certainly made a difference as the editing moved into its final stages. Ralph Johnson, who leads the group, nominated the Ajax Patterns for review, and Brian Foote kept me informed throughout the process. I'm also grateful to everyone who added to the public portions of the AjaxPatterns.org wiki and offered feedback on the online draft via email, blog comments, and in wiki discussions.

As the patterns were discovered from many existing applications, I would like to thank the creators of all the examples featured in the pattern descriptions, many of them true pioneers. Special thanks to those developers who explained aspects of their designs to me: Kevin Arthur (Stream), Richard Cowin and Bill Scott (OpenRico), Donovan Preston (LivePage), Jeremy Ruston (TiddlyWiki), and Tino "Crisp" Zijdel (DHTML Lemmings). The patterns are also based on numerous writings, code dissections, ideas, proofs-of-concept, and direct suggestions. In particular, a number of patterns are only here because of the contributions of the following authors and developers (also mentioned in individual pattern descriptions): Julien Couvreur, James Dam, Abe Fettig (Twisted, Jotspot), Chris Justus, Christopher Kruslicky, Alex Russell (Dojo, Jotspot), Richard Schwartz, Mike Stenhouse, Joel Webber, and Chris Were. In addition, there were several news-focused services that made it a lot easier to locate all this content as it happened (and also helped other people find AjaxPatterns.org!). Among these resources: Niner Niner's AjaxBlog.com, Chris Cornutt's AjaxDeveloper.org (Chris is now with Ajaxian.com), Dion Almaer, Ben Galbraith, and Rob Sanheim at Ajaxian.com (disclaimer: I've since joined the Ajaxian team), Shane Witbeck's AjaxMatters.com, Mike Papageorge's FiftyFourEleven.com, Brian Benzinger's SolutionWatch.com, and Mike Arrington's Techcrunch.com.

Last but not least, heaps of gratitude to my family for their support throughout the writing process.

ABOUT THE AUTHOR

Michael Mahemoff has been working with software for 22 years, including 10 years of commercial experience. At the University of Melbourne, he earned degrees in software engineering and psychology, and earned a Ph.D with a thesis on "Design Reuse in Software Engineering and Human-Computer Interaction." He lives in London and consults on software development issues in banking and healthcare. Michael's blog and online projects are linked from his homepage at *http://mahemoff.com*.

COLOPHON

The cover image is from *Cassell's Natural History*. The cover fonts are Akzidenz Grotesk and Orator. The text font is Adobe's Meridien; the heading font is ITC Bailey.

CONTENTS