Introduction

How many times have you seen a web site and said, “This would be exactly what I wanted—if only . . .” If only you could combine the statistics here with data from your company’s earnings projections. If only you could take the addresses for those restaurants and plot them on one map. How often have you entered the date of a concert into your calendar with a single click instead of retyping? How often do you wish that you could make all the different parts of your digital world—your e-mail, your word processor documents, your photos, your search results, your maps, your presentations—work together more seamlessly? After all, it’s all digital and malleable information—shouldn’t it all just fit together?

In fact, below the surface, all the data, web sites, and applications you use could fit together. This book teaches you how to forge those latent connections—to make the Web your own—by remixing information to create your own mashups. A *mashup*, in the words of the Wikipedia, is a web site or web application “that seamlessly combines content from more than one source into an integrated experience.”[1] Learning how to draw content from the Web together into new integrated interfaces and applications, whether for yourself or for other others, is the central concern of this book.

Let’s look at a few examples to see how people are remixing data and services to make something new and useful:

- Housingmaps.com brings together Google Maps and the housing and rental listings from Craigslist.com. Note that it was invented by neither Google nor Craigslist but by an individual programmer, Paul Radamacher. Housingmaps.com adds to the functionality of Craigslist, which will show you on a map where a specific listing is located but not all the rentals or houses in an area.[2]
- Google Maps in Flickr (GMiF) brings together Flickr pictures, Google Maps, Google Earth, and the Firefox browser via Greasemonkey.[3]
- The Library Lookup bookmark is a JavaScript bookmarklet that connects Amazon.com and your local library catalog.[4]

[1] http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid), accessed as http://en.wikipedia.org/w/index.php?title=Mashup_%28web_application_hybrid%29&oldid=98002063

[2] http://housingmaps.com/

[3] http://webdev.yuan.cc/gmif/

[4] http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookup.html

To create your own mashups and customize the Web, you will look at these examples in greater detail, in addition to many other examples large and small, in this book. You can solve countless specific problems by remixing information. Here are some examples of techniques you will learn in this book:

- Taking a book you found on Amazon.com and instantly locating it in your local library

- Synthesizing a single news feed from many news sources through Yahoo! Pipes
- Posting Flickr photos to blogs with a click of a button
- Displaying your photos in Google Maps and Google Earth
- Using special extensions to Firefox to learn how to program Google Maps
- Inserting extra information into web pages with Greasemonkey
- Plotting stories from your favorite news source (such as the *New York Times*) on a map
- Making your own web site remixable so that others can create mashups from your content
- Creating Google calendars from your event calendars from the Web
- Storing and retrieving your files from online storage (S3)
- Creating an online spreadsheet from your Amazon.com wishlist
- Recognizing and manipulating data embedded in web pages
- Adding an event listed on the Web to your calendar and e-mailing it to other people with one mouse click
- Building web search functionality into your own web applications
- Republishing word documents that are custom-formatted for your web site

Mashups are certainly hot right now, which is interesting because it makes you part of a shared undertaking, a movement. Mashups are fun and often educational. There's delight in seeing familiar things brought together to create something new that is greater than the sum of its parts. Some mashups don't necessarily ask to be taken that seriously. And yet mashups are also powerful—you can get a lot of functionality without a lot of effort. They might not be built to last forever, but you often can get what you need from them without having to invest more effort than you want to in the first place.

The Web 2.0 Movement

The Web 2.0 bandwagon is an important reason why mashups are popular now. Mashups have been identified explicitly (under the phrases "remixable data source" and "the right to remix") by Tim O'Reilly in "What is Web 2.0?"[5] Added to this, we have the development of what might be accurately thought of as "Web 2.0 technologies/mind-sets" to remix/reuse data, web services, and micro-applications to create hybrid applications. Recent developments bring us closer to enabling users to recombine digital content and services:

- Increasing availability of XML data sources and data formats in business, personal, and consumer applications (including office suites)
- Wide deployment of XML web services
- Widespread current interest in data remixing or mashups
- Ajax and the availability of JavaScript-based widgets and micro-applications

* Evolution of web browsers to enable greater extensibility (for example, Firefox extensions and Greasemonkey scripts)

* Explosive growth in “user-generated content” or “lead-user innovation”

* Wider conceptualization of the Internet as a platform (“Web 2.0”)

* Increased broadband access

[5] http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

These developments have transformed creating mashups from being technically challenging to nearly mainstream. It is not that difficult to get going, but you need to know a bit about a fair number of things, and you need to be playful and somewhat adventurous.

Will mashups remain cutting-edge forever? Undoubtedly, no, but not because they will prove to be an irrelevant fad but because the functionality we see in mashups will eventually be subsumed into the ordinary “what-we-expect-and-think-has-always-been-there” functionality of our electronic society.

Moreover, mashups reflect deeper trends, even the deepest trends of human desire. As the quality, quantity, and diversity of information grow, users long for tools to access and manage this bewildering array of information. Many users will ultimately be satisfied by nothing less than an information environment that gives them seamless access to any digital content source, handles any content type, and applies any software service to this content. Consider, for example, what a collection of bloggers expressed as their desires for next-generation blogging tools:[6]

> *Bloggers want tools that are utterly simple and allow them to blog everything that they can think, in any format, from any tool, from anywhere. Text is just the beginning: Bloggers want to branch out to multiple media types including rich and intelligent use of audio, photos, and video. With input, having a dialog box is also seen as just a starting place for some bloggers: everything from a visual tool to easy capture of things a blogger sees, hears, or reads point to desirable future user interfaces for new generations of blogging tools.*

Mashups are starting to forge this sought-after access and integration of data and tools—not only in the context of blogging but also to any point of interaction between users and content.

[6] http://www.cadence90.com/blogs/2004_03_01_nixon_archives.html#107902918872392913

Overall Flow of the Book

A central question of this book is, how can both nontechnical end users and developers recombine data and Internet services to create something new for their own use for and for others? Although this book focuses primarily on XML, web services, and the wide variety of web applications, I’ll also cover the role played by desktop applications and operating systems.

The Book’s Structure

The following is a breakdown of the parts and chapters in this book:

- **Part 1**, “Remixing Information Without Programming,” introduces mashups without demanding programming skills from you and teaches skills for deconstructing applications for their remix potential.
  - Chapter 1, “Learning from Specific Mashups,” analyzes in detail a selection of mashups/remixes (specifically, Housingmaps.com, Google Maps in Flickr, and the LibraryLookup bookmarklet) to get you oriented to mashups in general and to some general themes we will continually revisit throughout the book.
  - Chapter 2, “Uncovering the Mashup Potential of Web Sites,” analyzes Flickr (as our primary extended example) for what makes it the remix platform *par excellence* for learning how to remix a specific application and exploit features that make it so remixable. We compare and contrast Flickr with other remixable platforms such as del.icio.us, Google Maps, and Amazon.com.
  - Chapter 3, “Understanding Tagging and Folksonomies,” covers tagging. Tagging, which allows users to attach words to pictures, and websites—almost anything on the Web—is the glue that holds many things together, both within and across websites. This chapter illustrates how tags are used in Flickr, del.icio.us, and Technorati and discusses how to create interesting tag-centric mashups, how people are “hacking” the tagging system to create ad hoc databases, and how tags relate to other classification systems.
  - Chapter 4, “Working with Feeds, RSS, and Atom,” presents RSS and Atom, perhaps the most widespread dialects of XML, as both a potent technology for remixing in its own right and also as a specific way to learn about XML more generally. Not to be missed are the sections on the various RSS/Atom-related formats and their significance for information remix. The chapter includes a tutorial on using Yahoo! Pipes to filter and synthesize feeds.
  - Chapter 5, “Integrating with Blogs,” uses Flickr’s integration with weblogs as a jumping-off point for an exploration of weblogs and wikis and their programmability. Integration with blogging is an important topic since blogs represent a type of remixing in a narrative, as opposed to data-oriented remixing via tags and the straight RSS so far discussed. A brief discussion of integration with wikis concludes the chapter.
- **Part 2**, “Remixing a Single Web Application Using Its API,” concentrates on teaching the broad classes of web-based APIs by studying exemplars of each class.
  - Chapter 6, “Learning Web Services APIs Through Flickr,” studies Flickr in detail. In addition to be an exemplar for a range of nonprogramming remixing techniques in Part 1, Flickr is also an excellent playground for learning XML web services. This chapter will show you how to use the Flickr API, looking first at how to make a simple call to the API, next

looking at how to make sense of the entire variety of calls available, and then generalizing to handle authentication.

  * Chapter 7, “Exploring Other Web APIs,” explains commonalities and contrasts among various API providers, specifically those between Flickr and other systems, and surveys the types of services available and how to think about the sheer range of APIs. You will learn how to call REST, XML-RPC, and SOAP-based services. This chapter looks at sites, such as Programmableweb.com, that document these various APIs and the challenges faced in doing so.

  * Chapter 8, “Learning Ajax/JavaScript Widgets and Their APIs,” describes the other large class of web application remixability: those of JavaScript-based widgets, many of which are Ajax applications. This chapter contrasts old-style web applications with Ajax approaches through specific examples in Flickr and other applications and introduces the Yahoo! UI Library, a specific JavaScript widget library to demonstrate how to program widgets. You will also learn how to use the Firebug Firefox extension and the JavaScript Shell to learn about JavaScript. The chapter concludes with an introduction to using Greasemonkey.

* **Part 3**, “Making Mashups,” is the heart of the book; it’s a discussion of how to use what you learned in Parts 1 and 2 to create mashups.

  * Chapter 9, “Moving from APIs and Remixable Elements to Mashups,” analyzes mashups and their relationship to APIs through studying a series of specific problems for which mashups can provide useful solutions. The chapter looks at how you can track books, real estate, airfare, and current events by combining various APIs. You will learn how to use Programmableweb.com to analyze these problems.

  * Chapter 10, “Creating Mashups of Several Services,” teaches you how to write mashups by providing a detailed example that you’ll build from the ground up: a mashup of geotagged Flickr photos and Google Maps using first the Google Maps API and then the Google Mapplets API.

  * Chapter 11, “Using Tools to Create Mashups,” discusses tools that have been developed to make creating mashups easier than by using traditional web programming techniques. This chapter walks you through using one of these tools—the Google Mashup Editor—and briefly surveys other tools.

  * Chapter 12, “Making Your Web Site Mashable,” shifts the focus of the book briefly from the consumption to the production of data and APIs. This chapter is a guide to content producers who want to make their web sites friendly to mashups. That is, this chapter answers the question, how would you as a content producer make your digital content most effectively remixable and mashable to users and developers?

* **Part 4**, “Exploring Other Mashup Topics,” covers how to remix and integrate specific classes of applications, using the core conceptual framework of Parts 1 to 3 to guide the discussion.

* Chapter 13, “Remixing Online Maps and 3D Digital Globes,” covers popular online maps and virtual globes, offering examples of map-based mashups. You’ll learn about making maps without programming and data exchange formats (GeoRSS and KML), and then you’ll turn to the various APIs: Google Maps, Yahoo! Maps, and Microsoft Maps. I’ll also cover geocoding American and non-American addresses. The chapter closes with a discussion of Google Earth, its relationship to KML, and how to display Flickr photos via KML.

* Chapter 14, “Exploring Social Bookmarking and Bibliographic Systems,” covers how social bookmarking responds to a fundamental challenge—the job of keeping found things found on the Web, which, at a basic level, is done through URLs, but you’ll learn about other digital content such as images and data sets. Social bookmarking is interesting not only for the extensibility/remixability being built into these systems but also for the insight it offers into other systems. This chapter walks you through a select set of social bookmarking systems and their APIs, as well as discusses interoperability challenges among these systems. The chapter shows how to create a mashup of Flickr and del.icio.us.

* Chapter 15, “Accessing Online Calendars and Event Aggregators,” shows what data you can get in and out of calendars without programming (using iCalendar and XML feeds), how to program individual calendars (using Google Calendar and 30boxes), and how to program individual event aggregator APIs (using Upcoming.yahoo.com and Eventful.com). The chapter concludes with a mashup of a public events calendar with Google Calendar.

* Chapter 16, “Using Online Storage Services,” surveys the potentially important and growing area of online storage solutions and shows the basics of using Amazon S3.

* Chapter 17, “Mashing Up Desktop and Web-Based Office Suites,” shows how to do some simple parsing in ODF and OpenXML, demonstrates how to create a simple document in both ODF and OpenXML, explains some simple scripting of Microsoft Office and OO.o, and concludes with a mashup of Google Spreadsheets and Amazon.com web services.

* Chapter 18, “Using Microformats and RDFa As Embeddable Data Formats,” studies two answers to the problem of how to embed information in web pages that is easy to understand by both humans and computer programs: microformats and RDFa. You will learn how to use and program the Operator Firefox extension to recognize and manipulate microformats.

* Chapter 19, “Integrating Search,” shows how to use the Google Ajax Search API, Yahoo! Search APIs, and Microsoft Live.com search; the chapter also introduces OpenSearch and the Google Desktop HTTP/XML gateway.

Intended Audience

This book is accessible to a wide range of readers, including those who are curious about Web 2.0 applications and those who want to know more about the technical underpinnings of it. The technical perquisites are a good understanding of HTML, basic CSS, and basic JavaScript. References to appropriate background materials will be provided. In this book, most of the server-side code is presented in PHP. Some code is in Python.

At the same time, experienced developers will also be able to learn much from the book. Although there will be a breadth of coverage, I will strive to state deep, essential facts about the technologies in question (with respect to their applicability to remix)—aspects that might not be obvious at first glance.

Information remixing can easily come across as a confusing grab bag of techniques. Beginners have a hard time understanding the significance of XML, web services, Ajax, COM, and metadata for remixing data. It is not that difficult to get going, but you need to know a bit about a fair number of different topics, and you need to be playful and somewhat adventurous. Usually these topics are found scattered throughout a large selection of books; this book is the guide to show you where to begin.

Updates

Please go to `http://mashupguide.net` to find updates and supplementary materials for this book.