Chapter 6
In Defense of Ambiguity Redux

Patrick J. Hayes
Institute for Human and Machine Cognition, USA

Harry Halpin
School of Informatics, University of Edinburgh, UK

ABSTRACT

URIs, a universal identification scheme, are different from human names insofar as they can provide the ability to reliably access the thing identified. URIs also can function to reference a non-accessible thing in a similar manner to how names function in natural language. There are two distinctly different relationships between names and things: access and reference. To confuse the two relations leads to underlying problems with Web architecture. Reference is by nature ambiguous in any language. So any attempts by Web architecture to make reference completely unambiguous will fail on the Web. Despite popular belief otherwise, making further ontological distinctions often leads to more ambiguity, not less. Contrary to appeals to Kripke for some sort of eternal and unique identification, reference on the Web uses descriptions and therefore there is no unambiguous resolution of reference. On the Web, what is needed is not just a simple redirection, but a uniform and logically consistent manner of associating descriptions with URIs that can be done in a number of practical ways that should be made consistent.

DOI: 10.4018/978-1-60566-992-2.ch006
fier 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 concept of the universal set of objects, and of the universal set of names or addresses of objects” (Berners-Lee, 1994). The claim is that a URI can identify anything, not just a Web page. If the Web is to fulfill its ambition of being a universal information space, then the Web naturally has to stretch across to other information systems to objects one might want to access, like files through File Transfer Protocol or e-mail addresses through a mailto: URI scheme. These pose no philosophical problems as they are all streams of bits that happen to conform to different protocols.

Later, Berners-Lee and others attempted to define a resource as “anything that has identity” in RFC 2396 (1998). Even in this definition, a resource was defined as broader than just things that are accessible via the Web, since resources may 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). In the latest version, RFC 3986, the definition of resource is updated to state that “abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., “parent” or “employee”), or numeric values (e.g., zero, one, and infinity)” (Berners-Lee et al., 2005). So a resource is anything someone might want to identify with a URI, including things far outside the reach of the Web.

Most previous hypertext systems assumed that “resource” meant an accessible thing. This corresponds fairly accurately to the English meaning of resource, which is just a thing that can be used for support or help. Although Berners-Lee has always supported a wider role for URIs as “Universal Resource Identifiers,” at first he could only get through IETF standards track a specification for “Uniform Resource Locators” that was for a “resource available via the Internet” (Berners-Lee et al., 1994). When the idea became more prominent that things that were not accessible via the Web needed to be given some sort of Web name, a new scheme called Uniform Resource Names (URNs) was invented “to serve as persistent, location-independent, resource identifiers” (Moats, 1997). URNs do not have to access anything, although they could possibly, since unlike URLs they were centralized through a registry and through this registry had experimental resolution services (Mealling and Daniels, 1999). URNs are more like natural language names like “Eiffel Tower” rather than an address like “Tour Eiffel Champ de Mars 75007 Paris.” With a name one can talk about the Eiffel Tower without actually having any idea how to get there, but with a location one can go up and bump a toe against the Eiffel Tower. A URN lets one refer to some thing, while a URL would let one access some bits on the Web. However, both URNs and URLs were subsumed under the single idea of a URI in RFC 3986, since a URL was just a URI that “in addition to identifying a resource” specifies a method “of locating the resource by describing its primary access mechanism (e.g., its network location)” while URNs are just another URI scheme, albeit a centralized one without an agreed-upon access mechanism (Berners-Lee et al., 2005).

The end result of this saga of URNs and URLs merging into URIs is that on the Web there is a single universal identification scheme for both identifying accessible and non-accessible resources. In this regard the Web is radically different from previous identification schemes. In programming languages, an identifier translates into the identity of some block of memory, even if there is no code that runs at that location. In other hypertext systems, one assumed that the unique identifiers were allowing links between accessible documents or some sort of file. Yet on the Web one can have a URI for the “Eiffel Tower in itself,” such as http://www.example.dingansich.org/EiffelTower. This brings up a new