RICH INTERNET APPLICATIONS

Chapter 2: Ajax Basics

What is AJAX?

- Online brochures and catalogs no longer dominated the Internet as web applications began to emerge as a significant portion of online destinations.
- Web applications differed from their web site ancestors in that they provided an instant service to their users, not just information.
- Whether for business process management or personal interests, developers were forced to create new interaction paradigms as users came to expect richer functionality.
- Spurred on by little-known and lesser-used technologies that had been included in web browsers for some time, the Web took a bold step forward, shattering the traditional usage model that required a full page load every time new data or a new part of the application's logic was accessed.

- Companies began to experiment with dynamic reloading of portions of web pages, transmitting only a small amount of data to the client, resulting in a faster, and arguably better, user experience.
- New experiments conducted by Google engineers began popping up through a special part of the site called Google Labs (labs.google.com).
- Many of the projects at Google Labs, such as Google Suggest and Google Maps, involved only a single web page that was never unloaded but was constantly updated nevertheless.
- These innovations were praised around the Web.

- Numerous open source and commercial products began development to take advantage of this new web application model.
- These projects explained their technology using a variety of terms such as JavaScript remoting, web remote procedure calls, and dynamic updating.

Formal Definition of Ajax:

- AJAX = Asynchronous JavaScript and XML.
- AJAX is a technique for creating fast and dynamic web pages.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.
- Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook tabs.

How Ajax Works?

- When user first visits the page, the Ajax engine is initialized and loaded. From that point of time user interacts with Ajax engine to interact with the web server. The Ajax engine operates asynchronously while sending the request to the server and receiving the response from server.

  Ajax life cycle within the web browser can be divided into following stages:
- **User Visit to the page:** User visits the URL by typing URL in browser or clicking a link from some other page.

- **Initialization of Ajax engine:**
  When the page is initially loaded, the Ajax engine is also initialized. The Ajax engine can also be set to continuously refresh the page content without refreshing the whole page.

How Ajax Works? (Contd…)

- **Event Processing Loop:**
  * Browser event may instruct the Ajax engine to send request to server and receive the response data
  * Server response - Ajax engine receives the response from the server. Then it calls the JavaScript call back functions
  * Browser (View) update - JavaScript request call back functions is used to update the browser. DHTML and css is used to update the browser display.

How Ajax Works? (Contd…)

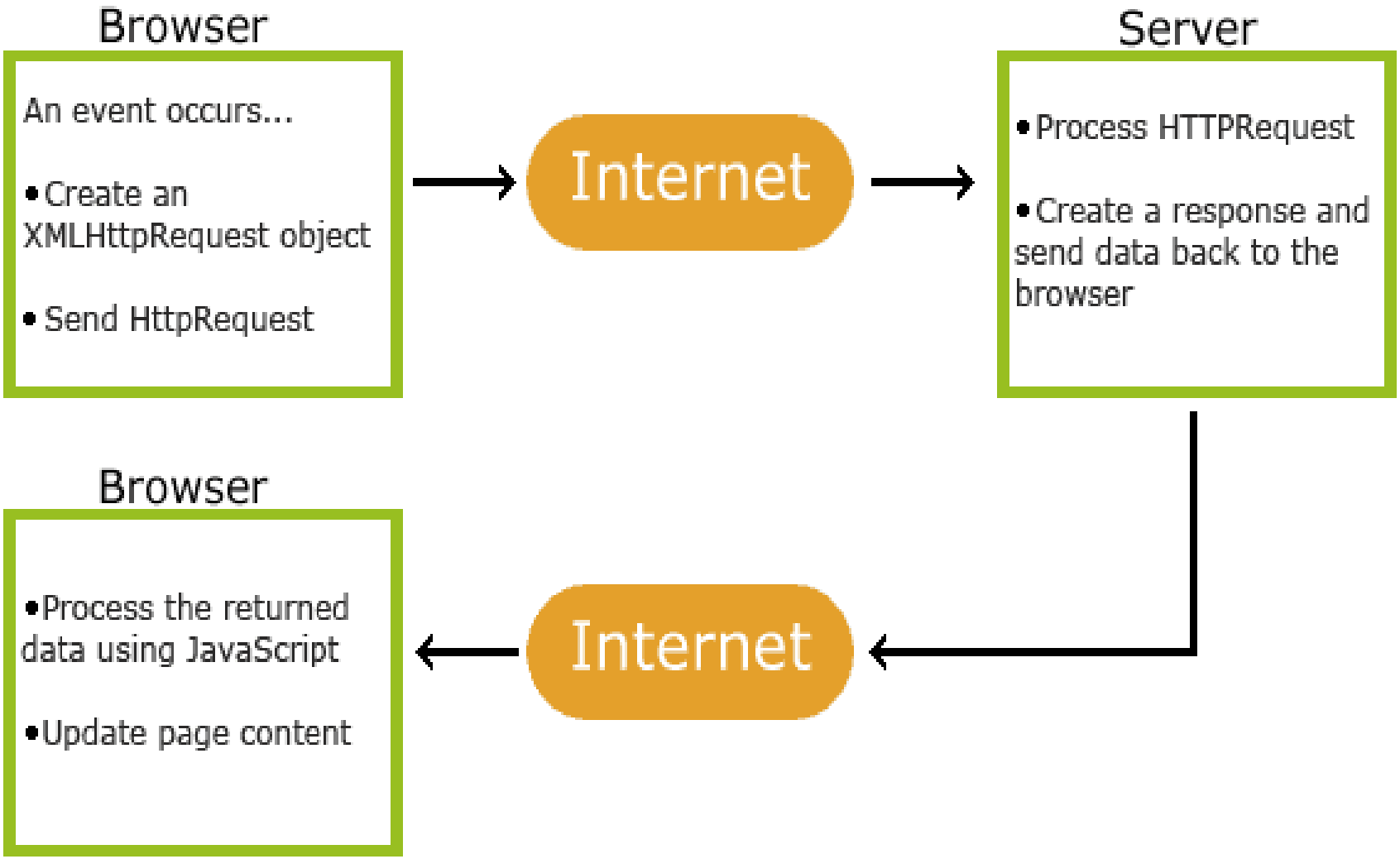

Figure 1.1 : How Ajax Works

AJAX is Based on Internet Standards

- AJAX is based on internet standards, and uses a combination of:
- XMLHttpRequest object (to exchange data asynchronously with a server)
- JavaScript/DOM (to display/interact with the information)
- CSS (to style the data)
- XML (often used as the format for transferring data)
- AJAX applications are browser- and platform-independent!

Google Suggest

- AJAX was made popular in 2005 by Google, with Google Suggest.
- Google Suggest is using AJAX to create a very dynamic web interface: When you start typing in Google's search box, a JavaScript sends the letters off to a server and the server returns a list of suggestions.

Ajax is Born

- In February 2005, Jesse James Garrett of Adaptive Path, LLC published an online article entitled, “Ajax: A New Approach to Web Applications” (still available at www.adaptivepath.com/publications/essays/archives/ 000385.php).
- In this essay, Garrett explained how he believed web applications were closing the gap between the Web and traditional desktop applications.
- He cited new technologies and several of the Google projects as examples of how traditionally desktop-based user interaction models were now being used on the Web.

Ajax is Born(Contd…)

- Then came the two sentences that would ignite a firestorm of interest, excitement, and controversy:

  Google Suggest and Google Maps are two examples of a new approach to web applications that we at Adaptive Path have been calling Ajax. The name is shorthand for Asynchronous JavaScript + XML, and it represents a fundamental shift in what's possible on the Web.

The Evolution of the Web

- When Tim Berners-Lee crafted the first proposal for the World Wide Web in 1990, the idea was fairly simple: to create a “web” of interconnected information using hypertext and Uniform Resource Identifiers (URIs).
- The ability to link disparate documents from all around the world held huge potential for scholarly endeavors, where people would be able to access referenced material almost instantly.
- Ultimately, it was from the static web pages that the Web grew.

The Evolution of the Web(Contd…)

JavaScript:

- Netscape Navigator was the first successful mainstream web browser, and as such, moved web technologies along quickly.
- Originally named LiveScript, JavaScript was created by Brendan Eich of Netscape and included in version 2.0 of the browser (released in 1995). For the first time, developers were able to affect how a web page could interact with the user.
- Instead of making constant trips to the server and back for simple tasks such as data validation, it became possible to transfer this small bit of processing to the browser.
- This ability was very important at a time when most Internet users were connected through a 28.8 Kbps modem, turning every request to the server into a waiting game.
- Minimizing the number of times that the user had to wait for a response was the first major step toward the Ajax approach.

The Evolution of the Web(Contd…)

Frames:

- Netscape Navigator 2.0 was the first browser to support frames and JavaScript together.
- This turned out to be a major step in the evolution of Ajax.

The Hidden Frame Technique:

- As developers began to understand how to manipulate frames, a new technique emerged to facilitate client-server communication.
- The hidden frame technique involved setting up a frameset where one frame was set to a width or height of 0 pixels, its sole purpose being to initiate communication with the Server.
- The hidden frame technique represented the first **asynchronous request/response** model for web applications.

The Evolution of the Web(Contd…)

Dynamic HTML & the DOM:

- DHTML enabled developers to alter any part of a loaded page by using JavaScript.
- **Combining DHTML** with the hidden frame technique meant that any part of a page could be **refreshed** with server information at any time.
- DHTML never made it to a standards body but, the introduction of DOM had a great influence.
- Unlike DHTML, which sought only to modify sections of a web page, the DOM had a more ambitious purpose:

  "To provide a structure for an entire web page."
- The manipulation of that structure would then allow DHTML-like modifications to the page.
- Introduction of DOM also led to the growth of Ajax.

The Evolution of the Web(Contd…)

Iframes :

- Although the hidden frame technique became incredibly popular, it had a downside—one had to plan ahead of time and write a frameset anticipating the usage of hidden frames.
- When the <iframe/> element was introduced as an official part HTML 4.0 in 1997, it represented another significant step in the evolution of the Web.
- Instead of defining framesets, developers could place iframes anywhere on a page.
- This enabled developers to forego framesets altogether and simply place invisible iframes (through the use of CSS) on a page to enable client-server communication.
- And when the DOM was finally implemented in Internet Explorer 5 and Netscape 6, it introduced the ability to dynamically create iframes on the fly, meaning that a JavaScript function could be used to create an iframe, make a request, and get the response—all without including any additional HTML in a page.
- This led to the next generation of the hidden frame technique: the hidden iframe technique.

The Evolution of the Web(Contd…)

XMLHttp:

- The browser developers at Microsoft must have realized the popularity of the hidden frame technique and the newer hidden iframe technique, because they decided to provide developers with a better tool for client-server interaction.
- That tool came in the form of an ActiveX object called XMLHttp, introduced in 2001.

Benefits of Ajax

Ajax is new very promising technology, which has become extremely popular these days.

Here are the benefits of using Ajax:

1. Ajax can be used for creating rich, web-based applications that look and works like a desktop application.

2. Ajax is easy to learn. Ajax is based on JavaScript and existing technologies like XML, CSS, DHTML. etc. So, its very easy to learn Ajax.

3. Ajax can be used to develop web applications that can update the page data continuously without refreshing the whole page.

The Real Ajax

- Ajax is nothing more than an approach to **web interaction**.
- **This approach involves transmitting only a small amount of information to and from the server in order to give the user the most responsive experience possible.**
- In traditional web application model, browser itself is responsible for **initiating requests** to, and **processing requests** from, the web server.
- The Ajax model provides an intermediate layer— what Garrett calls an ***Ajax engine****—to handle this communication.*
- *An **Ajax engine** is really just a **JavaScript** **object or function*** that is called whenever information needs to be requested from the server.
- The request is done **asynchronously**, meaning that code execution doesn't wait for a response before continuing.

The Real Ajax (Contd…)

- The server—which traditionally would serve up HTML, images, CSS, or JavaScript—is configured to return data that the **Ajax engine** can use.
- This data can be:
    i. plain text,
    ii. XML, or
    iii. any other data format
- The only requirement is that the Ajax engine can understand and interpret the data
- Refer Figure 1-1

The Real Ajax (Contd…)

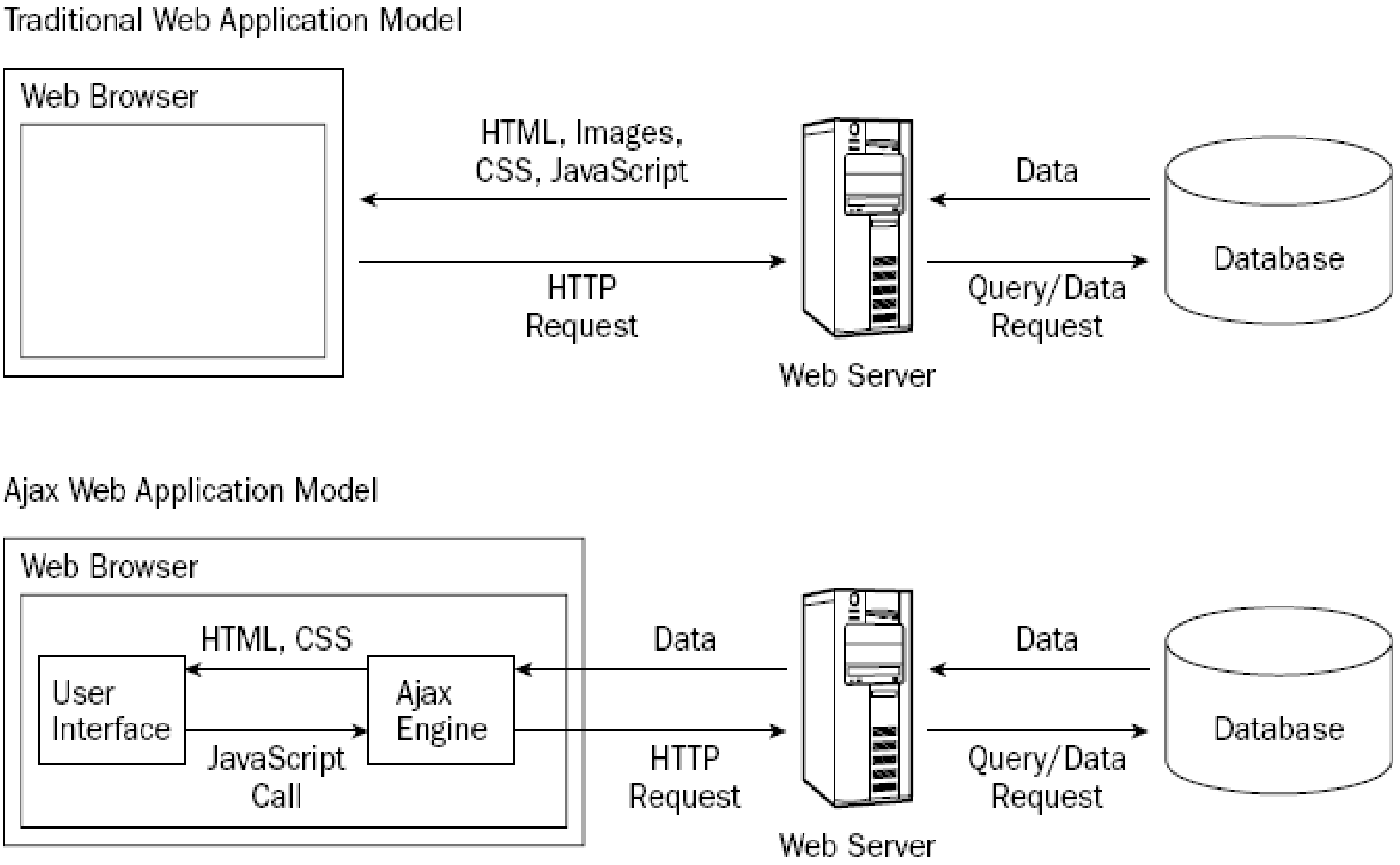

Figure 1-1

Ajax Principles

- Michael Mahemoff (www.mahemoff.com), a software developer and usability expert, identified several key principles of good Ajax applications that are worth repeating:

❑ **Minimal traffic:**

- Ajax applications should send and receive as little information as possible to and from the server.
- In short, Ajax can minimize the amount of traffic between the client and the server. Making sure that your Ajax application doesn't send and receive unnecessary information adds to its robustness.

❑ **No surprises:**

- Ajax applications typically introduce **different user interaction models** than traditional web applications.
- Some Ajax applications use other user interface paradigms such as **drag-and-drop or double-clicking**.
- No matter what user interaction model you choose, be consistent so that the user knows what to do next.

Ajax Principles(Contd…)

- **Established conventions:**

- Don’t waste time inventing new user interaction models that your users will be unfamiliar with.
- **Borrow heavily from traditional web applications** and **desktop applications**, so there is a minimal learning curve.

- **No distractions:**

- Avoid unnecessary and distracting page elements such as **looping animations and blinking page sections**.
- Such gimmicks distract the user from what he or she is trying to accomplish.

- **Accessibility:**

- Consider who your **primary and secondary users** will be and how they most likely will access your Ajax application.
- Don’t program yourself into a corner so that an unexpected new audience will be completely locked out.
- Will your users be using older browsers or special software? Make sure you know ahead of time and plan for it.

Ajax Principles(Contd…)

- **Avoid entire page downloads:**

- All server communication after the initial page download should be managed by the Ajax engine.

- **User first:**

- Design the Ajax application with the **users in mind** before anything else.
- Try to make the common use cases easy to accomplish and don't be caught up with how you're going to fit in advertising or cool effects.

Understanding the technology behind Ajax

- Ajax is not a single technology, but it is a combination of many technologies. These technologies are supported by modern web browsers. Following are techniques used in the Ajax applications.
- **JavaScript:**
  JavaScript is used to make a request to the web server. Once the response is returned by the web server, more JavaScript can be used to update the current page. **DHTML and CSS** is used to show the output to the user. **JavaScript** is used very heavily to provide the dynamic behavior to the application.

- **Asynchronous Call to the Server:**
  Most of the Ajax application used the **XMLHttpRequest object** to send the request to the web server. These calls are Asynchronous and there is no need to wait for the response to come back. User can do the normal work without any problem.

- **XML:**
  XML may be used to receive the data returned from the web server. JavaScript can be used to process the XML data returned from the web server easily.

Technologies behind Ajax

- Garrett's article mentions several technologies that he sees as parts of an Ajax solution. These are:

- **HTML/XHTML:** Primary content representation languages.
- **CSS:** Provides stylistic formatting to XHTML.
- **DOM:** Dynamic updating of a loaded page.
- **XML:** Data exchange format.
- **XSLT:** Transforms XML into XHTML (styled by CSS).
- **XMLHttp:** Primary communication broker.
- **JavaScript:** Scripting language used to program an Ajax engine.

Who is Using Ajax?

Some of the more well-known & well executed web applications that use Ajax are:

Google Suggest:

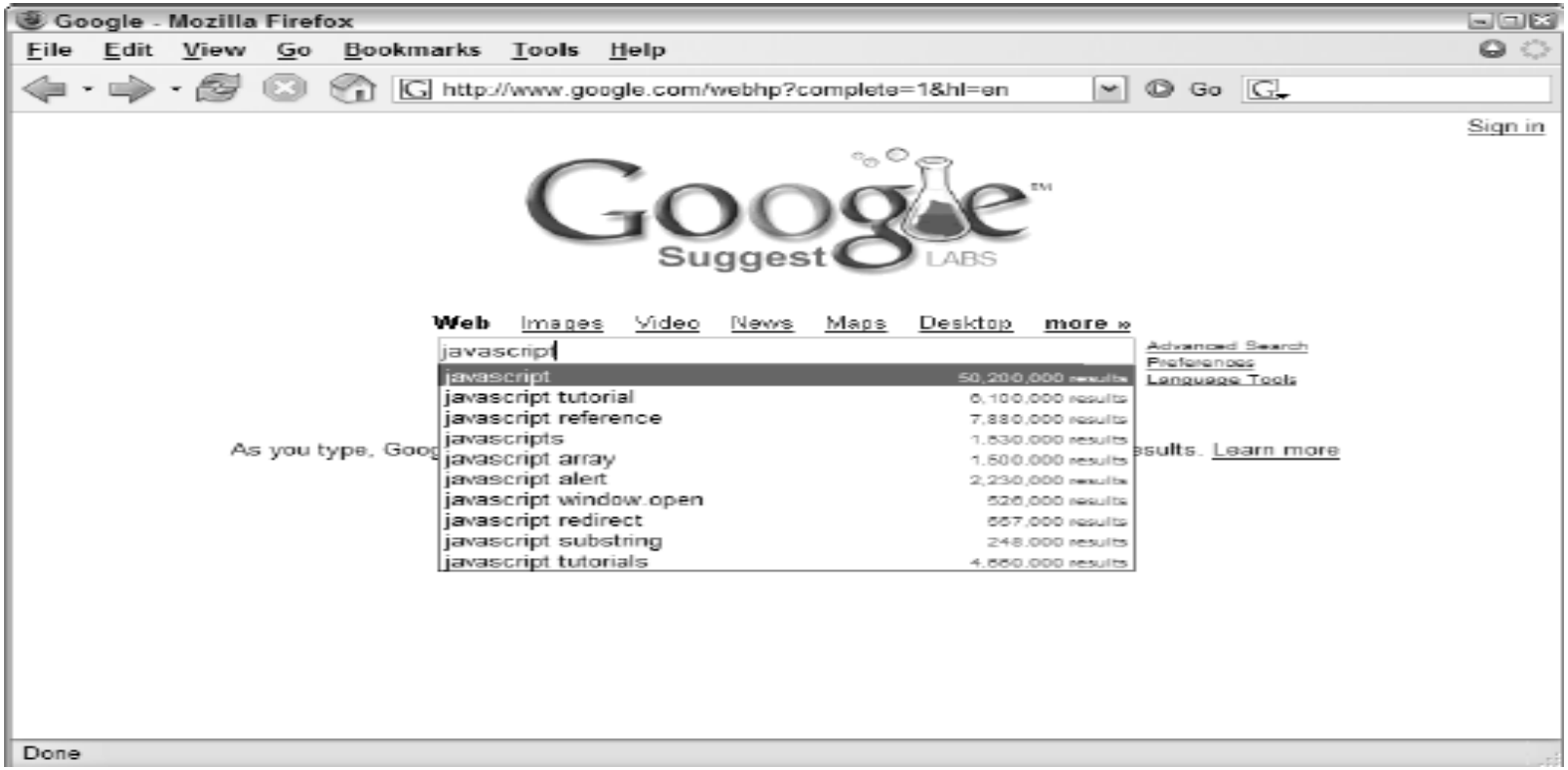

Figure 1.2 : User Interface of Google Suggest

Google Suggest(Contd…)

- 1st example of Ajax is Google Suggest. Everything appears to be the same until you start typing in the textbox. As you type, Google Suggest requests suggestions from the server, showing you a drop-down list of search terms that you may be interested in.
- Each suggestion is displayed with a number of results available for the given term to help you decide.
- This simple client-server interaction is very powerful and effective without being obtrusive to the user.
- The interface is responsive beyond what you may have learned to expect from a web application; it updates no matter how quickly you type and, as with autocomplete features in desktop software, you can use the up and down arrows to highlight and select each item in the suggestions list.
- Although still in beta, expect to see this approach make its way into the main Google page eventually.

Gmail

- Gmail, Google's free e-mail service, has been raved about as a marvel of client-server interaction in the age of Ajax.
- When you first log in to Gmail, a user interface engine is loaded into one of the few iframes the application uses.
- All further requests back to the server occur through this user interface engine through an XMLHttp object.
- The data being transferred back and forth is JavaScript code, which makes for fast execution once downloaded by the browser.

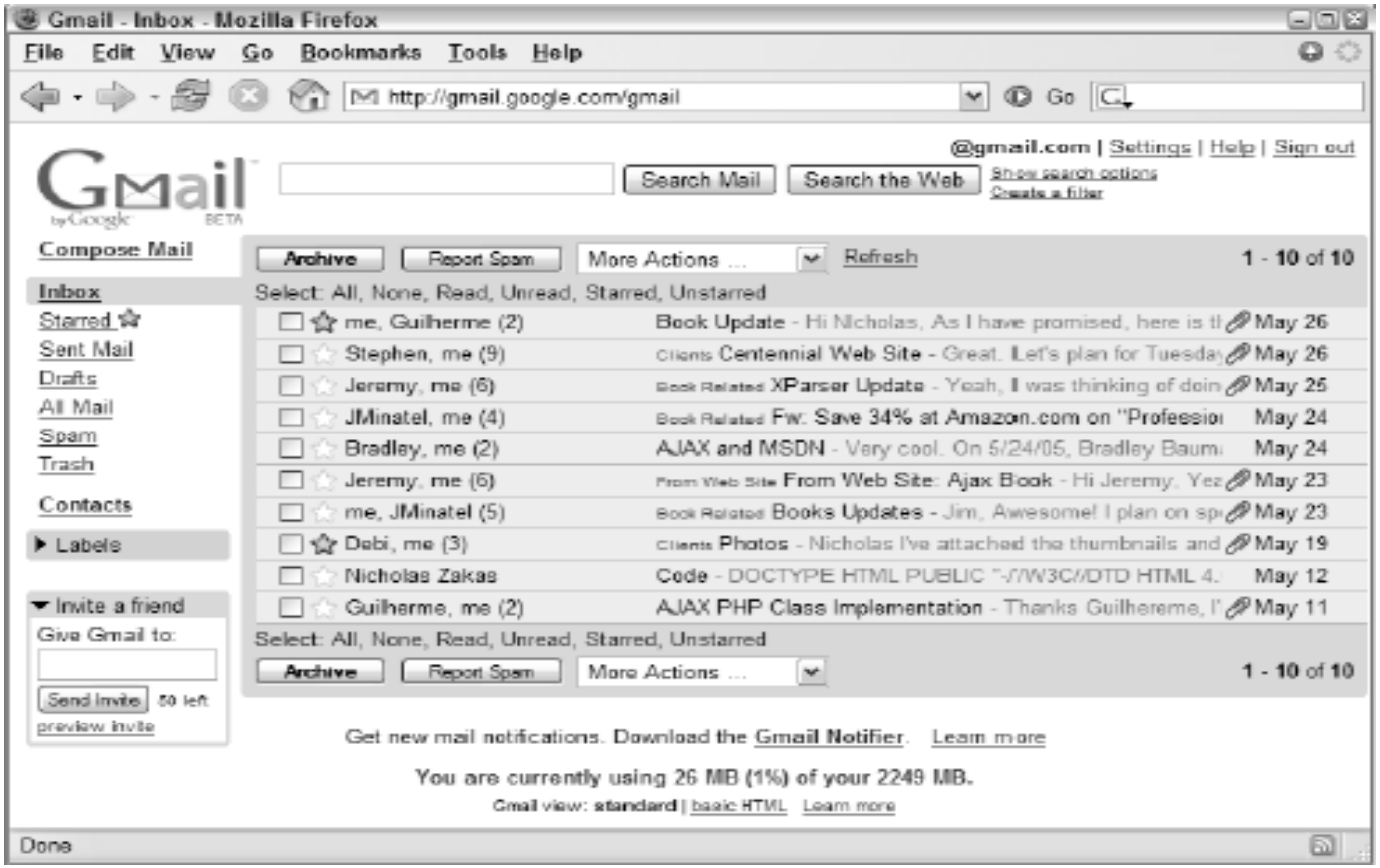

Figure 1.3 : User Interface of Gmail Account

- Additionally, the Gmail application uses several **frames and iframes** to manage and cache big user interface changes.
- The extremely complicated use of frames enables Gmail to function properly with the Back and Forward buttons, which is one of the advantages of using **frames or iframes** instead of or in conjunction with XMLHttp.

Google Maps

- Google Maps uses Ajax to avoid reloading its main page at all.
- Google Maps enables you to drag the map to move it in various directions.
- The dragging code is nothing new to JavaScript developers, but the tiling of the map and seemingly endless scrolling effect are another story.
- The map is broken up into a series of images that are tiled together to make the appearance of a contiguous image.
- The number of images used to display the map is finite, as creating new images every time the user moves the map would quickly lead to memory problems

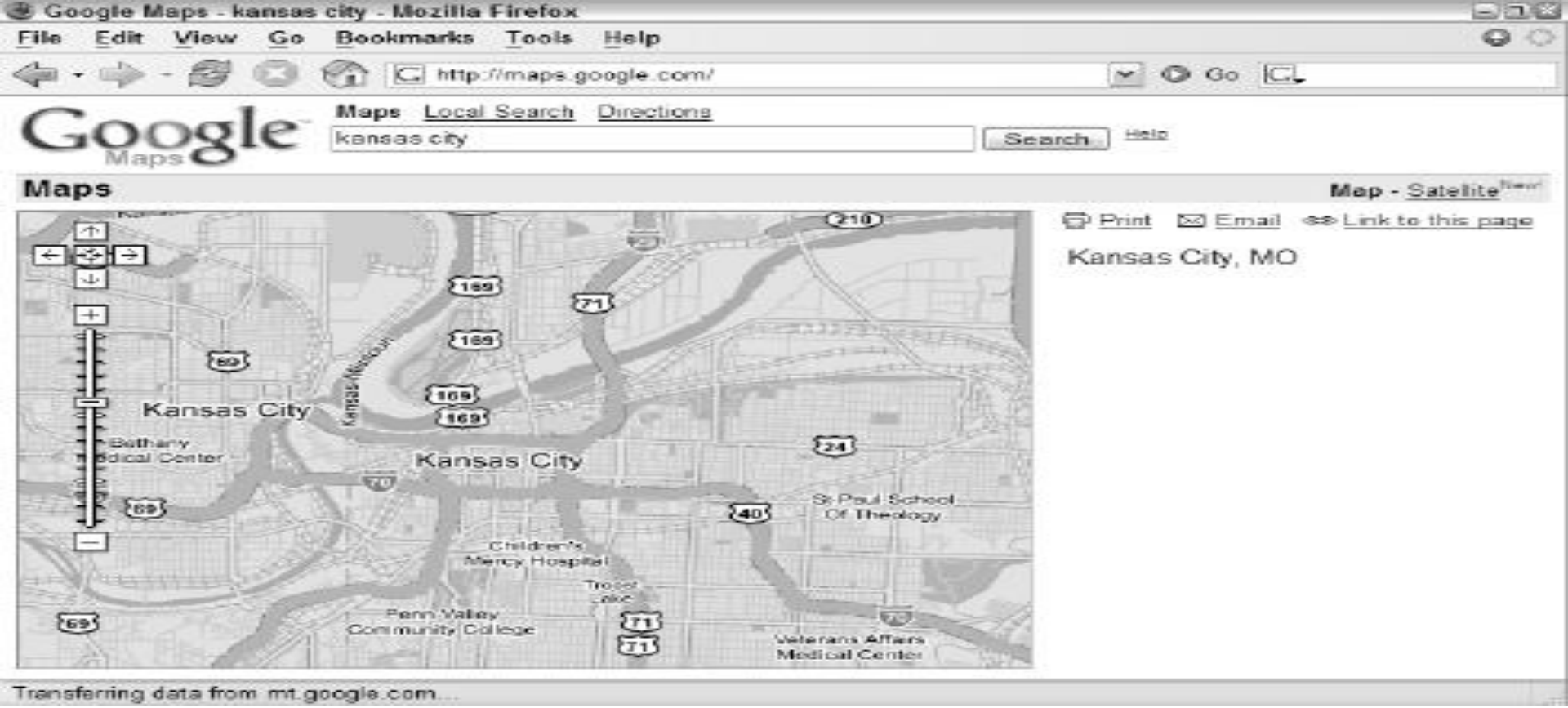

Figure 1.4 : User Interface of Google Maps

- The client-server communication is done through a hidden iframe.
- Whenever you do a search or ask for new directions, this information is submitted and returned within that iframe.
- The data returned is in XML format and is passed to a JavaScript function (the Ajax engine) to handle.

A9

- **Amazon.com** is world famous for being an online marketplace for just about anything, but when it released a search engine, it did so with little fanfare and attention.
- The introduction of A9 (www.a9.com) showed off enhanced searching, enabling you to search different types of information simultaneously.
- What makes A9 unique is how its user interface works.
- **A9.com** is a subsidiary of Amazon.com based in Palo Alto, California that develops search engine technology.
- A9 currently has over 100 employees in its Palo Alto, Bangalore, and Dublin offices.
- When you perform a search, the different types of results are displayed in different areas of the page (see Figure 1-5).

Figure 1.5 : User Interface of A9.com

Yahoo! News

- The new design features an interesting enhancement: when you move your mouse over a particular headline, a small box pops up with a summary and, optionally, a photo associated with that story (see Figure 1-6).

Figure 1.6 : User Interface of Yahoo! News

Bitflux Blog

- Another great example of using Ajax only as an enhancement is Bitflux Blog (blog.bitflux.ch), which features a technology called LiveSearch.
- LiveSearch works in conjunction with the search box on the site.
- As you type into the box, a list of possible search results is displayed immediately below (see Figure 1-7).

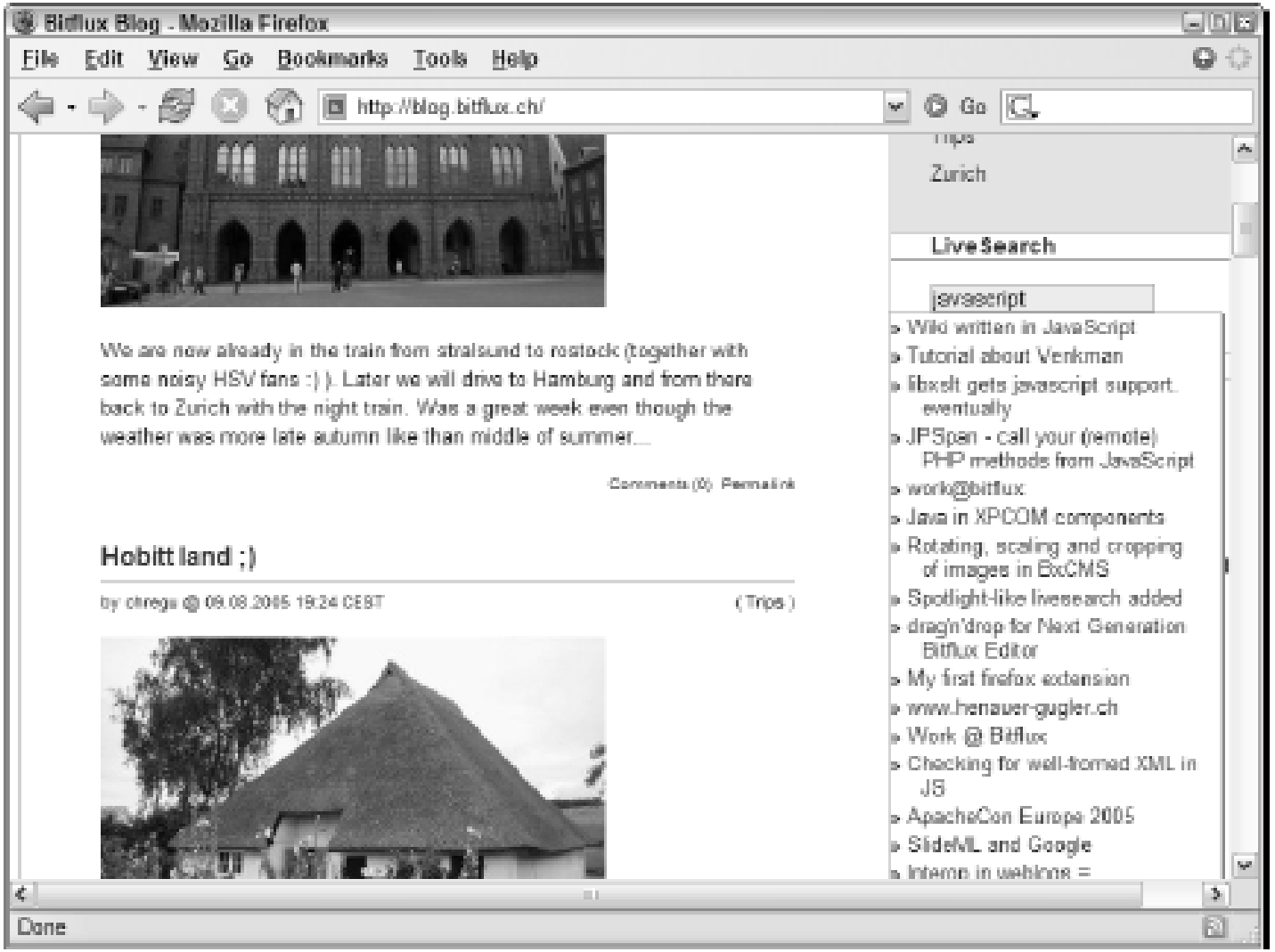

Figure 1.7 : User Interface of Bitflux Blog

Summary

- Short for Asynchronous JavaScript + XML, the term *Ajax was coined by* ***Jesse James Garrett*** *in an article posted on the Adaptive Path, LLC web site.*
- The article introduced Ajax as a new user interaction model for web applications in which full page loads are no longer necessary.
- This chapter also explored the **evolution of the Web in relation to the development of technologies that enable Ajax to be a reality today**.
- Throughout the evolution of new web technologies, Ajax methodologies such as the hidden frame technique developed.
- The introduction **of iframes and XMLHttp** really pushed Ajax development forward.
- Several of the most popular Ajax applications were also discussed, including **Google Suggest, Gmail, Google Maps, Yahoo! News, and the Bitflux Blog.**

THE END