Part I
Content Concepts

1

Enhancing the Web

Although most of this book can be seen as an attempt to navigate through a landscape of potential and opportunity for a future World Wide Web, it is prudent, as in any navigational exercise, to start by determining one's present location. To this end, the first chapter is a descriptive walkabout in the current technology of the Web – its concepts and protocols. It sets out first principles relevant to the following exploration, and it explains the terms encountered.

In addition, a brief Web history is provided, embedded in the technical descriptions. Much more than we think, current and future technology is designed and implemented in ways that critically depend on the technology that came before. A successor technology is usually a reaction, a complement, or an extension to previous technology – rarely a simple plug-in replacement out of the blue. New technologies invariably carry a legacy, sometimes inheriting features and conceptual aspects that are less appropriate in the new setting.

Technically savvy readers may recognize much material in this chapter, but I suspect many will still learn some surprising things about how the Web works. It is a measure of the success of Web technology that the average user does not need to know much of anything technical to surf the Web. Most of the technical detail is well-hidden behind the graphical user interfaces – it is essentially click-and-go. It is also a measure of success that fundamental enhancements (that is, to the basic Web protocol, not features that rely on proprietary plug-in components) have already been widely deployed in ways that are essentially transparent to the user, at least if the client software is regularly updated.

Chapter 1 at a Glance

Chapter 1 is an overview chapter designed to give a background in broad strokes on Web technology in general, and on the main issues that lead to the formulation of the Semantic Web. A clear explanation of relevant terms and concepts prepare the reader for the more technical material in the rest of the book.

There and Back Again sets the theme by suggesting that the chapter is a walkabout in the technology fields relevant to the later discussions that chapter by chapter revisit the main concepts, but in far greater detail.

The Semantic Web: Crafting Infrastructure for Agency Bo Leuf

- *Resource Identifiers* defines fundamental identity concepts, protocol basics, and how content can at all be located in the current Web by the user's client software.
- *Extending Web Functionality* examines proposed ways to enhance the basic Web transport protocol, as well as protocol-independent methods.
- *From Flat Hyperlink Model* describes the current navigational model of the Web, especially the hyperlink, and highlights the areas where it is lacking. After a wishlist of Web functionality, *To Richer Informational Structures* explores strategies for extending the hyperlink model with background information about the content.
- *The Collaboration Aspect* explores one of the driving forces for a new Web, after which *Extending the Content Model* shows why a unified way to handle content is important in any extension.
- *Mapping the Infosphere* discusses ways that have been tried to map what is on the Web so that users can find what they are looking for. *Well-Defined Semantic Models* introduces why current lexical mappings are insufficient for the task, especially if searching and processing is to be automated.

There and Back Again

The World Wide Web was conceived and designed as an open information space defined by the *hyperlink* mechanism that linked documents together. The technology enabled anyone to link to any other document from hyperlinks on a published Web page – a page anyone could see, and link to in turn. The whole system could self-grow and self-organize.

No permissions were required to set up such links; people just linked to whatever other published resources they found interesting and useful. The only requirements to participate were a simple Internet connection and a place to put a Web page. This open nature is fundamental to many of the early design decisions and protocol implementations, sometimes in ways that were later obscured or marginalized.

Bit 1.1 The Web is an open universe of network-accessible information

This definition of the Web, formulated by Tim Berners-Lee, provides in all its simplicity the most fundamental description of the Web's potential.

Open network access enables a potentially infinite resource, for people both to contribute to and use. The explosive growth of the Web and the content it mediates is in many ways a direct consequence of this design. It has given rise to a remarkable plethora of both content and functionality, sometimes unexpected.

> *I am very happy at the incredible richness of material on the Web, and in the diversity of ways in which it is being used. There are many parts of the original dream which are not yet implemented. For example, very few people have an easy, intuitive tool for putting their thoughts into hypertext. And many of the reasons for, and meaning of, links on the web is lost. But these can and I think will change.*
>
> Tim Berners-Lee (*www.w3.org/People/Berners-Lee/FAQ.html*),
> 'inventor' of the Web and director of the W3C.

In addition, the original design had the goal that it should not only be useful for human-to-human communication but also support rich human–machine and machine–machine interactions. In other words, the intent was that machines would be able to participate fully and help in the access and manipulation of this information space – as automated agents, for example.

Bit 1.2 The Web had the twin goals of interactive interoperability and creating an evolvable technology

The core values in Web design are expressed in the principle of universality of access – irrespective of hardware or software platform, network infrastructure, language, culture, geographical location, or physical or mental impairment.

Before going further into the nature of such interactions and the infrastructure that is to support them, we need to explore the fundamental issues of resource identity, and how naming schemes relate to the protocols used to access the resources.

Resource Identifiers

The full interoperability and open-ended nature of the Web was intended to be independent of language, as evident in the way the design specified the universality of referencing resources by identity through the **Universal Resource Identifier** (or **URI**).

The principle that absolutely anything 'on the Web' can be identified distinctly and uniquely by abstract pointers is central to the intended universality. It allows things written in one language to refer to things defined in another language.

Properties of naming and addressing schemes are thus defined separately, associated through the **dereferencing protocol**, allowing many forms of identity, persistence and equivalence to refer to well-defined resources on the Web. When the URI architecture is defined and at least one dereferencing protocol implemented, the minimum requirement for an interoperable global hypertext system is just a common format for the content of a resource (or Web object).

Anyone can create a URI to designate a particular Web resource (or anything, actually). This flexibility is at the same time both the system's greatest strength and a potential problem. Any URI is just an identifier (a 'name' often opaque to humans), so simple inspection of it in isolation does not allow one to determine with certainty exactly what it means. In fact, two different URIs might refer to the same resource – something we often also run across in naming schemes in the 'everyday' world.

The concept of unique identifiers finds expression in many fields, and is crucial to 'finding' and handling things in a useful way.

Any identifier scheme assuredly defines a useful **namespace**, but not all schemes provide any useful dereferencing protocol. Some examples from the latter category are the **MIME** content identifier (*cid*) or message identifier (*mid*) spaces, the MD5 **hash code** with verifiable pure identity (often used as secure verification of file identity), and the pseudo-random Universally Unique Identifier (*uuid*). They all identify but cannot locate.

The ability to utilize such namespace schemes in the URI context provides valuable functionality—as is partially illustrated by some peer-to-peer (p2p) networks, such as *Freenet* (described in the previous book, *Peer to Peer*).

Even a simple persistent identity concept for connection-oriented technologies, for which no other addressable content exists, can prove more useful than might be expected. Witness the ubiquitous mailbox namespace defined by the '*mailto:*' protocol – unfortunately named, however, since URIs are functionally nouns not verbs. The resulting URIs define connection endpoints in what is arguably the most well-known public URI space, persistent virtual locations for stores of e-mail messages.

Understanding HTTP Space

The other most well-known public URI space is the **HTTP** namespace – commonly called 'the Web'. It is characterized by a flexible notion of identity and supports a richness of information about resources and relating resources.

HTTP was originally designed as a protocol for remote operations on objects, while making the exact physical location of these objects transparent. It has a dereferencing algorithm, currently defined by HTTP 1.1, but augmented by caching, proxying and mirroring schemes. Dereferencing may therefore in practice take place even without HTTP being invoked directly.

The HTTP space consists of two parts:

- **Domain Name**, a hierarchically delegated component, for which the Domain Name System (DNS) is used. This component is a centrally registered top level domain (**TLD**): generic (**gTLD**, such as *example.com*) or national (**ccTLD**, *example.se*).
- **Relative Locator**, an opaque string whose significance is locally defined by the authority owning the domain name. This is often but need not (indeed should rarely) be a representation of a local directory tree path (relative to some arbitrary 'root' directory) and a file name (example: */some/location/resource*).

A given HTTP URI (or resource object identity) is commonly written as a **URL**, a single string representing both identity and a Web location. URL notation concatenates the parts and prefixes the protocol (as *http://*). As a rule, client software transparently maps any 'illegal' characters in the URL into protocol-acceptable representations, and may make reasonable assumptions to complete the abbreviated URL entry.

In practice, the domain component is augmented by other locally defined (and optional) prefix components. Although still formally DNS arbitrated, the prefix is determined by the local authority. It is resolved by the DNS and Web (or other service) servers in concert. Most common is '*www.*' but it may also be a server name, a so-called vanity domain, or any other local extension to the domain addressing scheme.

A related important feature is to dereference the physically assigned IP number bound to a particular machine. Domain names therefore improve **URI persistence**, for example when resources might move to other machines, or access providers reallocate. However, persistence of locators in HTTP space is in practice not realized fully on the Web.

Bit 1.3 URLs are less persistent overall than might reasonably be expected

Improving persistence involves issues of tool maturity, user education, and maturity of the Web community. At a minimum, administrators must be discouraged from restructuring (or 'moving') resources in ways that needlessly change Web addresses.

The design of HTTP and the DNS makes addressing more human-readable and enables almost complete decentralization of resources. Governance is freely delegated to local authorities, or to endpoint server machines. Implementing a hierarchical rather than flat namespace for hosts thus minimizes the cost of name allocation and management.

Only the domain-name component requires any form of formal centralization and hierarchical structure – or rather, only as currently implemented does it require centralized registrars and domain-name databases for each TLD.

The domain name is, strictly speaking, optional in HTTP. It is possible, if not always convenient (especially with the trend to share IP in virtual hosting), to specify HTTP resource addresses using the physical IP number locally assigned by an access provider. Other protocol spaces, such as *Freenet*, in fact dispense with domains altogether and rely instead on unique key identifiers and node searches.

As a feature common and mandatory to the entire HTTP Web, as it is currently used, the **DNS root** is a critical resource whose impartial and fair administration is essential for the world as a whole. Ownership and governance of the root of the DNS tree and *gTLD* subtree databases has in recent years been subject to considerable debate. The situation is only partially and nominally resolved under international arbitration by **ICANN**.

The Semantics of Domain Names

Another issue concerning *gTLD* allocation, with relevance to the concept of 'meaning' (or semantics) of the URI, is the design intent that the different domain categories say something about the owners.

The original international *gTLD*s and their intended application were clear, for example:

- '.com' for commercial organizations with a true international presence (those with presence just in a single national region were recommended to use instead country-code domains).
- '.org' for non-profit organizations with an international presence.
- '.net' for international providers of network services, for example Web hosts or service providers.

Such a division provides basic meta-information embedded in the URL in a way that is easy to see. However, so many U.S. companies registered under *.com* that the common perception became that *.com* designated a U.S. business. This skew was due mainly to the original strict rules concerning the allocation of *.us* domains by state, rather than allowing a company with a multi-state presence to use a single national domain.

The *.com* domains became in fact the most popular on the Web, a status symbol no matter what the purpose. It also gave rise to the pervasive '*dotcom*' moniker for any Internet-related

business venture. In a similar vein, though less so, the popular but mistaken association arose that *.org* and *.net* were U.S. domains as well. Administration by a U.S. central registrar only strengthened this false association.

- These three *gTLD*s are considered *open domains*, because there are no formal restrictions on who may register names within them. Anyone, anywhere, can therefore have a *dotcom*, *dotorg*, or *dotnet* domain – business, individual, whatever.

Further confusion in the role of these *gTLD*s arose when domain names became traded commodities. Public perception tends to equate brand name with domain name, regardless of the TLD. Attractive names thus became a limited commodity and the effective namespace smaller. Most brand owners are likely to register all three at once.

Bit 1.4 The TLD level of domain names currently lacks consistent application

The TLD system defines the root of the URLs that, as subsets of URIs, are supposed to provide an unambiguous and stable basis for resource mapping by resource owners. Reality is a bit more complex, also due to hidden agendas by the involved registrars.

Therefore, over time, the original semantic division was substantially diluted, and essentially it is today lost from public awareness. However, the intent of at least one of the original seven *gTLD* was preserved:

- '.int' is used only for registering organizations established by international treaties between governments, or for Internet infrastructure databases.

Since the early Internet was mostly implemented in the U.S. and the administration of the *gTLD* names was then under U.S. governance, the other three original categories were quickly reserved for U.S. bodies:

- '.edu' for universities and corresponding institutions of higher education that qualified, became in practice the domain for U.S.-based institutions.
- '.gov' became reserved exclusively for the United States Government and its federal institutions.
- '.mil' became reserved exclusively for the United States Military.

Other countries must use qualified *ccTLD*s for the same purpose, such as *.gov.uk.* A further special TLD, '.arpa' is provided for technical infrastructure purposes.

In an apparent attempt to reclaim some inherent TLD meaning, new usage-restricted *gTLD*s were proposed in 2000. First implemented were *.biz*, *.info*, *.pro*, and *.name*, while *.coop*, *.aero*, and *.museum* are still pending (see FAQ at *www.internic.net/faqs/new-tlds.html*). Since 2004, ICANN has added *.asia*, *.cat*, *jobs*, *.mail*, *.mobi*, *.post*, *.tel*, *.travel*, and *.xxx* to the proposal list.

Unfortunately, we already see semantic erosion in some of the newly implemented TLDs. Also, the addition of more TLD namespaces not only risks further confusion in public

perception (*.biz* or *.com*?), but can be detrimental to the original concept of reducing the cost of namespace management. Brand owners may feel compelled redundantly to add further TLDs to protect against perceived misappropriation.

Bit 1.5 Conflicting interests are fragmenting the Web at the TLD-level

Issues of ownership and intent involve complex and changing policy and politics, not easily pinned down in any lasting way, and often at odds with the Web's underlying concept of universality. The resulting confusion, market pressures, and market conflict subtly distort the way we can usefully map the Web.

In 2004, the W3C was, in fact, moved to state that the influx of new TLD subtrees was harmful to the Web infrastructure, or at the very least incurred considerable cost for little benefit. Detailed critique is given for some of these 'special-interest' proposals (see reasoning at *www.w3.org/DesignIssues/TLD* and *www.w3.org/2004/03/28-tld*), for example:

- Implementing *.mobi* would seem to partition the HTTP information space into parts designed for access from mobile devices and parts not so designed. Such a scheme destroys the essential Web property of device independence.
- The domain name is perhaps the worst possible way of communicating information about the device. A reasonable requirement for adaptive handling of mobile devices is that it be transparent, by way of stylesheets and content negotiation.
- Content-filtering through TLD (as in *.xxx*) is unlikely to be effective, even assuming best-case consensus, applicability, and binding international agreements on appropriate domain owners and suitable content in the respective TLD. We may also assume that many companies would merely redirect new special-interest domains (such as *.travel*) to existing ones (an established *.com*, for instance).

As it happens, even the superficially clear geographic relationship of the *ccTLDs* to the respective countries has been considerably diluted in recent years. The increased popularity of arbitrary businesses, organizations, or individuals registering attractive small-country domains as alternatives to the traditional *dotcom* ones causes more TLD confusion.

In part, this practice reflects a preference for country codes that impart some 'useful' association, which can become popular in the intended contexts – for example, Tongan '*.to*' as in '*http://come.to*', or Samoan '*.ws*' recast as meaning 'website' or 'worldsite'. In part, it is a result of the increasing scarcity of desired or relevant *dotcom* names. This 'outsourced' registration, usually U.S.-based through some licensing agreement, generates significant foreign income (in U.S. dollars) for many small Pacific island nations, but assuredly it confuses the previously clear knowledge of *ccTLD* ownership.

Pervasive Centralization

Despite the decentralized properties inherent to the HTTP space design, the subsequent evolution of the Web went for a time in a different direction.

A needless fragmentation situation arose, with different protocols used to transfer essentially the same kind of text (or message) objects in different contexts – for example, e-mail, Web page content, newsgroup postings, chat and instant messaging. Such fragmentation, which leads to multiple client implementations, is confusing to the user. The confusion is only recently in part resolved by advanced multiprotocol clients.

Web implementations also ignored to a great extent the interactive and interoperative design goals – much to the disappointment of the early visionaries who had hoped for something more open and flexible. This deficiency in the Web is part of what prompted the development of other clients in other protocols, to implement functionality that might otherwise have been a natural part of the HTTP Web.

The pervasive paradigm of the Web instead became one of centralized content-providing sites designed to serve unilaterally a mass of content-consuming clients. Such sites constrain user interaction to just following the provided navigational links.

Web pages thus became increasingly 'designer' imprinted, stylistic exercises 'enhanced' with attention-grabbing devices. Technology advances unfortunately tended to focus on this eyeball functionality alone. In fact, the main visitor metric, which tellingly was used to motivate Web advertising, became 'page hits' or 'eyeball click-through counts' rather than any meaningful interaction.

Most Web-browser 'improvements' since the original *Mosaic* client (on which *MS Internet Explorer*, the long dominant browser, is based) have thus dealt more with presentational features than any real navigational or user-interaction aspects. Worse, much of this development tended towards proprietary rather than open standards, needlessly limiting the reach of information formatted using these features.

Revival of Core Concepts

Despite the lackluster Web client implementations with respect to user functionality, the potential for interactive management of information on the Web remained an open possibility – and a realm in recent years extended by alternative peer architectures.

The way the Internet as a whole functions means that nothing stops anyone from deploying arbitrary new functionality on top of the existing infrastructure. It happens all the time. In fact, Internet evolution is usually a matter of some new open technology being independently adopted by such a broad user base that it becomes a *de facto* new standard – it becomes ever more widely supported, attracting even more users.

Bit 1.6 Internet design philosophy is founded in consensus and independent efforts contributing to a cohesive whole

Principles such as simplicity and modularity, basic to good software engineering, are paralleled by decentralization and tolerance – the life and breath of Internet.

Several open technologies have in recent years provided a form of revival in the field, with a focus on content collaboration. One such technology, explored in an earlier book, *The Wiki Way*, is a simple technology that has in only a few years transformed large parts of the visible Web and redefined user interaction with Web-published content.

- Wiki technology relies on the stock Web browser and server combination to make collaborative co-authoring a largely transparent process. It leverages the existing client-server architecture into a more peer-based collaboration between users.
- Prominent examples of large-scale deployment are *Wikipedia* (*www.wikipedia.com*) and related *Wikimedia* projects, and many open-source support sites (*sourceforge.net*).

Other extending solutions tend to be more complex, or be dependent on special extensions to the basic Web protocol and client-server software (such as plug-in components, or special clients). Yet they too have their validity in particular contexts.

This tension between peer and hierarchical models was further explored and analyzed in considerable detail in the book *Peer to Peer*, along with the concept of **agency**. Although these peer technologies lie outside the immediate scope of the current book, some aspects drawn from these discussions are taken up in relevant contexts in later chapters.

Extending Web Functionality

With the enormous growth of the Web and its content since its inception, it is clear that new implementations must build on existing content to gain widespread usage, or at least allow a relatively simple and transparent retrofit. This development can be more or less easy, depending on the intentions and at what level the change comes.

So far, we have seen such changes mainly at the application level. Long sought is a change at a more profound level, such as a major extension to the underlying Web protocol, HTTP. Such a change could harmonize many functionality extensions back into a common and uniform framework on which future applications can build.

Current HTTP does in fact combine the basic transport protocol with formats for limited varieties of **metadata** – information about the payload of information. However, because it is descended from the world of e-mail transport (and an old protocol), HTTP metadata support as currently implemented remains a crude architectural feature that should be replaced with something better.

Bit 1.7 The Web needs a clearer distinction between basic HTTP functionality and the richer world of metadata functionality

A more formalized extension of HTTP, more rigorous in its definitions, can provide such a needed distinction and thus bring the Semantic Web closer to reality.

Extending HTTP

HTTP was designed as part of a larger effort by a relatively small group of people within the IETF HTTP Working Group, but Henrik Frystyk Nielsen (the specification author) claims that this group did not actually control the protocol. HTTP was instead considered a ‘common good’ technology, openly available to all.

In this vein of freedom, HTTP was extended locally as well as globally in ways that few could predict. Current extension efforts span an enormous range of applications, including distributed authoring, collaboration, printing, and remote procedure call mechanisms.

However, the lack of a standard framework for defining extensions and separating concerns means that protocol extensions have not been coordinated. Extensions are often applied in an *ad hoc* manner which promotes neither reusability nor interoperability.

For example, in the variant **HTTPS** space, a protocol distinction is made needlessly visible in the URI. Although HTTPS merely implies the use of HTTP through an encrypted *Secure Socket Layer* (**SSL**) tunnel, users are forced to treat secure and insecure forms of the same document as completely separate Web objects. These instances should properly be seen as transparent negotiation cases in the process of dereferencing a single URI

Therefore, the *HTTP Extension Framework* was devised as a simple yet powerful mechanism for extending HTTP. It describes which extensions are introduced, information about who the recipient is, and how the recipient should deal with them.

The framework allows parameters to be added to method headers in a way that makes them visible to the HTTP protocol handler (unlike **CGI** parameters, for example, that must remain opaque until dealt with by the handler script on the target server). A specification of the HTTP Extension Framework is found in RFC 2774. (Among other sources, see user-friendly Web site *www.freesoft.org/CIE/RFC/r* to search and read **RFC** documents.)

Otherwise, the most ubiquitous and transparent functionality change in the Web in recent years was an incremental step in the basic Web protocol from HTTP 1.0 to HTTP 1.1. Few users noticed this upgrade directly, but a number of added benefits quickly became mainstream as new versions of server and client software adapted.

- Examples of new features include *virtual hosting* (to conserve the IP-number space), *form-based upload* (for browser management of Web sites), and *MIME extensions* (for better multimedia support).

Many would instead like to see something more like a step to 'HTTP-NG' (Next Generation) that could implement a whole new range of interoperable and interactive features within the base protocol.

- *WebDAV* is one such initiative, styled as completing the original intent of the Web as an open collaborative environment, and it is discussed in Chapter 4.

Consider, after all, how *little* HTTP 1.x actually gives in the form of 'exposed interfaces', otherwise often termed 'methods' and known by their simple names. These methods are in effect the 'action verbs' (one of many possible such sets) applied to the 'identifier nouns' (URI) of the protocol.

The main method is to access resources:

- **GET** is the core request for information in the Web. It takes the form of HTTP header specifying the desired information object as either the unique location (URL) or the process (CGI) to produce it. (A variant, HEAD, might support the return of header information but without data body.)

GET is actually the only HTTP method that is required always to be supported. It has a special status in HTTP, in that it implements the necessary dereferencing operation on identifiers – it defines the namespace. As such, GET must never be used in contexts that have

side-effects. Conversely, no other method should be used to perform only URI dereferencing, which would violate universality by defining a new namespace.

Many *ad hoc* extensions and p2p-application protocols are based solely on GET.

The GET-queried resource is expected to return the requested information, a redirection URI, or possibly a status or error message. It should never change state. In the request response, an appropriate *representation* of the URI-specified object is transferred to the client – *not*, as is commonly assumed, the stored literal data.

Representations, encoding, and languages acceptable may be specified in the GET-header request fields, along with any specific client-side information. These and other factors affect both what is returned and the chosen format or encoding.

Client-side input may be handled by two other methods:

- **PUT** is the method to store information at a particular Web location specified by a valid URL. It is structurally similar to GET, except that the header is also associated with a body containing the data to be stored. Data in the body comprise opaque bitstreams to HTTP agents.
- **POST** is in effect an *indirect* method to pass an information object to a Web server. It is entirely up to the server to determine both when and how to deal with it (process, store, defer, ignore, or whatever). The header URI specifies the receiving server agent process (such as a named script), possibly including a suggested URI for the object.

Other methods sometimes seen are essentially only extensions of these basic three. It is optional for the responding HTTP agent (the Web server) to implement an appropriate process for any of these input methods.

The following are some formally registered examples:

- CHECKOUT and CHECKIN are version-control variants corresponding to GET and PUT, but with the added functionality of locking and unlocking, respectively, the data object for access/change by other users.
- TEXTSEARCH and SPACEJUMP are extended forms of GET applied to search and map coordinate positioning.
- LINK and UNLINK are variants of POST to add and remove meta information (object header information) to an object, without touching the object's content.

In fact, it is relatively rare to see PUT used these days. Most servers are configured to deny such direct, external publishing requests, except perhaps in specific, well-authenticated contexts. Instead, POST is used to pass data from client to server in a more open-ended way, by requesting that a server-defined link be created to the passed object. Although, traditionally, POST is used to create, annotate, and extend server-stored information, it was successfully MTME-extended in v1.1 to serve as a generic data upload and download mechanism for modern browsers.

The use of POST allows tighter server control of any received information. Perhaps more relevant to security, it gives server control over processing, storage location, and subsequent access. The client may suggest a storage URI, but the server is never obliged to use it. Even if a POST is accepted by the server, the intended effect may be delayed or overruled by subsequent processing, human moderation, or batch processing. In fact,

the creation of a valid link may not only be significantly delayed, it may never occur at all.

Overall, the guiding design thought is that HTTP requests may be cached. From the perspective of the client/user, simple requests must be *stateless*, repeatable, and free of side-effects.

Bit 1.8 HTTP is, like most basic Internet protocols, stateless

State tracking is, however, desired in many situations, but must then be implemented using message-based mechanisms external to the base protocol. Information stored client-side in Web-browser 'cookies' are but one example of such workaround measures.

- It must be noted that POST is strictly speaking neither repeatable nor free from side-effect, and thus not a proper replacement for PUT (which is both). In fact, POST requests are by definition declared *noncachable*, even though it might prove possible in some situations to cache them with no side-effects.
- Modern browsers correctly warn the user if a POST request is about to be reissued. The reason is of course that a POST often alters information state at the server, for example to generate content change, or to commit a unique credit-card transaction.

State considerations place serious constraints on possible functionality in the context of the current HTTP Web.

Extending in Other Layers

Extending the transport protocol (HTTP) is by no means the only proposed solution to extending Web functionality. The growing adoption of **XML**, not just in preference to **HTML** as markup but crucially as a real language for defining other languages, provides an alternative extension framework.

XML is more broadly application-managed than HTML, and its functionality definitions are independent of whether the underlying HTTP is extended or not. Implementations include message-based protocols to extend Web functionality by embedding the extension methods inside the message content passed by the base protocol.

The use of XML for inter-company remote operations became prevalent in 2001, mainly because of its ability to encapsulate custom functionality and convey specialized content meaning. This kind of functionality extension is what led to the development of **Web Services**, a term implying standard functionality distributed across the Web.

Even the markup change alone has direct potential benefits on existing Web functionality – for example, search. Most search engines read the format languages, usually HTML tags that may often be applied inappropriately from the point of view of logical structure. Consequently, the search results tend to reflect the formatting tags rather than actual page content as expressed in natural language. XML markup can express semantic meaning and thus greatly improve search-result relevancy.

There are both pros and cons in the XML approach to extending the Web framework. Some of both aspects hinge on the fact that the implemented message passing often relies on

the HTTP POST method. Embedding messages in GET headers imposes unacceptable message constraints.

Extended headers, for example, might be passed as POST content and thus be opaque to the HTTP agent. When referring to a particular resource addresses, this embedded reference cannot be cached or stored in the way a GET-request URI can.

- A typical workaround is a syntax extraction procedure to let the agent recover a workable URI from the POST header response to a GET request. The solution allows HTTP agents not to consider content-embedded extensions when responding to arbitrary requests.

The XML extension approach shares header opaqueness, seen from the point of view of HTTP agents (such as servers, proxies, and clients), with more proprietary extensions anchored in specific applications. Nonetheless, XML extension at least has the virtue of being an open standard that can be adopted by any implementation and be freely adapted according to context. It is ongoing work.

Functionality extension is not simply about extending the protocol; other issues must also be considered.

Content Considerations

A major obstacle facing anyone who wishes to extend the usefulness of the Web information space to automated agents is that most information on the Web is designed explicitly for human consumption.

Some people question what they see as a 'geek' focus on extending and automating the Web with new gee-wizardry. Instead, they suggest that most people seem happy with the Web as designed for human browsing, and do not necessarily have a problem with that. Most people, they say, are finding more or less what they want, and are finding other things of interest as they browse around looking for what they want.

Just because current Web content is designed supposedly for humans does not mean that it is a 'successful' design, however. For example, published Web content commonly relies on any number of implied visual conventions and assumptions to convey intended structure and meaning. Many of these often complex stylistic elements are neither easily nor unambiguously interpreted by 'most people' even with explicit instruction, and much less is it possible to decode them by machines. Furthermore, such style conventions can vary significantly between different authors, sites, and contexts.

Embedded metadata describing the intent behind arbitrary visual convention would not just aid machines, but could provide unambiguous cues to the human reader – perhaps as consistent client-side styling according to reader preference, or as cue pop-up.

- The advantage of such explicit guidance is most obvious in the case of browsers for the visually impaired, a group of human readers who often have significant problems dealing with existing content supposedly designed for them.

Even in cases where the information is derived from a database, with reasonably well-defined meanings for at least some terms in its tabled layout, the structure of the presented data is not necessarily evident to a robot browsing the site. Almost any such

presentation has implicit assumptions about how to interpret and connect the data. Humans readers familiar with the material can deal with such assumptions, but unaided machines usually cannot.

For that matter, the material might not even be possible to browse for a machine following embedded hyperlinks. Other, visual navigational conventions might have been implemented (such as image maps, form buttons, downloaded *ActiveX* controls, scripted mouse-state actions, flash-animated widgets, and so on).

Bit 1.9 Machine complexities that depend on human interpretation can signal that the problem statement is incorrectly formulated

A fundamental insight for any attempt to solve a technological problem is that tools and technologies should be used in appropriate ways. In cases where it becomes increasingly difficult to apply the technology, then it is likely that significant advances require fundamentally changing the nature of the problem.

Instead of tackling the perhaps insurmountable artificial-intelligence problem of training machines to behave like people ('greater AI'), the Semantic Web approach is to develop languages and protocols for expressing information in a form that can be processed by machines.

- This 'lesser AI' approach is transparent to the human users, yet brings information-space usability considerably closer to the original and inclusive vision.

Another content issue is the proliferation of different content formats on the Web, which usually require explicit negotiation by the user. This barrier not only complicates and hides the essential nature of the information on the Web, but also makes the content more difficult to handle by software agents.

Granted, the initial design of HTTP did include the capability of negotiating common formats between client and server. Its inclusion was based on the correct assumption that the ideal of a single content format would remain a vision obstructed by the wild proliferation of proprietary data formats. However, this negotiation feature has never really been used to advantage. The rapid adoption of HTML as the common formatting language of the Web did make the need less pressing, but the real reason is that the ever-larger list of possible formats made automatic negotiation impractical.

Instead content format became classified (and clarified, compared to traditional extension-based advertising) by importing the MIME-type concept from the realm of e-mail. Reading the MIME declaration tells the handler/client how to handle an otherwise opaque data stream, usually deferring interpretation to a specified client-side application.

- Although MIME declarations now formally refer to a central registry kept by **IANA**, there is no reason why the Web itself cannot be used as a distributed repository for new types.
- A transition plan would allow such a migration. Unqualified MIME types are in this scheme interpreted as relative URIs within a standard reference URI, in an online MIME registry.

Services on the Web

The issues of distributed services (**DS**) and remote procedure calls (**RPC**) constitute a kind of parallel development to the previously considered Web extensions, and are also considered a future part of the Semantic Web. At present, they coexist somewhat uneasily with the Web, since they operate fundamentally outside Web address space yet fulfill some of the same functionality as the proposed **Web Services** (**WS**).

DS and RPC applications often use proprietary protocols, are highly platform dependent, and are tied to known endpoints. WS implementations function within the Web address space, using Web protocols and arbitrary endpoints. They also differ from DS and RPC remote operation work in that WS transactions are less frequent, slower, and occur between non-trusted parties. Issues such as 'proof of delivery' become important, and various message techniques can become part of the relevant WS protocols.

Further discussion of these issues is deferred to later chapters. Instead the next few sections trace the conceptual developments that underpin the Semantic Web.

From Flat Hyperlink Model

In the Beginning was the Hyperlink . . . The click-to-browse hyperlink, that core convention of text navigation, and ultimately of Web usability, is in reality a *key-data pair* interpreted in a particular way by the client software. It reads as a visible anchor associated with a hidden representation of a Web destination (usually the URL form of a URI).

The anchor can be any structural element, but is commonly a selection of text or a small graphic. By convention, it is rendered visually as underlined or framed, at least by default, though many other rendering options are possible.

Bit 1.10 Good user interface design imbues simple metaphors with the ability to hide complex processing

The clicked text hyperlink in its basic underlined style might not have been visually very pleasing, yet it was easily implemented even in feature-poor environments. Therefore, it quickly became the ubiquitous symbol representing 'more information here'.

Functionally, the hyperlink is an *embedded pointer* that causes the client to request the implied information when it is activated by the appropriate user action.

The pragmatics of this mechanism, and the reason it defined the whole experience of the Web, is that it makes Web addressing *transparent to the user.* A simple sequence of mouse clicks allows the user freely to browse content with little regard for its location.

The concept is simple and the implementation ingenious. The consequences were far-reaching. Figure 1.1 illustrates this concept.

This 'invention of the Web' has generally been attributed to Tim Berners-Lee (knighted for his achievements in 2004, thus now Sir Tim), who also founded the World Wide Web Consortium (W3C, *www.w3.org*) in 1994. A brief retrospective can summarize the early development history.

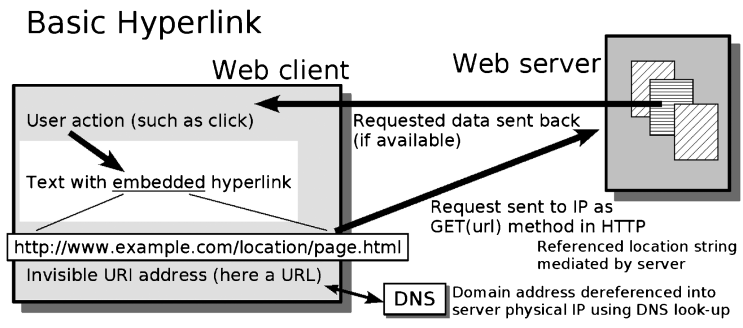

Figure 1.1 Conceptual view of hyperlink functionality as it is currently implemented to form the World Wide Web. The interlinking hyperlinks in Web content provide a navigational framework for users browsing it

In the late 1980s, Tim led an effort with Robert Cailliau at the CERN nuclear research center in Switzerland to write the underlying protocols (including HTTP) for what later came to be known as the World Wide Web. The protocols and technologies were disseminated freely with no thought of licensing requirements.

The early work on the Web was based on, among other things, earlier work carried out by Ted Nelson, another computer and network visionary who is generally acknowledged to have coined the term 'hypertext' in 1963 and used it in his 1965 book, *Literary Machines.* Hypertext linking subsequently turned up in several contexts, such as in online helpfiles and for CD-content navigation, but really only became a major technology with the growth of the public Web.

The matter of public availability deserves further discussion. Although it may seem like a digression, the issues raised here do in fact have profound implications for the Web as it stands now, and even more so for any future Semantic Web.

Patents on the Infrastructure

Since hyperlink functionality is seen as a technology, like any other innovation, it is subject to potential issues of ownership and control.

Even though the W3C group published Web protocols and the hyperlink-in-HTML concept as open and free technology (expressed as 'for the greater good'), it was only a matter of time before somebody tried to get paid for the use of such technologies when the economic incentive became too tempting.

The issue of patent licensing for the use of common media formats, client implementations, and even aspects of Internet/Web infrastructure is problematic. While the legal outcome of individual cases pursued so far can seem arbitrary, taken together they suggest a serious threat to the Web as we know it. Commercialization of basic functionality and infrastructure according to the models proposed by various technology stakeholders would be very restrictive and thus unfortunate for the usability of the Web.

Yet such attempts to claim core technologies seem to crop up more often. In some cases, the patent claims may well be legitimate according to current interpretations – for example, proprietary compression or encryption algorithms. But in the context of the Web's status as a global infrastructure, restrictive licensing claims are usually damaging.

Further complications arise from the fact that patents are issued by respective countries, each with its own variant patent laws. Trying to apply diverging country-specific patents to a global network that casually disregards national borders – in fact, appears innately to transcend such artificial boundaries – seems impossible.

It beomes especially difficult if one country's patented technology is another's public-domain technology (public-key cryptography was but one early example of this quandary). In practice, U.S. patent and copyright interpretations tend to be enforced on the Web, yet this interim state of affairs is unacceptable in the long term because it suggests that the global Internet is owned by U.S. interests.

If a particular technology is critical for a widely deployed functionality on the Web, allowing arbitrary commercial restrictions easily leads to unreasonable consequences. Many corporations, realizing this problem, are in fact open to free public licensing for such use (usually formally with the W3C). Sadly, not all business ventures with a self-perceived stake in Web technology are as amenable.

Bit 1.11 Commercialization (thus restriction) of access cripples functionality

This view, while not popular among companies that wish to stake out claims to potential pay-by-use markets, seems valid for information-bearing networks. The natural tendency is for access and transaction costs to rapidly go towards zero.

Unrestricted (free, or at least 'microcost') access benefits the network in that the increasing number of potential connections and relations exponentially increase its perceived value. Basic p2p network theory states that this value increases as a power relationship of the number of nodes – as 2^n.

Ownership issues should also be seen in relation to the general assault we can see on 'free' content. It is not easy to summarize the overall situation in this arena. Commercialization goals are pursued by some decidedly heavyweight interests, and often even long-established individual rights get trampled in the process. Legislation, interpretations, and application seem to shift almost daily.

At least for hyperlink technology, the legal verdict appears to be a recognition of its importance for the common good. For now, therefore, it remains free.

On the other hand, a failing perhaps of the current Web implementation is that it has no easy solutions to accommodate commercial interests in a user-friendly and unobtrusive way – the lack of a viable micropayment infrastructure comes to mind.

Bit 1.12 Issues of ownership compensation on the Web remain unresolved

Without transparent and ubiquitous mechanisms for tracking and on-demand compensation of content and resource usage, it seems increasingly likely that the Web will end up an impoverished and fragmented backwater of residual information.

Related to this issue is the lack of a clear demarcation of what exactly constitutes the 'public good' arena, one that despite commercial interests should remain free. Table 1.1

Table 1.1 The issue of 'free' usage and development or creation type

Type	Open	Proprietary
Free to use freely	free stuff	public good
Free to use, limited (licensed, non-commercial use)	community	promotional
Pay to use (buy or license)	rare, low cost	usual case

outlines one way of looking at the dimensions of 'free' in this context. Not specifically located in it is the 'free as in beer' stance – that is, affordable for anyone.

Hyperlink Usability

What, then, are the characteristics of usability of the hyperlink? Of particular interest to later discussions on extensibility is the question: In what way is the usability lacking in terms of how we would like to use the Web?

The current conceptual view of the hyperlink, as it has been implemented, has several implications for usability that are not always evident to the casual user. Current usage is deeply ingrained, and the functionality is usually taken for granted with little awareness of either original design intentions or potential enhancements.

The major benefit is of hiding the details of URI and URL addressing from the user, behind a simple user action: clicking on a rendered hyperlink. Another benefit is the concept of 'bookmarking' Web resources for later reference, again through a simple user action in the client. These actions quickly become second nature to the user, allowing rapid navigation around the Web.

On the other hand, despite the ingenious nature of the hyperlink construct, the deficiencies are numerous. They can be summarized as follows:

- **Unidirectional linkage**. Current implementations provide no direct means to ascertain whether hyperlinks elsewhere point to particular content. Even the simple quantitative metric that X sites refer to Web resource Y can be valuable as an approximate measure of authoritativeness. Mapping and presenting parent-child-sibling structures is also valuable. *Backlink* information of this nature can be provided indirectly by some search index services, such as *Google,* but this information is both frozen and often out-of-date.
- **Unverified destination**. There is no way to determine in advance whether a particular destination URL is valid – for example, if the server domain exists or the information is stored in the address space specified. The address can have been incorrectly entered into the hyperlink, or the server's content structure reorganized (causing the dreaded 'linkrot' effect).
- **Filtered access**. Many sites disallow direct or 'deep' linking, either by denying such linked access outright or by redirecting to some generic portal page on the site. Content bookmarked in the course of sequential browsing may therefore not be accessible when following such links out of context. Other access controls might also be in effect, requiring login or allowed categories of users – and increasingly, the subscription model of content access.

- **Unverified availability**. Short of trying to access the content and receiving either content or error code, a user cannot determine beforehand if the indicated content is available – for example, the server may be offline, incapable of responding, or just applying filtered access.
- **Relevance, reputation and trust issues**. The Web is superbly egalitarian; anyone can publish anything, and all content hyperlinks have the same superficial value. In the absence of mechanisms for hyperlink feedback, the user has no direct way of determining the relevance or trustworthiness of the linked content. Visual spoofing of link destination is easy, which can lead users astray to sites they would never wish to visit, something often exploited by malicious interests on Web pages.

As can be seen, the browsing user (also agents, and even the servers) could benefit from information about what exactly a hyperlink is pointing to, and from an indication of its status, *operational* as well as *reputational.*

One-way linkage cannot provide this information. Existing tools are 'out-of-band' in the sense that they are not integrated into the browsing experience (or the client-server transactions), but instead rely on external, human-interpreted aids such as search engines or separate applications (say hyperlink checkers).

Wouldn 't it be nice if ...

Some features users might find useful or convenient when browsing the Web can be listed as a sampler of what the current infrastructure and applications either do not do, or do only with difficulty or limitations in special cases/applications.

At this point, we are not concerned about the 'how' of implementation, only the user experience for some fairly simple and straightforward enhancements. Some of these features are hard to describe adequately in the absence of the 'back-end' data and automation, and they will depend greatly on commercial, political, and social decisions along the way as to what is desired or allowed.

- **Context awareness**, which would allow 'intelligent' client behavior. Ideally perhaps, semantic parsing of document context could reduce the need for user decision and simply present 'most-probable' options as distinctive link elements at the rendered location. Alternatively, the client might provide an on-the-fly generated sidebar of 'related links', including information about the site owner. Several proprietary variations of the theme are found in *adware*-related generation of 'spurious' hyperlinks in displayed content, which however is a more intrusive and annoying implementation in the absence of other cues.

  A simple example of a context feature for content is found in newer browser clients, where the user can highlight text and right-click to access Web search or other useful actions, as illustrated in Figure 1.2.
- **Persistent and shareable annotations**, where users can record private or public comments about Web content automatically attached to a given document (URI) and ideally internal location, in context. Public comments would then be shareable among an entire community of users – perhaps most usefully in goal-oriented groups.

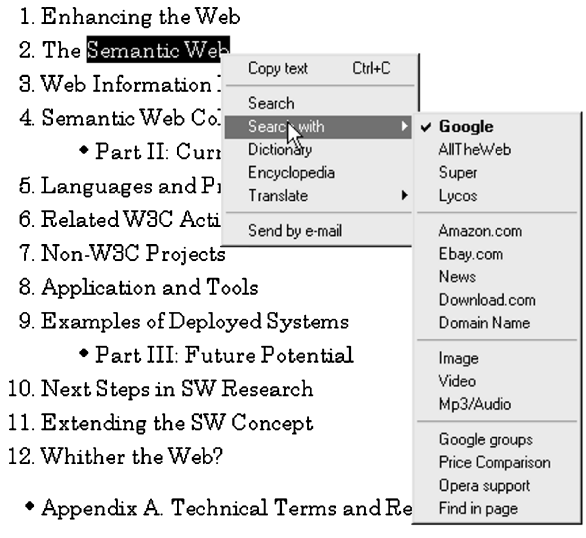

Figure 1.2 A simple example of context awareness that allows a browser user to initiate a number of operations (such as Web search, Dictionary or Encyclopaedia lookup, or translation) based on a highlighted block of text and the associated context menu

- **Persistent and shareable ratings**, which is a complement to annotations, providing some compiled rating based on individual user votes. A mouse-over of a link might then pop up a tooltip box showing, for example, that 67% of the voting users found the target document worthwhile. Of course, not everyone finds such ratings useful, yet such ratings (and their quality) might be improved by a consistent infrastructure to support them.
- **More realtime data**, which would result from the ability of Web clients and services to compile Web pages of data, or embed extended information, culled from a variety of sources according to the preferences of the user.
- **Persistent and shareable categorizations**, which would be a collaborative way to sort Web documents into searchable categories.

The common theme in most wish lists and methods of addressing the deficiencies of the current Web model is that the issues are based on adding information about the content. Then the user client can access, process, and display (or use it) in ways transparent but useful to the user – ideally with some measure of user control.

To Richer Informational Structures

The further information about the content that the previous shortcomings and wish list highlight include the metadata and relational aspects, some of which might be implemented as distributed services on the Web.

One idealized extension of the originally unidirectional information flow could look as in Figure 1.3.

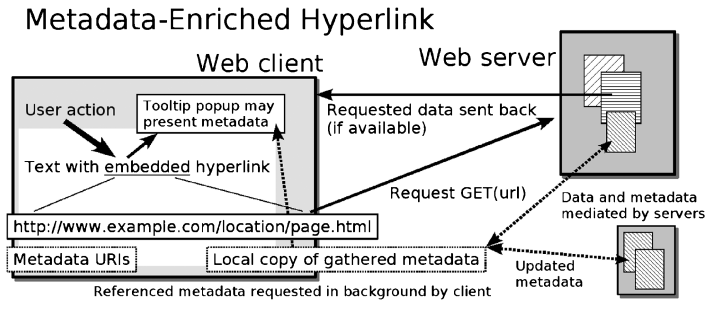

Figure 1.3 Conceptual view of hyperlink metadata functionality as it might be implemented to form an enhanced Web. Metadata and relational data are gathered and presented to the user for each embedded link

The point in the enriched hyperlink model is that further information about each hyperlink destination is automatically gathered and processed in the background. This behavior is an extension of how some caching solutions already now parse the currently loaded content and pre-fetch all hyperlink-referenced content into a local cache, ready for the next user selection.

In the new model, the client software would also be busy compiling information about the referenced content, ready to present this to the user even before a next destination is selected. Link metadata might be presented to the user as tooltip text, for example, or in ancillary windows on a mouse-over of the link.

A very limited and static simulation of this feature can be seen in the use of the 'title' attribute in a hyperlink markup (as shown in Figure 1.4) – the difference is that this kind of metainformation indicator must be precoded by the content author as an attribute in the hyperlink tag itself.

The Collaboration Aspect

The way that we have viewed the Web has generally been one of computer users browsing a web of content; independent readers of static information, or possibly also sole publishers of personal Web sites. This view was never the whole truth.

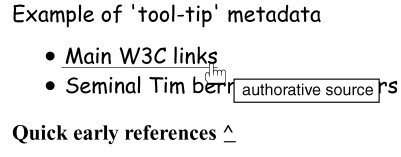

Figure 1.4 Static meta-information precoded by content author into the hyperlink tag of the content itself, using the 'title' attribute. When rendering the page, the browser pops up a 'tooltip' box to show the hidden text when the cursor is placed over the link

Bit 1.13 In the early days of the Web, considerable collaboration was the rule

What is interesting about the early collaborative effort is that it occurred in a smaller and more open community, with implicit trust relationships and shared values. Among those who maintained Web pages, compiling and publishing resource lists became a major part of the effort.

To find information on the Web, a user would visit one or another of these lists of hyperlinks and use it as a launchpad, long before the idea of 'portal sites' became popular on the Web. The implicit metadata in such a reference represents the value assessment of the person creating the list. Such personal lists became more specialized and interlinked as their maintainers cross-referenced and recommended each other, but they also became vastly more difficult to maintain as the Web grew larger.

Some directory lists, almost obsessively maintained, eventually grew into large indexing projects that attempted to span the entire Web – even then known to be an ultimately hopeless goal in the face of the Web's relentless and exponential growth.

One such set of lists, started as a student hobby by David Filo and Jerry Yang in 1994, quickly became a mainstay for many early Internet users, acquiring the name 'Yet Another Hierarchical Officious Oracle' (YAHOO). By 1996, Yahoo! Inc. (*www.yahoo.com*) was a viable IPO business, rapidly diversifying and attaining global and localized reach. *Yahoo!* quickly became the leader, in both concept and size, with a large staff to update and expand manually the categories from a mix of user submissions, dedicated browsing, and eventually search-engine database.

Other search engines went the other way: first developing the engine, then adding directories to the Web interface. A more recent collaborative venture of this kind is the *Open Directory project* (*www. dmoz. org*).

An offshoot to cross-linked personal lists and a complement to the ever-larger directories was the *'WebRing'* (*www.webring.org*) concept. Sites with a common theme join an indexed list (a 'ring') maintained by a theme ringmaster. A master meta-list of categories tracks all member rings.

- To appreciate the size of this effort, *WebRing* statistics were in July 2002 given as 62,000 rings and 1.08 million active sites, easily capable of satisfying any user's special interests indefinitely.
- However, the *WebRing* phenomenon appears to have peaked some years ago. presumably shrinking in the face of the ubiquitous search-engine query. The figures show a clear trend: 52,250 rings with 941,000 active sites in August 2003, and 46,100 rings with 527,000 sites in February 2005

The individual user of today tends increasingly to use the more impersonal and dynamic lists generated by search engines. An estimated 75% of user attempts to seek Web information first go through one of the major search engines.

Although search-engine automation has reduced the relative importance of individual collaboration in maintaining the older forms of resource lists, it is still not the panacea that most users assume. Issues include incomplete indexing of individual

documents, query-relevancy ranking, and handling of non-HTML and proprietary formats.

Search engines also face the same problem as other approaches of scaling in the ever-growing Web. They are unable to index more than a small fraction of it. Estimates vary, but the cited percentage has been shrinking steadily – recently 10%, perhaps even less.

The bottom line is that centralized indexing in the traditional sense (whether manual or automatic) is increasingly inadequate to map the distributed and exponential growth of the Web – growth in size, variety, updates, etc.

Bit 1.14 New requirements and usage patterns change the nature of the Web

The adaptation of any network is driven by requirements, technology adoption, resource allocation, and usage – if only because new and upgraded resources usually target areas of greatest loading in order to maintain acceptable levels of service. *Peer to Peer* (a previous book) is replete with examples.

Returning to collaboration, it has become more important to support new and more flexible forms – between individuals and groups, between people and software, and between different software agents.

Loose parallels could be drawn between this adaptive situation for the Web and the changing real-world requirements on telephone networks:

- As first, subscriber conversation habits changed slowly over the years, and group conferencing became more common for business.
- Later, cellular usage became ubiquitous, requiring massive capacity to switch calls between line and mobile subscribers from many diverse providers.
- As dial-up Internet access became more common, individual subscriber lines and exchanges became increasingly tied up for longer intervals.
- Digitizing the network and applying packet switching to all traffic helped solve some congestion problems by letting separate connections coexist on the same circuit.
- In all cases, telecom operators had to adapt the network, along with associated technologies, to maintain an acceptable service in the face of shifting requirements.

The Web is a virtual network. The underlying Internet infrastructure (and the physical access and transport connectivity under it) must constantly adapt to the overall usage patterns as expressed at each level. New collaboration technology has generally just deployed new application protocols on top of the existing Internet and Web ones, forming new virtual networks between the affected clients.

Better collaboration support is achieved by instead enhancing the basic Web layer protocol. Other, proprietary solutions are possible (and some do exist), but making these features a fundamental part of the protocol ensures maximum interoperability. Just as we take for granted that any Web browser can display any standard-compliant Web page, so should clients be able to access collaboration functionality. Providing mechanisms for dealing with content meaning that will automatically provide intelligent mechanisms for promoting collaboration in all its forms is also one of the aims of the Semantic Web.

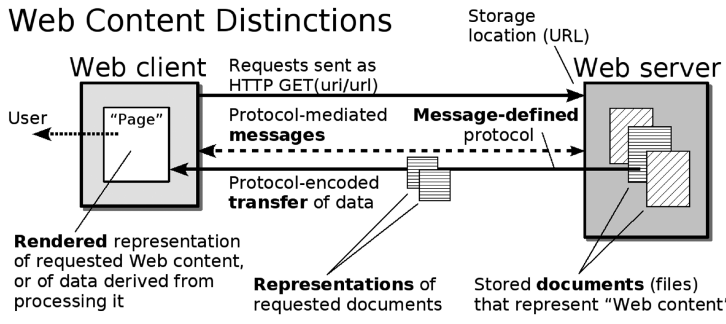

Figure 1.5 The traditional distinction between ''document'' and ''message'' is illustrated in the context of the client-server model, along with some terms relevant to requesting and viewing Web content

Extending the Content Model

Discussions of 'content' without qualifications or explanations might be misleading for several reasons, and the term is easily misunderstood; this is why the terms 'information' and 'information object' crop up in these discussions.

It is a common assumption that 'Web content' refers to the text or other media objects provided by Web pages and their related links – static information physically stored in well-defined server locations. In addition, historically, there has always been a distinction maintained between 'messages' on the one hand, and 'documents' (or files, or content) on the other (see Figure 1.5); that protocols are defined on top of messages in order to transport documents, and that further protocols are defined in terms of messages exchanged by previously defined protocols, and so on.

Such recursive definitions, while sometimes confusing, occur all the time and can prove very useful. As implied earlier, p2p protocols that define separate virtual networks are defined by the exchange of underlying HTTP messages.

However, especially in the context of XML, it is often better to view *content as the side-effect of an ongoing exchange of messages between various endpoints.* It matters not whether these endpoints are servers of static Web pages, database servers, Web services, other users, or any form of software agent. Also, it does not matter whether the information is actually stored somewhere, compiled from several sources, or generated totally on the fly.

Higher-level protocols are generally implemented as *message-based transactions* that can encapsulate information and pointers to information in many complex ways.

Recall that requesting information on the Web, even in a basic protocol such as HTTP, only returns a representation of the information, encoded in some way acceptable at both endpoints – only a message, in effect, from the process providing it and thus inextricably bound to the protocol.

Bit 1.15 Messages pass representations of information

The issue of information representation by proxy and subsequent interpretation is far more fundamental than it might seem. The concept has deep philosophical roots, and far-reaching implications about how we know anything at all and communicate.

This generalized view, that all content is the same exchange of messages expressed in a single protocol, amounts to a useful simplification. It can result in a merging of the current fragmentation of different protocols for e-mail, file transfer, newsgroup postings, instant messaging, chat, Web pages, and other 'content'.

The same client, based on such a harmonized protocol, could transparently handle arbitrary information objects in a consistent way from the user perspective.

Mapping the Infosphere

Navigating the wealth of information on the Web is not a trivial task, even assuming that it is all openly accessible. Various methods have gradually increased or decreased in relative importance for users over the years. In roughly chronological order, these navigational methods are summarized in the following list:

- **Resource lists** present hyperlinks that particular users have collected and recommended on their own Web sites. An outgrowth of this resource was the 'meta list', or list of hyperlinks to sites that maintained such lists. The quality and comprehensiveness of such lists could vary immensely.
- **Meta directories** are large hyperlink lists compiled by dedicated staff into some kind of hierarchical category trees, often with the aim of being a comprehensive guide to all content. Examples are the *Yahoo! Web Site Directory* (*www.yahoo.com*), *Google directory* (*directory.google.com*) and the *Open Directory Project* (*dmoz.org*). A problem with hierarchical listings is classification, which can be arbitrary, and may or may not coincide with the user's perception of an appropriate category.
- **Search engines** provide *ad hoc* lists of hyperlinks to Web pages (or other Internet resources) culled from indexed databases collected and updated by automated 'spiders' that systematically traverse the Web hyperlinks on each Web site they can reach. Search results are often ranked according to some opaque metric of 'relevancy', but depend critically on the user's ability to formulate 'optimal' search criteria based on suitable terms, along with the quality of the indexing process.
- **Special interest compilations** lie somewhere between idiosyncratic lists compiled by individuals, and the comprehensive Web directories. They aggregate self-nominated, like-themed sites into interlinked Web rings. Each site in a ring can directly or indirectly refer to its neighbors through a ring index, each ring organized in a category hierarchy in some meta ring.
- **Web services** can automate and manage resource discovery and content retrieval (including compilation and recasting) on behalf of a user. Such services might incorporate 'learning routines' to adapt to each user's history of requests to determine scope and relevancy criteria, for example.

Currently, Web information is largely 'atomistic', with relatively few and arbitrary relations defined between the different information objects – see also the previous discussion about hyperlink usability. This deficiency and the constraints of the current hyperlink model clearly have a severe impact on user navigation. The lack also severely hampers the development of automated agents for such purposes.

Structure and Relationship

Traditionally, maps of Internet resources in general, and the Web in particular, have by necessity been very simplistic and one-dimensional:

- **Human-compiled hierarchy directories** have the virtue of including some, albeit variable, amount of metadata based on human assessment of the link target and classification into categories. Expressed relational metainformation is an external construct, however, independent of any explicit hyperlinks between sites.
- **Automated link-walk compilations** result in maps that are in this case purely structural – a 'snapshot' of Web hyperlinks. While subsequent indexing of at least portions of content from each node does allow for term searches to create simple maps that reflect literal term usage (and thereby indicate term-related linking), the content semantics are largely ignored.

We can illuminate this distinction and some of the issues involved by looking to some well-established conventions from paper publishing.

Paper-published content is inherently linear, because it has its origins in oral narrative. Unlike transient speech, however, published sentences are frozen in a medium that the reader can to some extent perceive outside of time, freely choosing what to read at any time. This characteristic enables the concept of multiple entry points based on some overall mapping.

Paper publishing thus developed structural and semantic aids to the reader to provide such entry points into the content:

- **Chapters and sections**. As a structural device, the division of a text into chapters and sections imparts a layer of semantics in that the logical division helps order content and implies relationships between the parts.
- **Table of contents**. Given an expressed logical structure, it is possible to provide a list of headings and corresponding numbered pages. The reader can use the table of contents to gain an overview of the content structure and quickly find specific content by following the logical divisions.
- **Abstracts and summaries**. Older novels commonly provided short abstracts for each chapter, and although this practice was later dropped in fiction, the abstract lives on in academic publications. Summaries (especially executive summaries) are a similar device. Both variants are commonly set apart from the main text by special styling (such as extra-wide margins, vertical whitespace, and italic font). A listing of keywords may be considered as an extremely abbreviated summary.
- **Index**. From the other end (and it is not coincidence that the index is found at the back of a book), a list of terms and their page-location in the text constitutes an index. Creating a useful index is a highly developed art; it is not simply a compilation of occurrences of all the words. Instead It critically relies on the compiler's understanding of relevancy and importance – of which words or phrases to include, which occurrences to index in terms of the semantic context, which occurrences to ignore, which relationships to highlight using categories and multi-level listings, just to name a few considerations.

Other semantic-mapping contructs also exist in publishing, but those listed here suffice for the purpose of comparison with the Web. For example, comparing with the previous

examples of Web maps, we can see a rough correspondence between the human-compiled category list and the table of contents.

- In passing, we may note that despite well-developed mechanisms for multiple entry points for the reader, not to mention advanced non-linear reading possibilities in electronic form, most written content maintains the fiction of linear narrative, from beginning to end.

The search-engine list of links derived from automated indexing would perhaps approximate the index, except for the fact that such a list compilation is currently an index created solely by listing all occurrences of the search terms. It lacks the semantic order and weighting that the expert indexer imparts, and critically, the distinction between different usages and meanings – for example, between the noun 'book' and the verb 'to book' (as in make a reservation).

Natural language is replete with ambiguities of usage that only information about semantic context can even begin to resolve. With clever programming, Web searches can at least be made tolerant of some common typographical mistakes and regional spelling variations, but these searches remain semantically ignorant.

Proper logical markup of text structure in Web content can somewhat offset this analytical lack, but at the cost of reliance on external conventions that may or may not be correctly applied in particular instances. Text markup still does not convey inherent semantic information, only overlain syntactical.

Although *Google* is commonly cited as 'the best' search engine, mainly due to the way it leverages backlinks to provide relevant results ranking, it is not hard to find consistently 'bad' or skewed results in simple searches.

- Such frustrations are often due to the engine's strictly lexical basis combined with the skew effect of some kinds of linked content. For search words that also relate to product names, for example, *Google* ranks online shopping pages higher than any other results, probably because of the wealth of product links both on-site and from affiliate sites. Advanced syntax may therefore be needed to find information sites not devoted to selling products.

Links, Relationship, and Trust

The fact that object **A** unidirectionally links to object **B** says little about any real relationship between them. Any metainformation about this relationship is largely inferred by the user. For example, we are willing to believe in a high degree of relevance of a hyperlink on a page that refers to another page on the same site, even in the absence of any explicit declarations about it. Such trust may prove unfounded, however, especially if the link is 'broken' (that is, the URL is temporarily or permanently invalid).

But the point is that the hyperlink itself contains no metadata whatsoever about the relationship or target status. At best, the contextual text around the link might explain something about the target page; at worst, it might be totally misleading.

The situation is even less reliable when the link targets an object outside the site, because then the page/link author does not even have nominal control over the target object. Target site contact might have changed drastically since the link was created, for example, or the

target site might employ filtering or redirection that makes the direct reference invalid and the target hard or impossible to track down manually.

Trust is also at issue when a link (URL) points to a Web page that defines an association between URL and URI – perhaps the easiest way to create and publish a URI Anyone can do it, and anyone can on the face of it misrepresent an association to a particular URI.

Trust issues of this nature are probably best resolved by embedding currently accepted mechanisms based on content hashes and public key signing of the published assertion. The assertion can then be independently verified as unaltered and the publisher identity verified through some established chain of trust – in the worst case, just identified as the same publisher of other similarly-signed material.

Well-Defined Semantic Models

When considering strategies to map information on the Web, one can begin by outlining some of the features that the mapped content, commonly expressed as 'documents' in these contexts, should exhibit or promote:

- It must be possible to define precisely a language (as a set of tokens, grammar, and semantics) in which to deal with 'first class objects' (which means document objects that are URI-referenced). Such a language is a prerequisite to processing.
- It must be possible to make documents in a mixture of languages (that is, allow language mixing). It is a fact of life that no single language can or will be used in all contexts. Extensibility is natural and alternate structures must be allowed to coexist.
- Every document should be self-defining by carrying the URI(s) of the language(s) in which it is written. This mechanism enables both humans and more importantly software to determine the appropriate methods for analyzing and processing the information.
- It must be possible to process a document while understanding only a subset of the languages (that is, partial understanding). In other words, full functionality must 'degrade gracefully' so that simpler processing can safely ignore the full set of features.

The requirements in this list form the basis for XML development, for example, and it defines a first step towards enabling the encoding of semantic meta-information.

The current Web recommendation is that XML should be used when a new language is defined. A new language (or, for that matter, new features extending an existing language) must then be defined as a new namespace within XML.

Although this kind of language defining might be seen as akin to misguided attempts to (re)define all natural languages in terms of some grand metalanguage (as in the tradition of applying Latin grammar), as an artificial construct *a priori* applied to other artificial constructs, it can be both attainable and practical.

Given the language, how do we deal with the meaning of what is expressed in that language? This issue is at the core of the Semantic Web, and a discussion around it forms the bulk of Chapter 2.