

Chapter 2

Architecture of the World Wide Web

All the important revolutions that leap into view must be preceded in the spirit of the era by a secret revolution that is not visible to everyone, and still less observable by contemporaries, and that is as difficult to express in words as it is to understand.

G.W.F. Hegel (1959)

In order to establish the relative autonomy of the Web as a subject matter, we recount its origins and so its relationship to other projects, both intellectual such as Engelbart's Human Augmentation Project, as well as more purely technical projects such as the Internet (1962). It may seem odd to begin this book, which involves very specific questions about representation and meaning on the Web, with a historical analysis of the Web. To understand these questions we must first have an understanding of the boundaries of the Web and the normative documents that define the Web. The Web is a fuzzy and ill-defined subject matter – often considered a ill-defined 'hack' by both academic philosophers and computer scientists – whose precise boundaries and even definition are unclear. Unlike some subject matters like chemistry, the subject matter of the Web is not necessarily very stable, like a 'natural kind,' as it is a technical artifact subject to constant change. So we will take the advice of the philosopher of technology Gilbert Simondon, "Instead of starting from the individuality of the technical object, or even from its specificity, which is very unstable, and trying to define the laws of its genesis in the framework of this individuality or specificity, it is better to invert the problem: it is from the criterion of the genesis that we can define the individuality and the specificity of the technical object: the technical object is not this or that thing, given *hic et nunc*, but that which is generated" (1958). In other words, we must first trace the creation of the Web before attempting to define it, imposing on the Web what Fredric Jameson calls "the one absolute and we may even say 'transhistorical' imperative, that is: Always historicize!" (1981). Only once we understand the history and significance of the Web, will we then proceed to dissect its components one-by-one, and attempt to align them with certain still-subterranean notions from philosophy.

2.1 The History of the Web

What is the Web, and what is its significance? At first, it appears to be a relative upstart upon the historical scene, with little connection to anything before it, an historical and unprincipled ‘hack’ that came unto the world unforeseen and with dubious academic credentials. The intellectual trajectory of the Web is a fascinating, if unknown, revolution whose impact has yet to be historically comprehended, perhaps even by its creators. Although it is well-known that the Web bears some striking similarity to Vannevar Bush’s ‘Memex’ idea from 1945, the Web is itself usually thought of more as a technological innovation rather than an intellectually rich subject matter such as artificial intelligence or cognitive science (1945). However, the Web’s heritage is just as rich as artificial intelligence and cognitive science, and can be traced back to the selfsame root, namely the ‘Man-Machine Symbiosis’ project of Licklider (1960).

2.1.1 The Man-Machine Symbiosis Project

The first precursor to the Web was glimpsed, although never implemented, by Vannevar Bush, chief architect of the military-industrial complex of the United States of America. For Bush, the primary barrier to increased productivity was the lack of an ability to easily recall and create records, and Bush saw in microfiche the basic element needed to create what he termed the “Memex,” a system that lets any information be stored, recalled, and annotated through a series of “associative trails” (1945). The Memex would lead to “wholly new forms of encyclopedias with a mesh of associative trails,” a feature that became the inspiration for links in hypertext (Bush 1945). However, Bush could not implement his vision on the analogue computers of his day.

The Web had to wait for the invention of digital computers and the Internet, the latter of which bears no small manner of debt to the work of J.C.R. Licklider, a disciple of Norbert Wiener (Licklider 1960). Wiener thought of feedback as an overarching principle of organization in any science, one that was equally universal amongst humans and machines (1948). Licklider expanded this notion of feedback loops to that of feedback between humans and digital computers. This vision of ‘Man-Machine Symbiosis’ is distinct and prior to cognitive science and artificial intelligence, both of which were very infantile disciplines at the time of Licklider, and both of which are conjoined at the hip by hypothesizing that the human mind can be construed as either computational itself or even implemented on a computer. Licklider was not a true believer in the computational mind, but held that while the human mind itself might not be computational (Licklider cleverly remained agnostic on that particular gambit), the human mind was definitely complemented by computers. As Licklider himself put it, “The fig tree is pollinated only by the insect *Blastophaga grossorum*. The larva of the insect lives in the ovary of the fig tree,

and there it gets its food. The tree and the insect are thus heavily interdependent: the tree cannot reproduce without the insect; the insect cannot eat without the tree; together, they constitute not only a viable but a productive and thriving partnership. This cooperative ‘living together in intimate association, or even close union, of two dissimilar organisms’ is called symbiosis. The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today” (1960). The goal of ‘Man-Machine Symbiosis’ is then the enabling of reliable coupling between the humans and their ‘external’ information as given in digital computers. To obtain this coupling, the barriers of time and space needed to be overcome so that the symbiosis could operate as a single process. This required the invention of ever decreasing low latency feedback loops between humans and their machines.

In pursuit of that goal, the ‘Man-Machine Symbiosis’ project was not merely a hypothetical theoretical project, but a concrete engineering project. In order to provide the funding needed to assemble what Licklider termed his “galactic network” of researchers to implement the first step of the project, Licklider became the institutional architect of the Information Processing Techniques Office at the Advanced Research Projects Agency (ARPA) (Waldrop 2001). Licklider first tackled the barrier of time. Early computers had large time lags in between the input of a program to a computer on a medium such as punch-cards and the reception of the program’s output. This lag could then be overcome via the use of time-sharing, taking advantage of the fact that the computer, despite its centralized single processor, could run multiple programs in a non-linear fashion. Instead of idling while waiting for the next program or human interaction, in moments nearly imperceptible to the human eye, a computer would share its time among multiple humans (McCarthy 1992).

In further pursuit of its goal of human-machine symbiosis, in which some over-enthusiastic science-fiction fans or academics with a penchant for the literal might see the idea of a cyborg, the ‘Man-Machine Symbiosis’ project gave funding to two streams of research: artificial intelligence and another lesser-known strand, the work on ‘human augmentation’ exemplified by the Human Augmentation Project of Engelbart (1962). Human augmentation, instead of hoping to replicate human intelligence as artificial intelligence did, only thought to enhance it. At the same time Licklider was beginning his ‘Man-Machine Symbiosis’ project, Douglas Engelbart had independently generated a proposal for a ‘Human Augmentation Framework’ that shared the same goal as the ‘Man-Machine Symbiosis’ idea of Licklider, although it differed by placing the human at the centre of the system, focusing on the ability of the machine to extend to the human user. In contrast, Licklider imagined a more egalitarian partnership between humans and digital computers, more akin to having a somewhat intelligent machine as a conversational partner for the human (1962). This focus on human factors led Engelbart to the realization that the primary reason for the high latency between the human and the machine was the interface of the human user to the machine itself, as a keyboard was at

best a limited channel even compared to punchcards. After extensive testing of what devices enabled the lowest latency between humans and machines, Engelbart **invented the mouse** and other, less successful interfaces, like the one-handed ‘chord’ keyboard (Waldrop 2001). By employing these interfaces, the temporal latency between humans and computers was decreased even further. Strangely enough, we have not – despite all the hyperbole around tactile or haptic interfaces from various media-labs – gone far beyond keyboards, mice, and touch-screens in 50 years.

2.1.2 *The Internet*

The second barrier to be overcome was space, so that any computer should be accessible regardless of its physical location. The Internet “came out of our frustration that there were only a limited number of large, powerful research computers in the country, and that many research investigators who should have access to them were geographically separated from them” (Leiner et al. 2003). Licklider’s lieutenant Bob Taylor and his successor Larry Roberts contracted out Bolt, Beranek, and Newman (BBN) to create the **Interface Message Processor**, the hardware needed to connect the various time-sharing computers of Licklider’s “galactic network” **that evolved into the ARPANet** (Waldrop 2001). While BBN provided the hardware for the ARPANet, the software was left undetermined, so an **informal group of graduate students constituted the Internet Engineering Task Force (IETF)** to create software to run the Internet (Waldrop 2001).

The IETF has historically been the main standardization body that creates the protocols that run the Internet. It still maintains the informal nature of its foundation, with no formal structure such as a board of directors, although it is officially overseen by the Internet Society. The IETF informally credits as their main organizing principle the credo “We reject kings, presidents, and voting. We believe in rough consensus and running code” (Hafner and Lyons 1996). Decisions do not have to be ratified by consensus or even majority voting, but require only a rough measure of agreement on an idea. The most important product of these list-serv discussions and meetings are **IETF RFCs (Request for Comments)** which differ in their degree of reliability, from the unstable ‘Experimental’ to the most stable ‘Standards Track.’ **The RFCs define Internet standards such as URIs and HTTP** (Berners-Lee et al. 1996 2005). RFCs, while not strictly academic publications, have a de facto normative force on the Internet and therefore on the Web, and so they will be referenced considerably throughout this book.

Before the Internet, networks were assumed to be static and closed systems, so one either communicated with a network or not. However, early network researchers determined that there could be an **“open architecture networking”** where a **meta-level “internetworking architecture”** would allow diverse networks to connect to each other, so that “they required that one be used as a component of the other, rather than acting as a peer of the other in offering end-to-end service”

(Leiner et al. 2003). In the IETF, Robert Kahn and Vint Cerf devised a protocol that took into account, among others, four key factors, as cited below (Leiner et al. 2003):

1. Each distinct network would have to stand on its own and no internal changes could be required to any such network to connect it to the Internet.
2. Communications would be on a best effort basis. If a packet didn't make it to the final destination, it would shortly be retransmitted from the source.
3. Black boxes would be used to connect the networks; these would later be called gateways and routers. There would be no information retained by the gateways about the individual flows of packets passing through them, thereby keeping them simple and avoiding complicated adaptation and recovery from various failure modes.
4. There would be no global control at the operations level.

In this protocol, data is subdivided into 'packets' that are all treated independently by the network. Data is first divided into relatively equal sized packets by TCP (Transmission Control Protocol), which then sends the packets over the network using IP (Internet Protocol). Together, these two protocols form a single protocol, TCP/IP (Cerf and Kahn 1974). Each computer is named by an Internet Number, a 4 byte destination address such as *152.2.210.122*, and IP routes the system through various black-boxes, like gateways and routers, that do not try to reconstruct the original data from the packet. At the recipients end, TCP collects the incoming packets and then reconstructs the data.

The Internet connects computers over space, and so provides the physical layer over which the universal information space of the Web is implemented. However, it was a number of decades before the latency of space and time became low enough for something like the Web to become not only universalizing in theory, but universalizing in practice, and so actually come into being rather than being merely a glimpse in a researcher's eye. An historical example of attempting a Web-like system before the latency was acceptable would be the NLS (oNLine System) of Engelbart (1962). The NLS was literally built as the second node of the Internet, the Network Information Centre, the ancestor of the domain name system. The NLS allowed any text to be hierarchically organized in a series of outlines, with summaries, giving the user freedom to move through various levels of information and link information together. The most innovative feature of the NLS was a journal for users to publish information in and a journal for others to link and comment upon, a precursor of blogs and wikis (Waldrop 2001). However, Engelbart's vision could not be realized on the slow computers of his day. Although time-sharing computers reduced temporal latency on single machines, too many users sharing a single machine made the latency unacceptably high, especially when using an application like NLS. Furthermore, his zeal for reducing latency made the NLS far too difficult to use, as it depended on obscure commands that were far too complex for the average user to master within a reasonable amount of time. It was only after the failure of the NLS that researchers at Xerox PARC developed the personal computer, which by providing each user their own computer reduced the temporal latency to an acceptable amount (Waldrop 2001). When these computers were

connected with the Internet and given easy-to-use interfaces as developed at Xerox PARC, both temporal and spatial latencies were made low enough for ordinary users to access the Internet. This convergence of technologies, the personal computer and the Internet, is what allowed the Web to be implemented successfully and enabled its wildfire growth, while previous attempts like NLS were doomed to failure as they were conceived before the technological infrastructure to support them had matured.

2.1.3 *The Modern World Wide Web*

Perhaps due to its own anarchic nature, the IETF had produced a multitude of incompatible protocols such as FTP (File Transfer Protocol) and Gopher (Postel and Reynolds 1985; Anklesaria et al. 1993). While protocols could each communicate with other computers over the Internet, there was no universal format to identify information regardless of protocol. One IETF participant, Tim Berners-Lee, had the concept of a “universal information space” which he dubbed the “World Wide Web” (1992). His original proposal to his employer CERN brings his belief in universality to the forefront, “We should work towards a universal linked information system, in which generality and portability are more important than fancy graphics and complex extra facilities” (Berners-Lee 1989). The practical reason for Berners-Lee’s proposal was to connect the tremendous amounts of data generated by physicists at CERN together. Later as he developed his ideas he came into direct contact with Engelbart, who encouraged him to continue his work despite his work being rejected at conferences like ACM Hypertext 1991.¹

In the IETF, Berners-Lee, Fielding, Connolly, Masinter, and others spear-headed the development of URIs (Universal Resource Identifiers), HTML (HyperText Markup Language) and HTTP (HyperText Transfer Protocol). Since by being able to reference anything with equal ease due to URIs, a web of information would form based on “the few basic, common rules of ‘protocol’ that would allow one computer to talk to another, in such a way that when all computers everywhere did it, the system would thrive, not break down” (Berners-Lee 2000). The Web is a virtual space for naming information built on top of the physical infrastructure of the Internet that could move bits around, and it was built through specifications that could be implemented by anyone: “What was often difficult for people to understand about the design was that there was nothing else beyond URIs, HTTP, and HTML. There was no central computer ‘controlling’ the Web, no single network on which these protocols worked, not even an organization anywhere that ‘ran’ the Web. The Web was not a physical ‘thing’ that existed in a certain ‘place.’ It was a ‘space’ in which information could exist” (Berners-Lee 2000).

The very idea of a *universal* information space seemed at least ambitious, if not de facto impossible, to many. The IETF rejected Berners-Lee’s idea that any

¹Personal communication with Berners-Lee.

identification scheme could be universal. In order to get the initiative of the Web off the ground, Berners-Lee surrendered to the IETF and renamed URIs from *Universal Resource Identifiers* (URIs) to *Uniform Resource Locators* (URLs) (Berners-Lee 2000). The Web began growing at a prodigious rate once the employer of Berners-Lee, CERN, released any intellectual property rights they had to the Web and after Mosaic, the first graphical browser, was released. However, browser vendors started adding supposed ‘new features’ that soon led to a ‘lock-in’ where certain sites could only be viewed by one particular corporate browser. These ‘browser wars’ began to fracture the rapidly growing Web into incompatible information spaces, thus nearly defeating the proposed universality of the Web (Berners-Lee 2000).

Berners-Lee in particular realized it was in the long-term interest of the Web to have a new form of standards body that would preserve its universality by allowing corporations and others to have a more structured contribution than possible with the IETF. With the informal position of merit Berners-Lee had as the supposed inventor of the Web (although he freely admits that the invention of the Web was a collective endeavour), he and others constituted the World Wide Web Consortium (W3C), a non-profit dedicated to “leading the Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web” (Jacobs 1999). In the W3C, membership was open to any organization, commercial or non-profit. Unlike the IETF, W3C membership came at a considerable membership fee. The W3C is organized as a strict representative democracy, with each member organization sending one member to the Advisory Committee of the W3C, although decisions technically are always made by the Director, Berners-Lee himself. By opening up a “vendor neutral” space, companies who previously were interested primarily in advancing the technology for their own benefit could be brought to the table. The primary product of the World Wide Web Consortium is a W3C Recommendation, a standard for the Web that is explicitly voted on and endorsed by the W3C membership. W3C Recommendations are thought to be similar to IETF RFCs, with normative force due to the degree of formal verification given via voting by the W3C Membership and a set number of implementations to prove interoperability. A number of W3C Recommendations have become very well known technologies, ranging from the vendor-neutral later versions of HTML (Raggett et al. 1999), which stopped the fracture of the universal information space, to XML, which has become a prominent transfer syntax for many types of data (Bray et al. 1998).

This book will cite W3C Recommendations when appropriate, as these are one of the main normative documents that define the Web. With IETF RFCs, these normative standards collectively define the foundations of the Web. It is by agreement on these standards that the Web functions as a whole. However, the rough-and-ready process of the IETF and the more bureaucratic process of the W3C has led to a terminological confusion that must be sorted in order to grasp the nature of representations on the Web, causing even the most well-meaning of souls to fall into a conceptual swamp of undefined and fuzzy terms. This is true in spades when encountering the hotly-contested term ‘representation.’

2.2 The Terminology of the Web

Can the various technologies that go under the rubric of the World Wide Web be found to have common principles and terminology? This question would at first seem to be shallow, for one could say that any technology that is described by its creators, or even the public at large, can be considered trivially ‘part of the Web.’ To further complicate the matter, the terms the ‘Web’ and the ‘Internet’ are elided together in common parlance, and so are often deployed as synonyms. In a single broad stroke, we can distinguish the Web and the Internet. The Internet is a type of packet-switching network as defined by its use of the TCP/IP protocol. The purpose of the Internet is to get bits from one computer to another. In contrast, the Web is a space of names defined by its usage of URIs. So, the purpose of the Web is the use of URIs for accessing and referring to information. The Web and the Internet are then strictly separable, for the Web, as a space of URIs, could be realized on top of other types of networks that move bits around, much as the same virtual machine can be realized on top of differing physical computers. For example, one could imagine the Web being built on top of a network built on principles different from TCP/IP, such as OSI, an early competitor to the TCP/IP stack of networking protocols (Zimmerman 1980). Likewise, before the Web, there were a number of different protocols with their own naming schemes built upon the Internet like Gopher (Anklesaria et al. 1993).

Is it not presumptuous of us to hope that such an unruly phenomenon such as the Web even has guiding principles? Again we must appeal to the fact that unlike natural language or chemistry, the Web is like other engineered artifacts, created by particular individuals with a purpose, and designed with this purpose in mind. Unlike the case of the proper function of natural language, where natural selection itself will forever remain silent to our questions, the principal designers of the Web are still alive to be questioned in person, and their design rationale is overtly written down on various notes, often scribbled on some of the earliest web-pages of the Web itself. It is generally thought of that the core of the Web consists of the following standards, given in their earliest incarnation: HTTP (Berners-Lee et al. 1996), URI (Berners-Lee 1994a), and HTML (Berners-Lee and Connolly 1993). So the basic protocols and data formats that proved to be successful were the creations of a fairly small number of people, such as Tim Berners-Lee, Roy Fielding, and Dan Connolly.

The primary source for our terminology and principles of Web architecture is a document entitled *The Architecture of the World Wide Web* (AWWW), a W3C Recommendation edited by Ian Jacobs and Norm Walsh to “describe the properties we desire of the Web and the design choices that have been made to achieve them” (Jacobs and Walsh 2004). The AWWW is an attempt to systematize the thinking that went into the design of the Web by some of its primary architects,

and as such is both close to our project and an inspiration.² In particular, AWWW is an exegesis of Tim Berners-Lee’s notes on “Design Issues: Architectural and philosophical points”³ and Roy Fielding’s dissertation “Architectural Styles and the Design of Network-based Software Architectures” (Fielding 2010), often abbreviated as REST. The rationale for the creation of such a document of principles developed organically over the existence of the W3C, as new proposed technologies were sometimes considered to be either informally compliant or non-compliant with Web architecture. When the proponents of some technology were told that their particular technology was not compliant with Web architecture, they would often demand that somewhere there be a description of this elusive Web architecture. The W3C in response set up the Technical Architecture Group (TAG) to “document and build consensus” upon “the underlying principles that should be adhered to by all Web components, whether developed inside or outside W3C,” as stated in its charter.⁴ The TAG also maintains a numbered list of problems (although the numbers are in no way sequential) that attempts to resolve issues in Web architecture by consensus, with the results released as notes called ‘W3C TAG findings,’ which are also referred to in this discussion. The TAG’s only Recommendation at the time of writing is the aforementioned *Architecture of the Web: Volume 1* but it is reasonable to assume that more volumes of *Architecture of the Web* may be produced after enough findings have been accumulated. The W3C TAG’s AWWW is a blend of common-sense and sometimes surprising conclusions about Web architecture that attempts to unify diverse web technologies with a finite set of core design principles, constraints, and good practices (Jacobs and Walsh 2004). However, the terminology of AWWW is often thought to be too informal and ungrounded to use by many, and we attempt to remedy this in the next few chapters by fusing the terminology of Web architecture with our own peculiar brand of philosophical terminology.

To begin our reconstruction of Web architecture, the first task is the definition of terms, as otherwise the technical terminology of the Web can lead to as much misunderstanding as understanding. To cite an extreme example, people coming from communities like the artificial intelligence community use terms like ‘representation’ in a way that is different from those involved in Web architecture. We begin with the terms commonly associated with a typical exemplary Web interaction. For an agent to learn about the *resource* known as the Eiffel Tower in Paris, a person can access its *representation* using its *Uniform Resource Identifier (URI)* <http://www.tour-eiffel.fr/> and retrieve a web-page in the HTML *encoding* whose *content* is the Eiffel Tower using the HTTP *protocol*.

²Although to what extent the Web as it actually exists follows these design choices is still a matter for debate, and it is very clear some of the more important parts of the Web such as the ubiquity of scripting languages, and thus HTML as mobile code, are left unmentioned.

³These unordered personal notes are at: <http://www.w3.org/DesignIssues/>, which we also refer directly to in the course of this chapter.

⁴Quoted from their charter, available on the Web at: <http://www.w3.org/2001/07/19-tag> (last accessed April 20th, 2007).

2.2.1 Protocols

A **protocol** is a convention for transmitting information between two or more agents, a broad definition that encompasses everything from computer protocols like TCP/IP to conventions in natural language like those employed in diplomacy. A protocol often specifies more than just the particular encoding, but also may attempt to specify the interpretation of this encoding and the meaningful behaviour that the sense of the information should engender in an agent. An **agent** is *any thing capable of interacting via a protocol*. These are often called a ‘user agent’ on the Web, and the term covers both web-browsers, humans, web spiders, and even combinations such as humans operating web-browsers. A **payload** is *the information transmitted by a protocol*. Galloway notes that protocols are “the principle of organization native to computers in distributed networks” and that agreement on protocols are necessary for any sort of network to succeed in the acts of communication (2004).⁵ The paradigmatic case of a protocol is TCP/IP, where the payload transmitted is just bits in the body of the message, with the header being used by TCP to ensure the lossless delivery of said bits. TCP/IP transmits strictly an encoding of data as bits and does not force any particular interpretation on the bits; the payload could be a picture of the Eiffel Tower, web-pages about the Eiffel Tower, or just meaningless random bits. All TCP/IP does is move some particular bits from one individual computer to another, and any language that is built on top of the bit-level are strictly outside the bounds of TCP/IP. Since these bits are usually communication with some purpose, the payload of the protocol is almost always an encoding on a level of abstraction above and beyond that of the raw bits themselves.

The Web is based on a **client-server architecture**, meaning that *protocols take the form of a request for information and a response with information*. The **client** is defined as *the agent that is requesting information* and the **server** is defined as *the agent that is responding to the request*. In a protocol, an **endpoint** is *any process that either requests or responds to a protocol*, and so includes both client and servers. The client is often called a **user-agent** since it is the user of the Web. A user-agent may be anything from a web-browser to some sort of automated reasoning engine that is working on behalf of another agent, often the specifically human user. The main protocol in this exposition will be the **HyperText Transfer Protocol** (HTTP), as most recently defined by IETF RFC 2616 (Fielding et al. 1999). HTTP is a protocol originally intended for the transfer of hypertext documents, although its now ubiquitous nature often lets it be used for the transfer of almost any encoding over the Web, such as its use to transfer XML-based SOAP (originally the *Simple Object Access Protocol*) messages in Web Services (Box et al. 2000). HTTP consists of sending a **method**, *a request for a certain type of response from a user-agent to the server*, including information that may change the state of the server.

⁵Although unlike Galloway, who descends into a sort of postmodern paranoia of protocols, we recognize them as the very conditions of collectivity.

Fig. 2.1 An HTTP request from a client

```
GET /index.html HTTP/1.0
User-Agent: Mozilla/5.0
Accept: */*
Host: www.example.org
Connection: Keep-Alive
```

These methods have a list of *headers* that *specify some information that may be used by the server to determine the response*. The *request* is the method used by the agent and the headers, along with a blank line and an optional message body.

The methods in HTTP are HEAD, GET, POST, PUT, DELETE, TRACE, OPTIONS, and CONNECT. We will only be concerned with the most frequently used HTTP method, GET. GET is informally considered ‘commitment-free,’ which means that the method has no side effects for either the user-agent or the server, besides the receiving of the response (Berners-Lee et al. 1996). So a GET method should not be used to change the state of a user-agent, such as charging someone for buying a plane ticket to Paris. To change the state of the information on the server or the user-agent, either PUT (for uploading data directly to the server) or POST (for transferring data to the server that will require additional processing, such as when one fills in a HTML form) should be used. A sample request to <http://www.example.org> from a Web browser user-agent is given in Fig. 2.1.

The first part of an HTTP response from the server then consists of an HTTP *status code* which is *one of a finite number of codes which gives the user-agent information about the server’s HTTP response itself*. The two most known status codes are HTTP 200, which means that the request was successful, or 404, which means the user-agent asked for data that was not found on the server. The first digit of the status code indicates what general class of response it is. For example, the 200 series (2xx) response codes mean a successful request, although 206 means partial success. The 4xx codes indicate that the user-agent asked for a request that the server could not fulfill, while 1xx is informational, 3xx is redirectional, and 5xx means server error. After the status codes there is an *HTTP entity* which is “the information transferred as the payload of a request or response” (Fielding et al. 1999). This technical use of the word ‘entity’ should be distinguished from our earlier use of the term ‘entity’ like the Eiffel Tower which can only be realized by the thing itself, not in another realization. In order to do so, we will take care to preface the protocol name ‘HTTP’ before any ‘HTTP entity,’ while the term ‘entity’ by itself refers to the philosophical notion of an entity. An HTTP entity consists of “entity-header fields and...an entity-body” (Fielding et al. 1999) An *HTTP response* consists of *the combination of the status code and the HTTP entity*. These responses from the server can include an additional header, which specifies the date and last modified date as well as optional information that can determine if the desired representation is in the cache and the content-type of the representation. A sample HTTP response to the previous example request, excluding the HTTP entity-body, is given in Fig. 2.2.

In the HTTP response, an HTTP entity body is returned. The encoding of the HTTP entity body is given by the HTTP entity header fields that specify its

Fig. 2.2 An HTTP response from a server

```
HTTP/1.1 200 OK
Date: Wed, 16 Apr 2008 14:12:09 GMT
Server: Apache/2.2.4 (Fedora)
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=ISO-8859-1
Content-Language: fr
```

Content-type and Content-language. These are both considered different languages, as a single web-page can be composed in multiple languages, such as the text being given in English with various formatting given in HTML. Every HTTP entity body should have its particular encoding specified by the Content-type. *The formal languages that can be explicitly given in a response or request in HTTP are called **content types**.* In the example response, based on the header that the content type is text/html a user-agent can interpret ('display as a web-page') the encoding of the HTTP entity body as HTML. Since the same encoding can theoretically represent many different languages besides HTML, a user-agent can only know definitely how to process a message through the content type. If no content type is provided, the agent can guess the content type through various heuristics including looking at the bytes themselves, a process informally called *sniffing*. A user-agent can specify what media types they (can) prefer, so that a web-server that can only present JPEG images can specify this by also asking for the content type image/jpeg in the request.

Content-types in HTTP were later generalized as 'Internet Media Types' so they could be applied with any Internet protocol, not just HTTP and MIME (*Multimedia Internet Message Extensions*, an e-mail protocol) (Postel 1994). A **media type** consists of a *two-part scheme that separates the type and a subtype of an encoding*, with a slash indicating the distinction. Internet media types are centrally registered with IANA,⁶ although certain 'experimental' media types (those beginning with 'x-') can be created in a decentralized manner (Postel 1994). A central registry of media types guarantees the interoperability of the Web, although increasingly new media-types are dependent on extensions to specific applications (plug-ins) in order to run. Support for everything from new markup languages to programming languages such as Javascript can be declared via support of its media type.

To move from concrete bits to abstract definitions, a protocol can be defined and implemented in many different types of way. In the early ARPANet, the first wide-area network and foundation of the Internet, the protocol was 'hard-wired' in the hardware of the Interface Message Processor (IMP), a separate machine attached to computers in order to interface them with ARPANet (Hafner and Lyons 1996). As more and more networks multiplied, these heterogeneous networks began using different protocols. While the invention of TCP/IP let these heterogeneous networks communicate, TCP/IP does not interpret messages beyond bits. Further

⁶At <http://www.iana.org/assignments/media-types/>.

protocols are built on top of TCP/IP, such as FTP (File Transfer Protocol) for the retrieval of files (Postel and Reynolds 1985), Gopher for the retrieval of documents (Anklesaria et al. 1993), and SMTP (Simple Mail Transfer Protocol) for the transfer of mail (Postel 1982). Since one computer might hold many different kinds of information, IP addresses were not enough as they only identified where a particular device was on the network. Thus each protocol created its own naming scheme to allow it to identify and access things on a more fine-grained level than IP addresses. Furthermore, each of these protocols was often associated (via registration with a governing body like IANA, the *Internet Assigned Numbers Authority*) with particular ports, such that port 25 was used by SMTP and port 70 by Gopher. With this explosion of protocols and naming schemes, each Internet application was its own ‘walled garden.’ Names created using a particular protocol were incapable of being used outside the original protocol, until the advent of the naming scheme of the Web (Berners-Lee 2000).

2.2.2 Information Encoding and Content

There is a relationship between a server sending a message – such as a webpage about the Eiffel Tower – to a client in response to an HTTP request and certain notions from information theory, however hazy and qualitative. To phrase informally, **information** is *whatever regularities held in common between a source and a receiver* (Shannon and Weaver 1963). Note that the source and receiver do not have to be spatially separate, but can also be temporally separate, and thus the notion of a self-contained ‘message’ resembling a postcard being sent between sender and receiver is incomplete if not incorrect.⁷ To have something in common means to share the same regularities, e.g. parcels of time and space that cannot be distinguished at a given level of abstraction. This definition correlates with information being the inverse of the amount of ‘noise’ or randomness in a system, and the amount of information being equivalent to a reduction in uncertainty. It is precisely this preservation or failure to preserve information that can be thought of as sending of a *message* between the source and the receiver over a channel, where the channel is over time, space, and – most likely – both. *Whether or not the information is preserved over time or space is due to the properties of a physical substrate* known as the **channel**. So in our example, the channel is the fiber-optic or copper wires that must accurately carry the voltages which the bits consist of. The **message** is the *physical thing that realizes the regularities of the information due to its local characteristics*, which in this case would be particular patterns of bits being preserved over multiple channels as they are popped from an electro-magnetic hard-disk on a server to fibre-optic then over the air via wireless and finally back to the

⁷Imagine that your eye color not changing is a message from yourself at 10 years old to yourself at 70!

electric charges stored in memory chips in a client device, such as a web browser on a mobile phone. These messages are often called the *realization* of some abstract informational content.

Already, information reveals itself to be not just a singular thing, but something that exists at multiple levels: How do the bits become a message in HTTP? In particular, we are interested in the distinction in information between content and encoding. Here our vague analogy with Shannon's information theory fails, as Shannon's theory deals with finding the optimal encoding and size of channel so that the message can be guaranteed to get from the sender to the receiver, which in our case is taken care of by the clever behavior of the TCP/IP protocol operating over a variety of computational devices (Shannon and Weaver 1963). Yet, how can an encoding be distinguished from the content of information itself in a particular HTTP message? Let's go back to bits by leaning on aesthetic theory of all things; art critic and philosopher Nelson Goodman defines a *mark* as a *physical characteristic* ranging from marks on paper one can use to discern alphabetic characters to ranges of voltage that can be thought of as bits (1968). To be reliable in conveying information, an encoding should be physically 'differentiable' and thus maintain what Goodman calls 'character indifference' so that (at least within some context) each character (as in 'characteristic') can not be mistaken for another character. One cannot reconstruct a message in bits if one cannot tell apart 1 and 0, much as one cannot reconstruct a HTML web-page if one cannot tell the various characters in text apart. So, an *encoding* is a *set of precise regularities that can be realized by the message*. Thus, one can think of multiple levels of encoding, with the very basic encoding of bits being handled by the protocol TCP/IP, and then the protocol HTTP handling higher-level encodings in textual encodings such as HTML.

Unfortunately, we are not out of the conceptual thicket yet; there is more to information than encoding. Shannon's theory does not explain the notion of information fully, since giving someone the number of bits that a message contains does not tell the receiver *what* information is encoded. Shannon explicitly states, "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem" (1963). He is correct, at least for his particular engineering problem. However, Shannon's use of the term 'information' is for our purposes the same as the 'encoding' of information, but a more fully-fledged notion of information is needed. Many intuitions about the notion of information have to deal with not only how the information is encoded or how to encode it, but what a particular message is about, the *content* of an information-bearing message.⁸ 'Content' is a term we adopt from Israel and Perry,

⁸An example of the distinguishment between content and encoding: Imagine Daniel sending Amy a secret message about which one of her co-employees won a trip to the Eiffel Tower. Just determining that a single employee out of 8 won the lottery requires at least a 3 bit encoding

as opposed to the more confusing term ‘semantic information’ as employed by Floridi and Dretske (Israel and Perry 1990; Dretske 1981; Floridi 2004). One of the first attempts to formulate a theory of informational content was due to Carnap and Bar-Hillel (1952). Their theory attempted to bind a theory of content closely to first-order predicate logic, and so while their “theory lies explicitly and wholly within semantics” they explicitly do not address “the information which the sender intended to convey by transmitting a certain message nor about the information a receiver obtained with a certain message,” since they believed these notions could eventually be derived from their formal apparatus (Carnap and Bar-Hillel 1952). Their overly restrictive notion of the content of information as logic did not gain widespread traction, and neither did other attempts to develop alternative theories of information such as that of Donald McKay (1955). In contrast, Dretske’s *semantic theory of information* defines the notion of content to be compatible with Shannon’s information theory, and his notions have gained some traction within the philosophical community (Dretske 1981). To him, the content of a message and the amount of information – the number of bits an encoding would require – are different, for “saying ‘There is a gnu in my backyard’ does not have more content than the utterance ‘There is a dog in my backyard’ since the former is, statistically, less probable” (Dretske 1981). According to Shannon, there is more information in the former case precisely because it is less likely than the latter (Dretske 1981). So while information that is less frequent may require a larger number of bits in encoding, the content of information should be viewed as to some extent separable if compatible with Shannon’s information theory, since otherwise one is led to the “absurd view that among competent speakers of language, gibberish has more meaning than semantic discourse because it is much less frequent” (Dretske 1981). Simply put, Shannon and Dretske are talking about distinct notions that should be separated, the notions of encoding and content respectively.

Is there a way to precisely define the content of a message? Dretske defines the content of information as “a signal r carries the information that s is F when the conditional probability of s ’s being F , given r (and k) is 1 (but, given k alone, less than 1). k is the knowledge of the receiver” (1981). To simplify, the *content* of any

and does not tell Amy (the receiver) which employee in particular won the lottery. Shannon’s theory only measures how many bits are needed to tell Amy precisely who won. After all, the false message that her office-mate Sandro won a trip to Paris is also 3 bits. Yet content is not independent of the encoding, for content is conveyed by virtue of a particular encoding and a particular encoding imposes constraints on what content can be sent (Shannon and Weaver 1963). Let’s imagine that Daniel is using a code of bits specially designed for this problem, rather than natural language, to tell Amy who won the free plane ticket to Paris. The content of the encoding 001 could be yet another co-employee Ralph while the content of the encoding 010 could be Sandro. If there are only two possible bits of information and all eight employees need one unique encoding, Daniel cannot send a message specifying which friend got the trip since there aren’t enough options in the encodings to go round. An encoding of at least 3 bits is needed to give each employee a unique encoding. If 01 has the content that ‘either Sandro or Ralph won the ticket’ the message has not been successfully transferred if the purpose of the message is to tell Amy *precisely* which employee won the ticket.

information-bearing message is *whatever is held in common between the source and the receiver as a result of the conveyance of a particular message*. While this is similar to our definition of information itself, it is different. The content is whatever is shared in common as a result of a *particular* message, such as the conveyance of the sentence ‘The Eiffel Tower is 300m high.’ The content of a message is called the “facts” by Dretske, (*F*). This content is conveyed from the source (*s*) successfully to the receiver (*r*) when the content can be used by the receiver with certainty, *and* that before the receipt of the message the receiver was not certain of that particular content. Daniel can only successfully convey the content that ‘Ralph won a trip to Paris’ if before receiving the message Amy does not know ‘Ralph won a trip to Paris’ and after receiving the message Amy does know that fact. Dretske himself notes that information “does not mean that a signal must tell us everything about a source to tell us something,” it just has to tell enough so that the receiver is now certain about the content within the domain (1981). Millikan rightfully notes that Dretske states his definition too strongly, for this probability of 1 is just an approximation of a statistically “good bet” indexed to some domain where the information was learned to be recognized (2004). For example, lightening carries the content that “a thunderstorm is nearby” in rainy climes but in an arid prairie lightning can convey a dust-storm. However, often the reverse is true, as the same content is carried by messages in different encodings, like a web-page about the Eiffel Tower being encoded in either English or French. These notions of encoding and content are not strictly separable, which is why they together compose the notion of information. An updated famous maxim of Hegel could be applied: for information, there is no encoding without content, and no content without encoding (1959).

The relationship of an encoding to its content, is an **interpretation**. The interpretation ‘fills’ in the necessary background left out of the encoding, and maps the encoding to some content. In our previous example using binary digits as an encoding scheme, a mapping could be made between the encoding 001 to the content of the Eiffel Tower while the encoding 010 could be mapped to the content of the Washington Monument. When the word ‘interpretation’ is used as a noun, we mean the content given by a particular relationship between an agent and an encoding, i.e. the interpretation. Usual definitions of “interpretation” tend to conflate these issues. In formal semantics, the word “interpretation” often can be used either in the sense of “an interpretation structure, which is a ‘possible world’ considered as something independent of any particular vocabulary” (and so any agent) or “an interpretation mapping from a vocabulary into the structure” or as shorthand for both (Hayes 2004). The difference in use of the term seems somewhat divided by fields. For example, computational linguistics often use “interpretation” to mean what Hayes called the “interpretation structure.” In contrast, we use the term ‘interpretation’ to mean what Hayes called the “interpretation mapping,” reserving the word ‘content’ for the “interpretation structure” or structures selected by a particular agent in relationship to some encoding. Also, this quick aside into matters of interpretation does not explicitly take on a formal definition of interpretation as done in model theory, although our general definition has been designed to be compatible with model-theoretic and other formal approaches to interpretation.

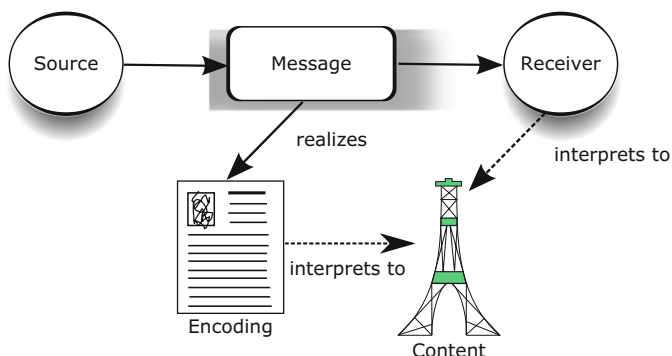


Fig. 2.3 Information, encoding, content

These terms are all illustrated in Fig. 2.3. A source is sending a receiver a message. The information-bearing message realizes some particular encoding such as a few sentences in English and a picture of the Eiffel Tower, and the content of the message can be interpreted to be about the Eiffel Tower.

The encoding and content of information do not in general come in self-contained bundles, with each encoding being interpreted to some free-standing propositional content. Instead, encodings and content come in entire interlocking informational systems. One feature of these systems is that encodings are layered inside of each other and content is also layered upon other content. The perfect example would be an English sentence in an e-mail message, where a series of bits are used to encode the letters of the alphabet, and the alphabet is then used to encode words. Likewise, the content of a sentence may depend on the content of the words in the sentence. When this happens, one is no longer dealing with a simple message, but some form of language. A **language** can be defined as *a system in which information is related to other information systematically*. In a language, this is a relationship between how the encoding of some information can change the interpretation of other encodings. Messages always have encodings, and usually these encodings are part of languages. To be more brief, information is *encoded in languages*. The relationships between encodings and content are usually taken to be based on some form of (not necessarily formalizable or even understood) rules. If one is referring to *a system in which the encoding of information is related to each other systematically*, then one is talking about the **syntax** of a language. If one is referring to *a system in which the content of information is related to each other systematically*, then one is referring to the **semantics** of the language. The lower-level of a language can be **terms**, *regularities in marks*, that may or may not have their own interpretation, such as the words or alphabet. Any combination of terms that is valid according to the language's syntax is a **sentence** (sometimes an 'expression') in the language, and any combination of terms that has an interpretation to content according to the language's semantics is a **statement** in the language.

Particular encodings and content are then accepted by or considered valid by the syntax and semantics of a language respectively (and thus the normative importance of standardization on the Web in determining these criteria). Also, we do not restrict our use of the word ‘language’ to primarily linguistic forms, but use the term ‘language’ for anything where there is a systematic relationship between syntax and (even an informal) semantics. For example HTML is a language for mapping a set of textual tags to renderings of bits on a screen in a web browser. One principle used in the study of languages, attributed to Frege, is the principle of **compositionality**, where *the content of a sentence is related systematically to terms in which it is composed*. Indeed, while the debate is still out if human languages are truly compositional (Dowty 2007), computer languages almost always are compositional. In English, the content of the sentence such as ‘Tim has a plane ticket to Paris so he should go to the airport!’ can then be composed from the more elementary content of the sub-statements, such as ‘Tim has a plane ticket’ which in turn has its content impacted by words such as ‘Tim’ and ‘ticket.’ The argument about whether sentences, words, or clauses are the minimal building block of content is beyond our scope. Do note one result of the distinction between encoding and content is that sentences that are accepted by the syntax (encoding) of a language, such as Chomsky’s famous “Colorless green ideas sleep furiously” may have no obvious interpretation (to content) outside of the pragmatics of Chomsky’s particular exposition (1957).

2.2.3 Uniform Resource Identifiers

The World Wide Web is defined by the AWWW as “an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI)” (Jacobs and Walsh 2004). This naming scheme, not any particular language like HTML, is the primary identifying characteristic of the Web. URIs arose from a need to organize the “many protocols and systems for document search and retrieval” that were in use on the Internet, especially considering that “many more protocols or refinements of existing protocols are to be expected in a field whose expansion is explosive” (Berners-Lee 1994a). Despite the “plethora of protocols and data formats,” if any system was “to achieve global search and readership of documents across differing computing platforms,” gateways that can “allow global access” should “remain possible” (Berners-Lee 1994a). The obvious answer was to consider all data on the Internet to be a single space of names with global scope.

URIs accomplish their universality over protocols by moving *all the information used by the protocol within the name itself*. The information needed to identify any protocol-specific information is all specified in the name itself: the name of the protocol, the port used by the protocol, any queries the protocol is responding to, and the hierarchical structure used by the protocol. The Web is then first and foremost a naming initiative “to encode the names and addresses of objects on the Internet”

rather than anything to do with hypertext (Berners-Lee 1994a). The notion of a URI can be viewed as a “meta-name,” a name which takes the existing protocol-specific Internet addresses and wraps them in the name itself, a process analogous to reflection in programming languages (Smith 1984). Instead of limiting itself to only existing protocols, the URI scheme also abstracts away from any particular set of protocols, so that even protocols in the future or non-Internet protocols can be given a URI; “the web is considered to include objects accessed using an extendable number of protocols, existing, invented for the web itself, or to be invented in the future” (Berners-Lee 1994a).

One could question why one would want to name information outside the context of a particular protocol. The benefit is that the use of URIs “allows different types of resource identifiers to be used in the same context, even when the mechanisms used to access those resources may differ” (Berners-Lee et al. 2005). This is an advantage precisely because it “allows the identifiers to be reused in many different contexts, thus permitting new applications or protocols to leverage a pre-existing, large, and widely used set of resource identifiers” (Berners-Lee et al. 2005). This ability to access with a single naming convention the immense amount of data on the entire Internet gives an application such as the ubiquitous Web browser a vast advantage over an application that can only consume application-specific information.

Although the full syntax in Backus-Naur form is given in IETF RFC 3986 (Berners-Lee et al. 2005), a URI can be given as the regular expression `URI=[scheme ":"][hierarchical component]*["?" query]?["#" fragment]?`. First, a *scheme* is a name of the protocol or other naming convention used in the URI. Note that the scheme of a URI does not determine the protocol that a user-agent has to employ to use the URI. For example, a HTTP request may be used on <ftp://www.example.org>. The scheme of a URI merely indicates a preferred protocol for use with the URI. A *hierarchical component* is the left to right dominant component of the URI that syntactically identifies the resource. URIs are federated, insofar as each scheme identifies the syntax of its hierarchical component. For example, with HTTP the hierarchical component is given by `[authority] [//] [":" port]? ["/" path component]*`. The *authority* is a name that is usually a domain name, naming authority, or a raw IP address, and so is often the name of the server. However, in URI schemes like `tel` for telephone numbers, there is no notion of an authority in the scheme. The hierarchical component contains special reserved characters that are in HTTP characters such as the backslash for locations as in a file system. For *absolute URIs*, there must be a single scheme and the scheme and the hierarchical component must together identify a resource such as <http://www.example.com:80/monument/EiffelTower> in HTTP, which signals port 80 of the authority www.example.com with the path component `/monument/EiffelTower`. The port authority is usually left out, and assumed to be 80 by HTTP-enabled clients. Interestingly enough there are also *relative URIs* in some schemes like HTTP, where the path component itself is enough to identify a resource within certain contexts, like that of a web-page. This is because the scheme and authority itself may have substituted some special characters that serve as indexical expressions, such as ‘.’ for the current

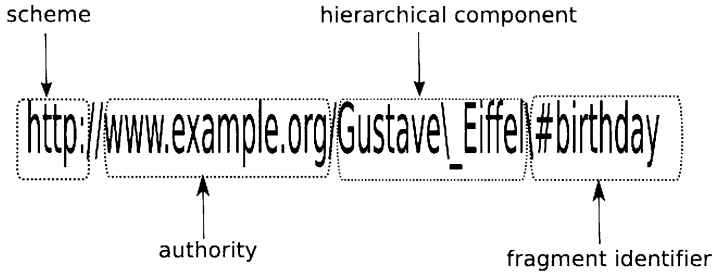


Fig. 2.4 An example URI, with components labelled

place in the path component and ‘..’ as the previous level in the path component. So, `../EiffelTower` is a perfectly acceptable relative URI. Relative URIs have a straightforward translation into absolute URIs, and it is trivial to compare absolute URIs for equality (Berners-Lee et al. 2005).

The ‘hash’ (#) and ‘question mark’ (?) are special characters at the end of a URI. The question mark denotes ‘query string.’ The ‘query string’ allows for the parameterization of the HTTP request, typically in the cases where the HTTP response is created dynamically in response to specifics in the HTTP request. The ‘hash’ traditionally declares a *fragment identifier*, which identifies fragments of a hypertext document but according to the TAG, it can also identify a “secondary resource,” which is defined as “some portion or subset of the primary resource, some view on representations of the primary resource, or some other resource defined or described by those representations” where the “primary resource” is the resource identified by the URI without reference to either a hash or question mark (Jacobs and Walsh 2004). The fragment identifier (specified by a ‘hash’ followed by some string of characters) is stripped off for the request to the server, and handled on the client side. Often the fragment identifier causes the local client to go to a particular part of the accessed HTTP entity. If there was a web-page about Gustave Eiffel, its introductory paragraph could be identified with the URI <http://www.example.com/EiffelTower#intro>. Figure 2.4 examines a sample URI, http://example.org/Gustave_Eiffel#birthday:

The first feature of URIs, the most noticeable in comparison to IP addresses, is that they can be human-readable, although they do not have to be. As an idiom goes, URIs can be ‘written on the side of a bus.’ URIs can then have an interpretation due to their use of terms from natural language, such as <http://www.whitehouse.gov> referring to the White House or the entire executive branch of the United States government. Yet it is considered by the W3C TAG to be bad practice for any agent to depend on whatever information they can glean from the URI itself, since to a machine the natural language terms used by the URI have no interpretation. For an agent, all URIs are opaque, with each URI being just a string of characters that can be used to either refer to or access information, and so syntactically it can only be checked for equality with other URIs and nothing more. This is captured well by the good practice of *URI opacity*, which states that “agents making use of URIs

should not attempt to infer properties of the referenced resource” (Jacobs and Walsh 2004). So, just because a URI says <http://www.eiffel-tower.com> does not mean it will not lead one to a web-page trying to sell one cheap trinkets and snake oil, as most users of the Web know. Second, a URI has an owner. The *owner is the agent that is accountable for a URI*. Interestingly enough, the domain name system that assigns control of domain names in URIs is a legally-binding techno-social system, and thus to some extent a complex notion of accountability for the name is built into URIs. Usually for URI schemes such as HTTP, where the hierarchical component begins with an authority, the owner of the URI is simply whoever controls that authority. In HTTP, since URIs can delegate their relative components to other users, the owner can also be considered the agent that has the ability to create and alter the information accessible from the URI, not just the owner of the authority. Each scheme should in theory specify what ownership of a URI means in context of the particular scheme.

2.2.4 Resources

While we have explained how a URI is formed, we have yet to define what a URI is. To inspect the acronym itself, a Uniform Resource Identifier (URI) is an identifier for a ‘resource.’ Yet this does not solve any terminological woes, for the term ‘resource’ is undefined in the earliest specification for “Universal Resource Identifiers” (Berners-Lee 1994a). Berners-Lee has remarked that one of the best things about resources is that for so long he never had to define them (Berners-Lee 2000). Eventually Berners-Lee attempted to define a resource as “anything that has an identity” (Berners-Lee et al. 1998). Other specifications were slightly more detailed, with Roy Fielding, one of the editors of HTTP, defining (apparently without the notice of Berners-Lee) a resource as “a network data object or service” (Fielding et al. 1999). However, at some later point Berners-Lee decided to generalize this notion, and in some of his later works on defining this slippery notion of ‘resource,’ Berners-Lee was careful not to define a resource only as information that is accessible via the Web, since not only may resources be “electronic documents” and “images” but also “not all resources are network retrievable; e.g., human beings, corporations, and bound books in a library” (Berners-Lee et al. 1998). Also, resources do not have to be singular but can be a “collection of other resources” (Berners-Lee et al. 1998).

Resources are not only concrete messages or sets of possible messages at a given temporal junction, but are a looser category that includes individuals changing over time, as “resources are further carefully defined to be information that may change over time, such as a service for today’s weather report for Los Angeles” (Berners-Lee et al. 1998). Obviously, a web-page with “today’s weather report” is going to change its content over time, so what is it that unites the notion of a resource over time? The URI specification defines this tentatively as a “conceptual mapping” (presumably located in the head of an individual creating the representations for

the resource) such that “the resource is the conceptual mapping to an entity or set of entities, not necessarily the entity which corresponds to that mapping at any particular instance in time. Thus, a resource can remain constant even when its content – the entities to which it currently corresponds – changes over time, provided that the conceptual mapping is not changed in the process” (Berners-Lee et al. 1998). This obviously begs an important question: If resources are identified as conceptual mappings in the head of an individual(s), then how does an agent know, given a URI, what the resource is? Is it our conceptual mapping, or the conceptual mapping of the owner, or some consensus conceptual mapping? The latest version of the URI specification deletes the confusing jargon of “conceptual mappings” and instead re-iterates that URIs can also be things above and beyond concrete individuals, for “abstract concepts can be resources, such as the operators and operands of a mathematical equation” (Berners-Lee et al. 2005). After providing a few telling examples of precisely how wide the notion of a resource is, the URI specification finally ties the notion of resource directly to the act of identification given by a URI, for “this specification does not limit the scope of what might be a resource; rather, the term ‘resource’ is used in a general sense for whatever might be identified by a URI” (Berners-Lee et al. 2005). Although this definition seems at best tautological, the intent should be clear. A **resource** is *any thing capable of being content*, or in other words, an ‘identity’ in a language. Since a sense is not bound to particular encoding, in practice within certain protocols that allow access to information, *a resource is typically not a particular encoding of some content but some content that can be given by many encodings*. To rephrase in terms of sense, *the URI identifies content on a level of abstraction, not the encoding of the content*. So, a URI identifies the ‘content’ of the Eiffel Tower, not just a particular web-page which is subject to change. However, there is nothing to forbid someone from identifying a particular encoding of information with its own URI and resource. For example, one could also have a distinct URI for a web-page about the Eiffel Tower in English, or a web-page about the Eiffel Tower in English in HTML. In other words, a resource can be given *multiple URIs*, each corresponding to a different encoding or even different levels of abstraction. Furthermore, due to the decentralized nature of URIs, often different agents create *multiple URIs for the same content*, which are then called in Web architecture **co-referential URIs**.

We illustrate these distinctions in a typical HTTP interaction in Fig. 2.5, where an agent via a web browser wants to access some information about the Eiffel Tower via its URI. While on a level of abstraction a protocol allows a user-agent to identify some resource, what the user-agent usually accesses concretely is some realization of that resource in a particular encoding, such as a web-page in HTML or a picture in the JPEG language (Pennebaker and Mitchell 1992). In our example, the URI is resolved using the domain name system to an IP address of a concrete server, which then transmits to the user-agent some concrete bits that realizes the resource, i.e. that can be interpreted to the sense identified by the URI. In this example, all the interactions are local, since the web-page *encodes* the content of the resource. This HTTP entity can then be interpreted by a browser as a rendering on the screen of Ralph’s browser. Note this is a simplified example, as some status codes like 307

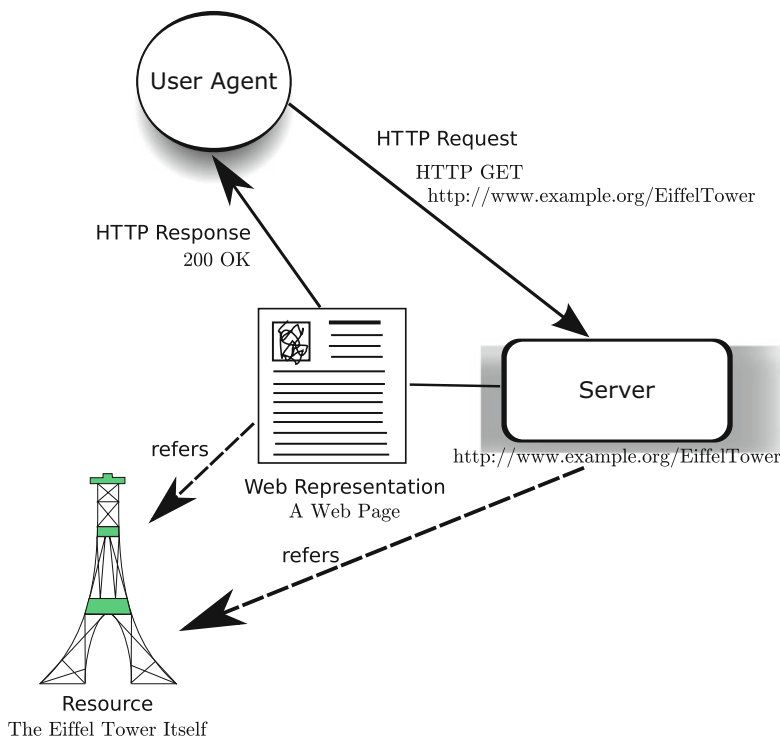


Fig. 2.5 A user agent accessing a resource

may cause a redirection to yet another URI and so another server, and so on possibly multiple times, until an HTTP entity may finally be retrieved.

One of the most confusing issues of the Web is that a URI does not necessarily retrieve a single HTTP entity, but can retrieve multiple HTTP entities. This leads to a surprising and little-known aspect of Web architecture known as content negotiation. **Content Negotiation** is a mechanism defined in a protocol that makes it possible to respond to a request with different Web representations of the same resource depending on the preference of the user-agent. This is because information may have multiple encodings in different languages that all encode the same sense, and thus the same resource which should have a singular URI. A ‘representation’ on the Web is then just “an entity that is subject to content negotiation” (Fielding et al. 1999). Historically, the term ‘representation’ on the Web was originally defined in HTML as “the encoding of information for interchange” (Berners-Lee and Connolly 1993). A later definition given by the W3C did not mention content negotiation explicitly, defining a representation on the Web as just “data that encodes information about resource state” (Jacobs and Walsh 2004). To descend further into a conceptual swamp, ‘representation’ is one of the most confusing terms in Web architecture, as the term ‘representation’ is used differently across philosophy.

In order to distinguish the technical use of the term ‘representation’ within Web architecture from the standard philosophical use of the term ‘representation,’ we shall use the term ‘Web representation’ to distinguish it from the ordinary use of the term ‘representation’ as given earlier in Sect. 2.2.6. A **Web representation** is the *encoding of the content given by a resource given in response to a request that is subject to content negotiation*, which must then include any headers that specify an interpretation, such as character encoding and media type. So a Web representation can be considered to have *two* distinct components, and the headers such as the media type that lets us interpret the encoding, and the payload itself, which is the encoding of the state of the resource at a given point in time (i.e. the HTML itself). So, **web-pages** are *web representations given in HTML*. Web resources can be considered resources that under ‘normal’ conditions result in the delivery of web-pages.

Our typical Web transaction, as given earlier in Fig. 2.5, can become more complex due to this possible separation between content and encoding on the Web. Different kinds of Web representations can be specified by user-agents as preferred or acceptable, based on the preferences of its users or its capabilities, as given in HTTP. The owner of a web-site about the Eiffel Tower decides to host a resource for images of the Eiffel Tower. The owner creates a URI for this resource, <http://www.eiffeltower.example.org/image>. Since a single URI is used, the sense (the depiction) that is encoded in either SVG or JPEG is the same, namely that of an image of the Eiffel Tower. That is, there are two distinct encodings of the image of the Eiffel Tower available on a server in two different iconic languages, one in a vector graphic language known as SVG and one in a bitmap language known as JPEG (Ferraiolo 2002; Pennebaker and Mitchell 1992). These encodings are rendered identically on the screen for the user. If a web-browser only accepted JPEG images and not SVG images, the browser could request a JPEG by sending a request for `Accept: image/jpeg` in the headers. Ideally, the server would then return the JPEG-encoded image with the HTTP entity header `Content-Type: image/jpeg`. Had the browser wished to accept the SVG picture as well, it could have put `Accept: image/jpeg, image/svg+xml` and received the SVG version. In Fig. 2.6, the user agent specifies its preferred media type as `image/jpeg`. So, both the SVG and JPEG images are Web representations of the same resource, an image of the Eiffel Tower, since both the SVG and JPEG information realize the same information, albeit using different languages for encoding. Since a single resource is identified by the same URI <http://www.example.org/EiffelTower/image>, different user-agents can get a Web representation of the resource in a language they can interpret, even if they cannot all interpret the same language. In Web architecture, content negotiation can also be deployed over not only differing computational languages such as JPG or SVG, but differing natural languages, as the same content can be encoded in different natural languages such as French and English. An agent could request the description about the Eiffel Tower from its URI and set the preferred media type to ‘`Accept-Language: fr`’ so that they receive a French version of the web-page as opposed to an English version. Or they could set their preferred language as English but by using ‘`Accept-Language: en`.’

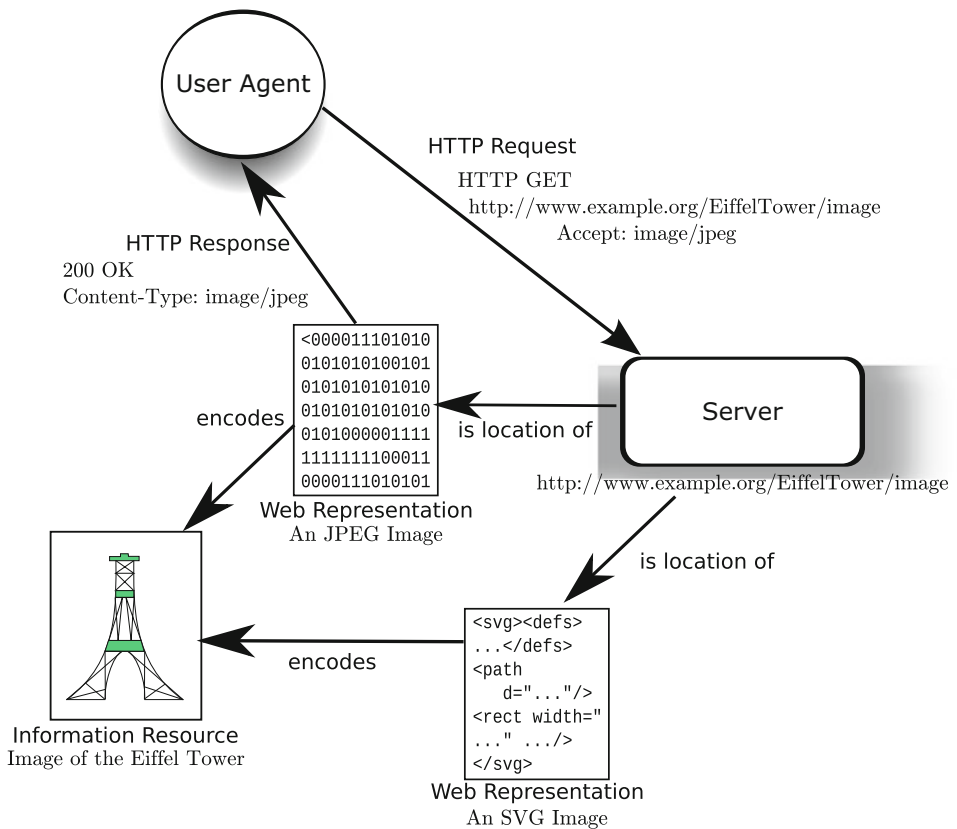


Fig. 2.6 A user agent accessing a resource using content negotiation

The preferences specified in the headers are not mandatory for the server to follow, the server may only have a French version of the resource available, and so send the agent a French version of the description, encoded in HTML or some other formal language, regardless of their preference.⁹ Figure 2.6 shows that the Web representations are distinct from the resource, even if the Web representations are bound together by realizing the same information given by a resource, since accessing a resource via a single URI can return *different* Web representations depending on content negotiation.

⁹It is well-known there are some words in French that are difficult if not impossible to translate into English, such as ‘frileusement.’ Indeed, saying that one natural language encodes the same content as another natural language is akin to hubris in the general case. If this is the case, then it is perfectly reasonable to establish different resources and so URIs for the French and English language encodings of the resource, such as <http://www.eiffeltower.example.org/francais> and <http://www.eiffeltower.example.org/english>. In fact, if one believes the same image cannot be truly expressed by both SVG and JPEG image formats, one could give them distinct URIs as well.

The only architectural constraint that connects Web representations to resources is that they are retrieved by the same URI. So one could imagine a resource with a URI called <http://www.example.org/Moon>, that upon accessing using English as the preferred language would provide a web-page with a picture of the moon, and upon accessing with something other than English as the preferred language would provide a picture of blue cheese. While this seems odd, this situation is definitely possible. What binds Web representations to a resource? Is a resource *really* just a random bag of Web representations? Remember that the answer is that the Web representations should have the same *content* regardless of their particular encoding if it is accessible from the same URI, where content is defined by an appeal to Dretske's semantic theory of information (Dretske 1981). To recall, Dretske's definition of semantic information, "a signal r carries the information that s is F when the conditional probability of s 's being F , given r (and k) is 1 (but, given k alone, less than 1). k is the knowledge of the receiver" (Dretske 1981). We can then consider the signal r to be a Web representation, with s being a resource and the receiver being the user-agent. However, instead of some fact F about the resource, we want an interpretation of the Web representation by *different* user-agents to be to the same content.¹⁰ From a purely normative viewpoint in terms of relevant IETF and W3C standards, it is left to the owner to determine whether or not two Web representations are equivalent and so can be hosted using content negotiation at the same URI. The key to content negotiation is that the owner of a URI never knows what the capabilities of the user-agent are, and therefore what natural and formal languages are supported by it. This is analogous to what Dretske calls the "knowledge" or k of the receiver (1981). The responsibility of the owner of a URI should be, in order to share their resource by as many user-agents as possible, to provide as many Web representations in a variety of formats as they believe are reasonably necessary. So, the owner of the URI for a website about the Eiffel Tower may wish to have a number of Web representations in a wide variety of languages and formats. By failing to provide a Web representation in Spanish, they prevent speakers of only Spanish from accessing their resource. Since the maintainer of a resource cannot reasonably be expected to predict the capabilities of all possible user-agents, the maintainer of the resource should try their best to communicate their interpretation within their finite means. The reason URIs identify resources, and not individual Web representations, is that Web representations are too ephemeral to

¹⁰Of course, one cannot control the interpretations of yet unknown agents, so all sorts of absurdities are possible in theory. As the interpretation of the same encoding can differ among agents, there is a possibility that the owner of the URI <http://www.example.org/Moon> really thinks that for French speakers a picture of blue cheese has the same sense as a picture of the Moon for English speakers, even if users of the resource disagree. However, it should be remembered that the Web is a space of communication, and that for communication to be successful over the Web using URIs, it is in the interest of the owner of the resource to deploy Web representations that they believe the users will share their interpretation of. So content negotiation between a picture of blue cheese and a picture of the moon for a resource that depicts the Moon is, under normal circumstances, the Web equivalent of insanity at worst, or bad manners at best.

want to identify in and of themselves, being by definition the response of a server to a *particular* response and request for information. While one could imagine wanting to access a particular Web representation, in reality what is usually wanted by the user-agent is the content of the resource, which may be present in a wide variety of languages. What is important is that the sense gets transferred and interpreted by the user agent, not the individual bytes of a particular encoding in a particular language at a particular time.

2.2.5 *Digitality*

The Web is composed of not just representations, but digital representations. One of the defining characteristics of information on the Web is that this information is digital, bits and bytes being shipped around by various protocols. Yet there is no clear notion of what ‘being’ digital consists of, and a working notion of digitality is necessary to understand what can and can not be shipped around as bytes on the Web. Much like the Web itself, we can know something digital when we spot it, and we can build digital devices, but developing an encompassing notion of digitality is a difficult task, one that we only characterize briefly here.

Goodman defined marks as “*finitely differentiable*” when it is possible to determine for any given mark whether it is identical to another mark or marks (Goodman 1968). This can be considered equivalent to how in categorical perception, despite variation in handwriting, a person perceives hand-written letters as being from a finite alphabet. So, *equivalence classes of marks can be thought of as an application of the philosophical notion of types*. This seems close to ‘digital,’ so that given a number of types of content in a language, a system is digital if any mark of the encoding can be interpreted to one and only one type of content. Therefore, in between any two types of content or encoding there cannot be an infinite number of other types. Digital systems are the opposite of Bateson’s famous definition of information: Being digital is simply having a difference that does not make difference (Bateson 2001). This is not to say there are characteristics of a mark which do not reflect its assignment in a type, and these are precisely the characteristics which are lost in digital systems. So in an analogue system, every difference in some mark makes a difference, since between any two types there is another type that subsumes a unique characteristic of the token. In this manner, the prototypical digital system is the discrete distribution of integers, while the continuous numbers are the analogue system par excellence, since between any real number there is another real number.

Lewis took aim at Goodman’s interpretation of digitality in terms of determinism by arguing that digitality was actually a way to represent possibly continuous systems using the combinatorics of discrete digital states (1971). To take a less literal example, discrete mathematics can represent continuous subject matters. This insight caused Haugeland to point out that digital systems are always abstractions built on top of analog systems (1981). The reason we build these abstractions is

because digital systems allow perfect reliability, so that once a system is in a digital type (also called a ‘digital state’), it does not change unless it is explicitly made to change, allowing both flawless copying and perfect reliability. Haugeland reveals the purpose of digitality to be “a mundane engineering notion, root and branch. It only makes sense as a practical means to cope with the vagarities and vicissitudes, the noise and drift, of earthy existence” (Haugeland 1981). Yet Haugeland does not tell us what digitality actually is, although he tells us what it does, and so it is unclear why certain systems like computers have been wildly successful due to their digitality (as the success of analogue computers was not so widespread), while others like ‘integer personality ratings’ have not been as successful. Without a coherent definition of digitality, it is impossible to even in principle answer questions like whether or not digitality is *purely* subjective (Mueller 2008). Any information is **digital** when *the boundaries in a particular encoding can converge with a regularity in a physical realization*. This would include sentences in a language that can be realized by sound-waves or the text in an e-mail message that can be re-encoded as bits, and then this encoding realized by a series of voltages. Since the encoding of the information can be captured perfectly by a digital system, it can be copied safely and effectively, just as an e-mail message can be sent many times or a digital image reproduced countlessly.

To implement a digital system, there must be a small chance that the information realization can be considered to be in a state that is not part of the discrete types given by the encoding. The regularities that compose the physical boundary allows within a margin of error a discrete boundary decision to be made in the interpretation of the encoding. So, anything is capable of upholding digitality if that buffer created by the margin of error has an infinitesimal chance at any given time of being in a state that is not part of the encoding’s discrete state. For example, the hands on a clock can be on the precise boundary between the markings on the clock, just not for very long. In a digital system, on a given level of abstraction, the margin of error does not propagate upwards to other levels of abstraction that rest on the earlier level of abstractions. Since we can create physical systems through engineering, we can create physical substrata that have low probabilities of being in states that do not map to digital at a given level of abstraction. As put by Turing, “The digital computers. . . may be classified amongst the ‘discrete state machines,’ these are the machines which move by sudden jumps or clicks from one quite definite state to another. These states are sufficiently different for the possibility of confusion between them to be ignored. Strictly speaking there are no such machines. Everything really moves continuously” (Turing 1950). **Analogue** is the rather large and heterogeneous set of *everything that is not digital*. This would include people, such as Tim Berners-Lee himself, who can be represented but not realized as a message, as well as places, like Mount Everest, whose precise boundaries are rather indeterminate. While, according to Hayles, “the world as we sense it on the human scale is basically analogue,” and the Web is yet another development in a long-line of biological modifications and technological prostheses to impose digitalization on an analogue world (2005). The vast proliferation of digital technologies is possible because there are physical substrata, some more so than others, which support

the realization of digital information and give us the advantages that Haugeland rightfully points out is the purpose of the digital: flawless copying and perfect reliability in a flawed and imperfect world (1981).

2.2.6 Representations

A web-page about the Eiffel Tower seems to be an obvious representation. One can sit at home on one's computer far away from Paris and access a web-page that features a clear picture of – a representation! – of the Eiffel Tower. Furthermore, others from Japan to Egypt should be able to access the exact same representation by accessing the same URI. By claiming to be a “universal space of information,” the Web is asserting to be a space where any encoding can be transferred about any content (Berners-Lee et al. 1992). However, there are some distinct differences between kinds of content, for some content can be distal and other content can be local. *Things that are separated by time and space* are **distal** while *those things that are not separated by time and space* are **proximal**. As synonyms for distal and proximal, we will use **non-local** and **local**, or just **disconnected** and **connected**. Although this may seem to be an excess of adjectives to describe a simple distinction, this aforementioned distinction will underpin our notions of representation. In a message between two computers, if the content is a set of commands to ‘display these bytes on the screen’ then the client can translate these bytes to the screen directly without any worry about what those bytes represent to a human user. However, the content of the message may involve some distal components, such as the string “The Eiffel Tower is in Paris,” which refers to many things outside of the computer. Differences between receivers allow the self-same content of a message to be both distal and local, depending on the interpreting agent. The message to ‘display these bytes on the screen’ could cause a rendering of a depiction of the Eiffel Tower to be displayed on the screen, so the self-same message causes not only a computer to display some bytes but also causes a human agent to receive information about what the Eiffel Tower in Paris looks like.

Any *encoding of information that has distal content* is called a **representation**, regardless of the particular encoding of the information. Representations are then a subset of information, and inherit the characteristics outlined of all information, such as having one or more possible encodings and often a purpose and the ability to evoke normative behaviour from agents. To have some relationship to a thing that one is disconnected from is to be *about* something else. Generally, *the relationship of a thing to another thing to which one is immediately causally disconnected* is a relationship of **reference** to a **referent** or **referents**, *the distal thing or things referred to by a representation*. The thing which refers to the referent(s) we call the ‘representation,’ and take this to be equivalent to being a *symbol*. *Linguistic expressions of a natural or formal language* are called **descriptions** while *the expressions of an iconic language* are called **depictions**. To refer to something is to *denote* something, so the content of a representation is its *denotation*. In the tradition

of Bretano, the reference relation is considered *intentional* due to its apparent physical spookiness. After all, it appears there is some great looming contradiction: if the content is whatever is held in common between the source and the receiver as a result of the conveyance of a particular message, then how can the source and receiver share some information they are disconnected from?

On the surface this aspect of ‘representation’ seems to be what Brian Cantwell Smith calls “physically spooky,” since a representation can refer to something with which it is not in physical contact (Smith 1996). This spookiness is a consequence of a violation of *common-sense* physics, since representations allow us to have some sort of what appears to be a non-physical relationship with things that are far away in time and space. This relationship of ‘aboutness’ or *intentionality* is often called ‘reference.’ While it would be premature to define ‘reference,’ a few examples will illustrate its usage: someone can think about the Eiffel Tower in Paris without being in Paris, or even having ever set foot in France; a human can imagine what the Eiffel Tower would look like if it were painted blue, and one can even think of a situation where the Eiffel Tower wasn’t called the Eiffel Tower. Furthermore, a human can dream about the Eiffel Tower, make a plan to visit it, all while being distant from the Eiffel Tower. Reference also works temporally as well as distally, for one can talk about someone who is no longer living such as Gustave Eiffel. Despite appearances, reference is not epiphenomenal, for reference has real effects on the behaviour of agents. Specifically, one can remember what one had for dinner yesterday, and this may impact on what one wants for dinner today, and one can book a plane ticket to visit the Eiffel Tower after making a plan to visit it.

We will have to make a somewhat convoluted trek to resolve this paradox. The very idea of representation is usually left under-defined as a “standing-in” intuition, that a representation is a representation by virtue of “standing-in” for its referent (Haugeland 1991). The classic definition of a symbol from the Physical Symbol Systems Hypothesis is the genesis of this intuition regarding representations (Newell 1980): “An entity *X* designates an entity *Y* relative to a process *P*, if, when *P* takes *X* as input, its behaviour depends on *Y*.” There are two subtleties to Newell’s definition. Firstly, the notion of a representation is grounded in the behaviour of an agent. So, what precisely counts as a representation is never context-free, but dependent upon the agent completing some purpose with the representation. Secondly, the representation *simulates* its referent, and so the representation must be local to an agent while the referent may be non-local: “This is the symbolic aspect, that having *X* (the symbol) is tantamount to having *Y* (the thing designated) for the purposes of process *P*” (Newell 1980). We will call *X* a representation, *Y* the *referent* of the representation, a process *P* the representation-using *agent*. This definition does not seem to help us in our goal of avoiding physical spookiness, since it pre-supposes a strangely Cartesian dichotomy between the referent and its representation. To the extent that this distinction is held a priori, then it is physically spooky, as it seems to require the referent and representation to somehow magically line up in order for the representation to serve as a substitute for its missing referent.

The only way to escape this trap is to give a non-spooky theory of how representations arise from referents. Brian Cantwell Smith tackles this challenge by developing a theory of representations that explains how they arise temporally (1996). Imagine Ralph, the owner of a URI at which he wants to host a picture of the Eiffel Tower, finally gets to Paris and is trying to get to the Eiffel Tower in order to take a digital photo. In the distance, Ralph sees the Eiffel Tower. At that very moment, Ralph and the Eiffel Tower are both physically connected via light-rays. At the moment of tracking, connected as they are by light, Ralph, its light cone, and the Eiffel Tower are a system, not distinct individuals. An alien visitor might even think they were a single individual, a ‘Ralph-Eiffel Tower’ system. While walking towards the Eiffel Tower, when the Eiffel Tower disappears from view (such as from being too close to it and having the view blocked by other buildings), Ralph keeps staring into the horizon, focused not on the point the Eiffel Tower was at before it went out of view, but the point where he thinks the Eiffel Tower would be, given his own walking towards it. Only when parts of the physical world, Ralph and the Eiffel Tower, are now physically separated can the agent then use a representation, such as the case of Ralph using an internal “mental image” of the Eiffel Tower or the external digital photo to direct his walking towards it, even though he cannot see it. The agent is distinguished from the referent of its representation by virtue of not only disconnection but by the agent’s attempt to track the referent, “a long-distance coupling against all the laws of physics” (Smith 1996). The local physical processes used to track the object by the subject are the representation, be they ‘inside’ a human in terms of a memory or ‘outside’ the agent like a photo in a digital camera.

This notion of representation is independent of the representation being either internal or external to the particular agent, regardless of how one defines these boundaries.¹¹ Imagine that Ralph had been to the Eiffel Tower once before. He could have marked its location on a piece of paper by scribbling a small map. Then, the marking on the map could help guide him back as the Eiffel Tower disappears behind other buildings in the distance. This characteristic of the definition of representation being capable of including ‘external’ representations is especially important for any definition of a representation to be suitable for the Web, since the Web is composed of information that is considered to be external to its human users.

However fuzzy the details of Smith’s story about representations may be, what is clear is that instead of positing a connection between a referent and a representation a priori, they are introduced as products of a temporal process. This process is at least theoretically non-spooky since the entire process is capable of being grounded out in physics without any spooky action at a distance. To be grounded out in physics, all changes must be given in terms of connection in space and time, or in other words, via effective reach. Representations are “a way of exploiting local freedom or slop in order to establish coordination with what is beyond effective reach” (Smith 1996). In order to clarify Smith’s story and improve the definition of

¹¹The defining of “external” and “internal” boundaries is actually non-trivial, as shown in Halpin (2008a).

the Physical Symbol Systems Hypothesis, we consider Smith's theory of the "origin of objects" to be a *referential chain* with distinct stages (Halpin 2006):

- **Presentation:** Process S is connected with process O .
- **Input:** The process S is connected with R . Some local connection of S puts R in some causal relationship with process O via an encoding. This is entirely non-spooky since S and O are both connected with R . R eventually becomes the representation.
- **Separation:** Processes O and S change in such a way that the processes are disconnected.
- **Output:** Due to some local change in process S , S uses its connection with R to initiate local meaningful behaviour that is in part caused by R .¹²

In the 'input' stage, the *referent* is the cause of some characteristic(s) of the information. The relationship of *reference* is the relationship between the encoding of the information (the representation) and the referent. The relationship of interpretation becomes one of reference when the distal aspects of the content are crucial for the meaningful behaviour of the agent, as given by the 'output' stage. So we have constructed an ability to talk about representations and reference while not presupposing that behaviour depends on internal representations or that representations exist a priori at all. Representations are only needed when the relevant intelligent behaviour requires some sort of distal co-ordination with a disconnected thing.

So the interpretation of a representation – a particular kind of encoding of content – results in behavior by the user-agent that is dependent on a distal referent via the referential chain (Fig. 2.7). In this manner, the act of reference can then be defined as the interpretation of a representation. This would make our notion of representation susceptible to being labelled a *correspondence theory of truth* (Smith 1986), where a representation refers by some sort of structural correspondence to some referent. However, our notion of representation is much weaker, requiring only a causation between the referent and the representation – and not just any causal relationship, but one that is meaningful for the interpreting agent – as opposed to some tighter notion of correspondence such as some structural 'isomorphism' between a representation and its "target," the term used by Cummins to describe what we have called the "referent" of a representation (1996). So an interpretation or an act of reference should therefore not be viewed as a mapping to referents, but as a mapping to some content – where that content leads to meaningful behaviour precisely because of some referential chain. This leads to the notion of a Fregean 'objective' sense, which we turn to later.

Up until now, it has been implicitly assumed that the referent is some physical entity that is non-local to the representation, but the physical entity is still existent, such as the Eiffel Tower. However, remember that the definition of non-local includes *anything* the representation is disconnected from, and so includes physical

¹²In terms of Newell's earlier definition, O is X while S is P and R is Y .

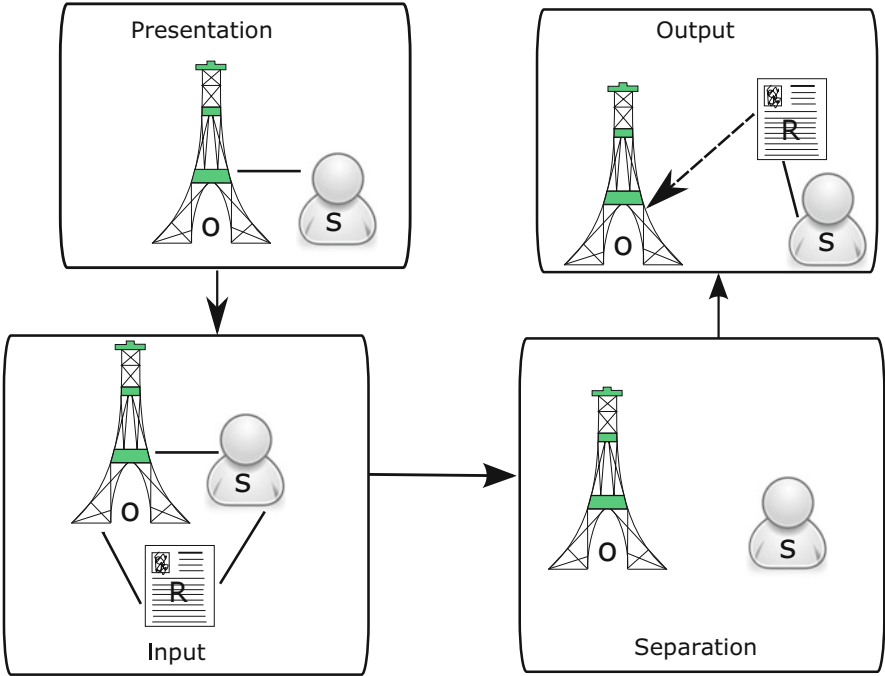


Fig. 2.7 The referential chain

entities that may exist in the past or the future. The existence of a representation does not imply the existence of the referent or the direct acquaintance of the referent by the agent using a representation – a representation only implies that some aspect of the content is non-local. However, this seems to contradict our ‘input’ stage in the representational cycle, which implies that part of our definition of representation is historical: for every *re*-presentation there must be a presentation, an encounter with the thing presented. By these conditions, the famous story of Putnam’s example of an ant tracing a picture of Winston Churchill by sheer accident in the sand would not count as a representation (1975). If a tourist didn’t know where the Eiffel Tower was, but navigated the streets of Paris and found the Eiffel Tower by reference to a tracing of a Kandinsky painting in his notebook, then the tourist would not then be engaged in any representation-dependent meaningful behaviour, since the Kandinsky painting lacks the initial presentation with the Eiffel Tower. The presentation does not have to be done by the subject that encountered the thing directly. However, the definition of a representation does not mean that the *same* agent using the representation had to be the agent with the original presentation. A representation that is created by one agent in the presence of a referent can be used by another agent as a ‘stand-in’ for that referent if the second agent shares the same interpretation from encoding to distal content. So, instead of relying on his own vision, a tourist buys a map and so relies on the ‘second-order’ representation of the

map-maker, who has some historical connection to someone who actually travelled the streets of Paris and figured out where the Eiffel Tower was. In this regard, our definition of representation is very much historical, and the original presentation of the referent can be far back in time, even evolutionary time, as given by accounts like those of Millikan (1984). One can obviously refer to Gustave Eiffel even though he is long dead and buried, and so no longer exists.

Also, the referent of a representation may be what we think of as real-world patches of space and time like people and places, abstractions like the concept of a horse, to unicorns and other imaginary things, future states such as ‘see you next year,’ and descriptive phrases whose supposed *exact* referent is unknown, such as ‘the longest hair on your head on your next birthday.’ While all these types of concepts are quite diverse, they are united by the fact that they cannot be completely realized by local information, as they depend on partial aspects of an agent’s local information, the future, or things that do not exist. Concepts that are constructed by definition, including imaginary referents, also have a type of ‘presence,’ it is just that the ‘presentation’ of the referent is created via the initial description of the referent. Just because a referent is a concept – as opposed to a physical entity – does not mean the content of the representation cannot have an meaningful effect on the interpreter. For example, exchanging representations of ‘ghosts’ – even if they do not quite identify a coherent class of referents – can govern the behavior of ghost-hunters. Indeed, it is the power and flexibility of representations of these sorts that provide humans the capability to escape the causal prison of their local environment, to plan and imagine the future.

2.3 The Principles of Web Architecture

It is now possible to show how the various Web terms are related to each other in a more systematic way. These relationships are phrased as five finite principles that serve as the normative Principles of Web architecture: The Principles of Universality, Linking, Self-Description, the Open World, and Least Power. In practice many applications violate these principles, and by virtue of their use of URIs and the HTTP protocol, many of these applications would be in some sense ‘on the Web.’ However, these principles are normative insofar as they define what could be considered as compliance with Web architecture, and so an application that embodies them is compliant with Web architecture.

2.3.1 Principle of Universality

The *Principle of Universality* can be defined as *any resource that can be identified by a URI*. The notion of both a resource and a URI was from their onset universal in its ambition, as Berners-Lee said, “a common feature of almost all the data

models of past and proposed systems is something which can be mapped onto a concept of ‘object’ and some kind of name, address, or identifier for that object. One can therefore define a set of name spaces in which these objects can be said to exist. In order to abstract the idea of a generic object, the web needs the concepts of the universal set of objects, and of the universal set of names or addresses of objects” (1994a). The more informal notes of Berners-Lee are even more startling in their claims for universality, stating that the first ‘axiom’ of Web architecture is “universality” where “by ‘universal’ I mean that the Web is declared to be able to contain in principle every bit of information accessible by networks” (1996b). Although it appears he may be constraining himself to only talk about digital ‘objects’ that are accessible over the Internet in this early IETF RFCs, in later IETF RFCs the principle quickly ran amok, as users of the Web wanted to use URIs to refer to “human beings, corporations, and bound books in a library” (Berners-Lee et al. 1998).

There seems to be a certain way that web-pages are ‘on the Web’ in a way that human beings, corporations, unicorns, and the Eiffel Tower are not. Accessing a web-page in a browser means to receive some bits, while one cannot easily imagine what accessing the Eiffel Tower itself or the concept of a unicorn in a browser even means. This property of being ‘on the Web’ is a common-sense distinction that separates things like a web-page about the Eiffel Tower from things like the Eiffel Tower itself. This distinction is that between the use of URIs to *access* and *reference*, between the local and the distal. The early notes of Berners-Lee that pre-date the notion of URIs itself address this distinction between access and reference, phrasing it as a distinction between locations and names. As Berners-Lee states, “conventionally, a ‘name’ has tended to mean a logical way of referring to an object in some abstract name space, while the term ‘address’ has been used for something which specifies the physical location” (1991). So, a **location** is a term that can be used to access the thing, while a **name** is a term that can be used to refer to a thing. Unlike access, reference is the use of an identifier for a thing to which one is immediately causally disconnected. **Access** is the use of an identifier to create immediately a causal connection to the thing identified (Hayes and Halpin 2008). The difference between the use of a URI to access a hypertext web-page or other sort of information-based resource and the use of a URI to refer to some non-Web accessible entity or concept ends up being quite important, as this ability to representationally use URIs as ‘stands-in’ for referents forms the basis of the distinction between the hypertext Web and the Semantic Web.

Names can serve as identifiers and even representations for distal things. However, Berners-Lee immediately puts forward the hypothesis that “with wide-area distributed systems, this distinction blurs” so that “things which at first look like physical addresses... cease to give the actual location of the object. At the same time, a logical name... must contain some information which allows the name server to know where to start looking” (1991). He posits a third neutral term, “identifier” that was “generally referred to a name which was guaranteed to be unique but had little significance as regards the logical name or physical address” (Berners-Lee 1991). In other words, an **identifier** is a term that can be used to either access or

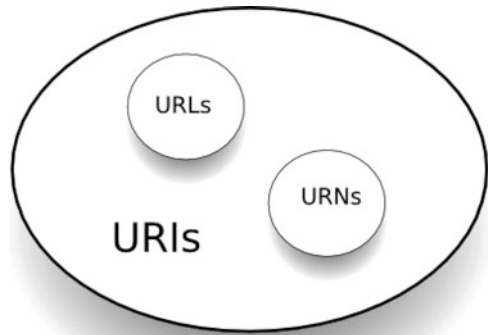
refer, or both access and refer to, a thing. The problem at hand for Berners-Lee was how to provide a name for his distributed hypertext system that could get “over the problem of documents being physically moved” (1991). Using simple IP addresses or any scheme that was tied to a single server would be a mistake, as the thing that was identified on the Web should be able to move from server to server without having to change identifier.

For at least the first generation of the Web, the way to overcome this problem was to provide a translation mechanism for the Web that could provide a methodology for transforming “unique identifiers into addresses” (Berners-Lee 1991). Mechanisms for translating unique identifiers into addresses already existed in the form of the domain name system that was instituted by the IETF in the early days of the expansion of ARPANet (Mockapetris Novemeber 1983). Before the advent of the domain name system, the ARPANet contained one large mapping of identifiers to IP addresses that was accessed through the Network Information Centre, created and maintained by Engelbart (Hafner and Lyons 1996). However, this centralized table of identifier-to-address mappings became too unwieldy for a single machine as ARPANet grew, so a decentralized version was conceived based on *domain names*, where each domain name is *a specification for a tree structured name space, where each component of the domain name (part of the name separated by a period) could direct the user-agent to a more specific “domain name server” until the translation from an identifier to an IP address was complete.*

Many participants in the IETF felt like the blurring of this distinction that Berners-Lee made was incorrect, so URIs were bifurcated into two distinct specifications. *A scheme for locations that allowed user-agents via an Internet protocol to access information* was called **Uniform Resource Locations** (URLs) (Berners-Lee et al. 1994) while *a scheme whose names could refer to things outside of the causal reach of the Internet* was called **Uniform Resource Names** (URNs) (Sollins and Masinter 1994). Analogue things like concepts and entities naturally had to be given URNs, and digital information that can be transmitted over the Internet, like web-pages, were given URLs. Interestingly enough, URNs count *only* as a naming scheme, as opposed to a protocol like HTTP, because they cannot access any information. While one could imagine a particular Web-accessible realization, like a web-page, disappearing from the Web, it was felt that identifiers for things that were not accessible over the Web should “be globally unique forever, and may well be used as a reference to a resource well beyond the lifetime of the resource it identifies or of any naming authority involved in the assignment of its name” (Mealling and Daniel 1999).

Precisely because of their lack of ability to access information, URNs never gained much traction, while URLs to access web-pages became the norm. Building on this observation about the “blurring of identifiers,” the notion of URIs implodes the distinction between identifiers used only for access (URLs) and the identifiers used for reference (URNs). A **Uniform Resource Identifier** is *a unique identifier whose syntax is given by its latest IETF RFC that may be used to either or both refer to or access a resource* (Berners-Lee et al. 2005). URIs subsume both URLs and URNs, as shown in Fig. 2.8. Berners-Lee and others were only able to push

Fig. 2.8 A Venn diagram describing the relationships between URIs, URNs, and URLs



this standard through the IETF process years after the take-off of the Web. Indeed, early proposals for universal names, ranging from Raymond Lull to Engelbart's 'Every Object Addressable' principle (1990), all missed the crucial advantage of the Web; while classically names in natural language are used for reference, on the Web names can be used to access information. In a decentralized environment this is crucial for discovering the sense of a URI, as illustrated by the notions of 'linking' and 'self-description' detailed next in Sects. 2.3.2 and 2.3.3.

2.3.2 Principle of Linking

The **Principle of Linking** states that *any resource can be linked to another resource identified by a URI*. No resource is an island, and the relationships between resources are captured by linking, transforming lone resources into a Web. A **link** is a *connection between resources*. The *resource that the link is directed from* is called its **starting resource** while the *resource a link is directed to* is the **ending resource** (DeRose et al. 2001).

What are links for? Just as URIs, links may be used for either access or reference, or even both. In particular, in HTML the purpose of links is for access to additional hypertext documents, and so they are sometimes called hyperlinks. This access is often called *following* the link, a transversal from one Web representation to another, that results in access to Web representations of the ending resource. A unidirectional link that allows access of one resource from another is the predominant kind of link in hypertext. Furthermore, access by linking is transitive, for if a user-agent can access a Web representation of the ending resource from the starting resource, then it can access any links present in the Web representation, and thereby access a Web representation of an ending resource. It is precisely this ability to transitively access documents by following links that led the original Web to be a seamless Web of hypertext. While links can start in Web representations, the

main motivation for using URIs as the ending resource of a link as opposed to a specific Web representation is to prevent *broken links*, where a user-agent follows a link to a resource that is no longer there, due to the Web representation itself changing. As put by the TAG, “Resource state may evolve over time. Requiring a URI owner to publish a new URI for each change in resource state would lead to a significant number of broken references. For robustness, Web architecture promotes independence between an identifier and the state of the identified resource” (Jacobs and Walsh 2004).

However, one of the distinguishing features of the Web is that links may be broken by having any access to a Web representation disappear, due to simply the lack of hosting a Web representation, loss of ownership of the domain name, or some other reason. These reasons are given in HTTP status codes, such as the infamous 404 *Not Found* that signals that while there is communication with a server, the server does not host the resource. Further kinds of broken links are possible, such as 301 *Moved Permanently* or a 5xx server error, or an inability to even connect with the server leading to a time-out error. This ability of links to be ‘broken’ contrasts to previous hypertext systems. Links were not invented by the Web, but by the hypertext research community. Constructs similar to links were enshrined in the earliest of pre-Web systems, such as Engelbart’s *oNLine System* (NLS) (1962), and were given as part of the early hypertext work by Theodor Nelson (1965). The plethora of pre-Web hypertext systems were systematized into the Dexter Reference Model (Halasz and Schwartz 1994). According to the Dexter Reference Model, the Web would not even qualify as hypertext, but as “proto-hypertext,” since the Web did not fulfill the criteria of “consistency,” which requires that “in creating a link, we must ensure that all of its component specifiers resolve to existing components” (Halasz and Schwartz 1994). To ensure a link must resolve and therefore not be broken, this mechanism requires a centralized link index that could maintain the state of each resource and not allow links to be created to non-existent or non-accessible resources. Many early competitors to the Web, like HyperG, had a centralized link index (Andrews et al. 1995). As an interesting historical aside, it appears that the violation of this principle of maintaining a centralized link index was the main reason why the World Wide Web was rejected from its first academic conference, ACM Hypertext 1991, although Engelbart did encourage Berners-Lee and Connolly to pursue the Web further.¹³ While a centralized link index would have the benefit of not allowing a link to be broken, the lack of a centralized link index removes a bottleneck to growth by allowing the owners of resources to link to other resources without updating any index besides their own Web representations. This was doubtless important in enabling the explosive growth of linking. The lack of any centralized link index, and index of Web representations, is also precisely what search engines like Google create post-hoc through spidering, in order to have an index of links and web-pages that enable their keyword search and page ranking algorithms. As put by Dan Connolly in response to Engelbart, “the design of the

¹³Personal communication with Tim Berners-Lee.

Web trades link consistency guarantees for global scalability” (2002). So, broken links and 404 Not Found status codes are purposeful *features*, not defects, of the Web.

2.3.3 Principle of Self-description

One of the goals of the Web is for resources to be ‘self-describing,’ currently defined as “individual documents become self-describing, in the sense that only widely available information is necessary for understanding them” (Mendelsohn 2006). While it is unclear what “widely-available” means, one way for information to be widely-available is for it to be linked to from the Web representation itself. The **Principle of Self Description** states that *the information an agent needs in order to have an interpretation of a Web Representation (resource) should be accessible from the Web representation itself (URI)*.

How many and what sort of links are necessary to adequately describe a resource? A resource is successfully described if an interpretation of a sense is possible. Any representation can have links to other resources which in turn can determine valid interpretations for the original resource. This process of following whatever data is linked in order to determine the interpretation of a URI is informally called ‘following your nose’ in Web architecture.

The **Follow-Your-Nose algorithm** states that if a user-agent encounters a representation in a language that the user-agent cannot interpret, the user-agent should, in order:

1. **Dispose of Fragment Identifiers:** As mandated (Berners-Lee et al. 2005), user-agents can dispose of the fragment identifier in order to retrieve whatever Web representations are available from the *racine* (the URI without fragment identifier). For example, in HTML the fragment identifier of the URI is stripped off when retrieving the webpage, and then when the browser retrieves a Web representation, the fragment identifier can be used to locate a particular place within the Web representation.
2. **Inspect the Media Type:** The media type of a Web representation provides a normative declaration of how to interpret a Web representation. Since the number of IETF media-types is finite and controlled by the IETF, a user-agent should be able to interpret these media types.¹⁴
3. **Follow any Namespace Declarations:** Many Web representations use a generic format like XML to in turn specify a customized dialect. In this case, a language or dialect is itself given a URI, called a **namespace URI**, a URI that identifies that particular dialect. A namespace URI then in turn allows access to a **namespace**

¹⁴The finite list is available at <http://www.iana.org/assignments/media-types/>, and a mapping from media types to URIs has been proposed at <http://www.w3.org/2001/tag/2002/01-uriMediaType-9>.

document, a Web representation that provides more information about the dialect. In a Web representation using this dialect, a *namespace declaration* then specifies the namespace URI. In this case, the user-agent may follow these namespace declarations in order to get the extra information needed to interpret the Web representation. As a single Web representation may be encoded in multiple languages, it may have multiple namespace URIs to follow.

4. **Follow any links:** The user-agent can follow any links. There are some links in particular languages that may be preferred, such as the ending resource of a `link` header in HTML or RDF Schema links such as *rdfs:isDefinedBy* links, or links like OWL by the *owl:imports*. If links are typed in some fashion, each language may define or recommend links that have the normative status, and normative links should be preferred. However, for many kinds of links, their normative status is unclear, so the user-agent may have to follow any sort of link as a last resort.

Using this algorithm, the user-agent can begin searching for some information that allows it to interpret the Web representation. It can follow the first three guidelines and then follow the fourth, applying the above guidelines recursively. Eventually, this recursive search should bottom out either in a program that allows an interpretation of the Web representation (such as a rendering of a web-page or inferences given by a Semantic Web language) or specifications given by the IETF in plain, human-readable text, the natural bottoming point of self-description. This final fact brings up the point that the information that gets one an interpretation is not necessarily a program, but could be a human-readable specification that requires a human to make the mapping from the names to the intended sense.

2.3.4 The Open World Principle

The *Open World Principle* states that *the number of resources on the Web can always increase*. There can always be new acts of identification, carving out a new resource from the world and identifying it with a URI. At any given moment, a new web-page may appear on the Web, and it may or may not be linked to. This is a consequence of the relatively decentralized creation of URIs for resources given by the Principle of Universality and the decentralized creation of links by the Principle of Linking. Without any centralized link index, there is no central repository of the state of the *entire* Web. While approximations of the state of the entire Web are created by indexing and caching web-pages by search engines like Google, due to the Open World Principle, none of these alternatives will necessarily ever be guaranteed to be complete. Imagine a web-spider updating a search engine index. At any given moment, a new resource could be added to the Web that the web-spider may not have crawled. So to assume that any collection of resources of the Web can be a complete picture of the whole Web is at best impudent.

The ramifications of the Open World Principle are surprising, and most clear in terms of judging whether a statement is true or false. These repercussions transform the Open World Principle into its logical counterpart, the ***Open World Assumption***, which logically states that *statements that cannot be proven to be true cannot be assumed to be false*. Intuitively, this means that the world cannot be bound. On the Web, the Open World Principle holds that since the Web can always be made larger, with any given set of statements that allow an inference, a new statement relevant to that inference may be found. So any agent’s knowledge of the Web is always partial and incomplete, and thus the Open World Assumption is a safe bet for agents on the Web. The Open World Principle is one of the most influential yet challenging principles of the Web, the one that arguably separates the Web from traditional research in artificial intelligence and databases in practice. In these fields, systems tend to make the opposite of the Open World Assumption, the Closed World Assumption. The ***Closed World Assumption*** states that logically *statements that cannot be proven to be true can be assumed to be false*. Intuitively, this means that somehow the world can be bounded. The Closed World Assumption has been formalized on a number of different occasions, with the first formalization being due to Reiter (1978). This assumption has often been phrased as an appeal to the Law of the Excluded Middle ($\forall p.p \vee \neg p$) in classical logic (Detlefsen 1990). *Negation as failure* is an implementation of the Closed World assumption in both logic programming and databases, where failure for the program to prove a statement is true implies the statement is false (Clark 1978).

2.3.5 Principle of Least Power

The Principle of Least Power states that a *Web representation given by a resource should be described in the least powerful but adequate language*. This principle is also normative, for if there are multiple possible Web representations for a resource, the owner should choose the Web representation that is given in the ‘least powerful’ language. The Principle of Least Power seems odd, but it is motivated by Berners-Lee’s observation that “we have to appreciate the reasons for picking not the most powerful solution but the least powerful language” (1996b). The reasons for this principle are rather subtle. The receiver of the information accessible from a URI has to be able to decode the language that the information is encoded in so the receiver can determine the sense of the encoding. Furthermore, an agent may be able to decode multiple languages, but the owner of the URI does not know what languages an agent wanting to access their URI may possess. Also, the same agent may be able to interpret multiple languages that can express the same sense. So, the question always facing any agent trying to communicate is what language to use? In closed and centralized systems, this is ordinarily not a problem, since each agent can be guaranteed to use the same language. In an open system like the Web, where one may wish to communicate a resource to an unknown number of agents, each of which may have different language capabilities, the question of which language to

deploy becomes nearly insurmountable. Obviously, if an agent is trying to convey some sense, then it should minimally choose a language to encode that sense which is capable of conveying that sense. Yet as the same sense can be conveyed by different languages, what language to choose?

The Principle of Least-Power is a common-sense engineering solution to this problem of language choice. The solution is simply to build first a common core language that fulfills the minimal requirements to communicate whatever sense one wishes to communicate, and then extend this core language. Using HTML as an example, one builds first a common core of useful features such as the ability to have text be bold and have images inserted in general areas of the text, and then as the technology matures, to slowly add features such as the precise positioning of images and the ability to specify font size. The Principle of Least Power allows a straightforward story about compatibility to be built to honor the “be strict when sending and tolerant when receiving” maxim of the Internet, since it makes the design of a new version an exercise in strictly extending the previous version of the language (Carpenter 1996). A gaping hole in the middle of the Principle of Least Power is that there is no consistent definition of the concept of ‘power,’ and the W3C TAG seems to conflate power with the Chomsky Hierarchy. However, the problem of defining ‘power’ formally must be left as an open research question.

2.4 Conclusions

The Web, while to a large extent being an undisciplined and poorly-defined space, does contain a set of defining terms and principles. While previously these terms and principles have been scattered throughout various informal notes, IETF RFCs, and W3C Recommendations, in this chapter we have systematized both the terminology and the principles in a way that reveals how they internally build off each other. In general, when we are referring to the *hypertext Web*, we are referring to the use of URIs and links to access hypertext web-pages using HTTP. Yet there is more to the Web than hypertext. The next question is how can these principles be applied to domains outside the hypertext Web, and this will be the topic of Chap. 3 as we apply these principles to the Semantic Web, a knowledge representation language for the Web.



<http://www.springer.com/978-1-4614-1884-9>

Social Semantics

The Search for Meaning on the Web

Halpin, H.

2013, XVI, 220 p., Hardcover

ISBN: 978-1-4614-1884-9