

In Defense of Ambiguity

Patrick J. Hayes

Institute for Human and Machine Cognition

40 South Alcaniz Street

Pensacola, FL 32502

USA

and

Harry Halpin

School of Informatics

University of Edinburgh

2 Buccleuch Place

Edinburgh UK EH8 9LW

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.

THE IDENTITY CRISIS ON THE WEB

One might imagine that there would be no problems of identification on the Web. After all, the Web seems to be about identifying and accessing web pages. Yet the Web's ambition goes beyond normal hypertext systems content with accessing documents, for it wishes to use its identification system to identify objects that aren't even accessible via the Web. The ongoing problems and confusion that plague Web architecture follow from this goal. This is because there are two distinct relationships between names and things—reference and access.

The idea of a resource on the Web was from its outset universal: “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” (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 it. 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.singandich.org/EiffelTower>. This brings up a new type of problem for users, for if they access that URI, how do they know it identifies the Eiffel Tower itself and not just a Web page about the Eiffel Tower? Assuming it is useful to identify non-accessible things on the Web using URIs, should we distinguish between these two types of things and if so, how? Should a URI for “The Eiffel Tower itself” bear some special marking that makes it different from a URI that lets one access Web pages about the Eiffel Tower?

REFERENCE AND ACCESS

There are two distinct relationships between names and things: reference and access. The architecture of the Web determines access, but has no direct influence on reference. Identifiers like URIs can be considered types of names. It is important to distinguish these two possible different relationships between a name and a thing.

1. *accesses*, meaning that the name provides a causal pathway to the thing, perhaps mediated by the Web.
2. *refers to*, meaning that the name is being used to mention the thing.

Current practice in Web Architecture uses “identifies” to mean both or either of these, apparently in the belief that they are synonyms. They are not, and to think of them as being the same is to be profoundly confused. For example, when uttering the name “Eiffel Tower” one does not in anyway get magically transported to the Eiffel Tower. One can talk about it, have beliefs, plan a trip there, and otherwise have intentions about the Eiffel Tower, but the name has no causal path to the Eiffel Tower itself. In contrast, the URI <http://www.tour-eiffel.fr/> offers us access to a group of Web pages via an HTTP-compliant agent. A great deal of the muddle Web architecture finds itself in can be directly traced to this confusion between access and reference.

If one understands “access” as inclusively as possible, it includes uses of a URI to access a Web site, that is, an HTTP endpoint that produces a “representation” (in the REST sense) of a resource such as a Web page, a file, a mailbox, a webcam, a Coke machine connected to the Internet in Australia, and so on endlessly: anything at all that can receive, send or be directly influenced by, or indeed itself be, any piece of information that can be transferred by a transfer protocol, either now or in the foreseeable future. Cast this net as broadly as you like, the accessible things will always be an extremely small subset of the set of all things that can be referred to. Even in the possible world where every single thing that we normally consider a physical item was somehow given a URI and attached via some sort of chip that allowed the thing to send a byte back and forth across the Web, there would still be imaginary things and things in the future or past, and this class would be larger than the class of accessible things. Moreover, although one can of course refer to accessible things, most acts of reference will be to things that are not accessible, because most of the world's business is concerned with things other than the architecture of the Web, like the weather in Oaxaca. Most of the things in the world do not have a URI and are not accessible over the Web. Reference is not a secondary phenomenon, but a primary one.

Reference has to do with the semantics of language; access has to do with network architecture. Successful reference is part of a communicative act between cognitive agents capable of using

language. Successful access requires neither cognition nor linguistic communication; it is a purely causal interaction, not necessarily informational. Access is mediated by transmission over a network, and uses energy. The act of reference and its referent do not have a physical relationship at the time of use of a name. Although reference is built on top of robustly physical acts like uttering a name and some sort of neural processing, the cone of reference far outstrips local physical causation. One can use a very physical mouth and lips to communicate about a distant referent by uttering words such as “The Eiffel Tower is in France” when one is in Boston.

Reference is grounded in our physical orientation towards the world. However, it is still not scientifically or even philosophically understood how reference is established and communicated. In the classical analysis of Russell, reference can be established by acquaintance or by description (1911). A theory where a name directly refers to its referent is called a “direct theory of reference.” Russell states that “patches of sense-data” known through “direct acquaintance” allow one to ground the use of a name by attaching it to the physical world through the use of demonstratives. This grounding can be signaled by pointing or the use of demonstratives in natural language, such as when one approaches the Eiffel Tower, points at it and says “That is the Eiffel Tower.” According to Russell, a person has then identified the Eiffel Tower with that particular patch of sense-data, and so has directly “named” the Eiffel Tower. After that, the person can refer to the Eiffel Tower even if they are no longer in direct physical contact with it.

In contrast, description is inherently ambiguous. If a person is trying to identify the Eiffel Tower to a friend, then the person may attempt to communicate their thought about the Eiffel Tower by uttering a description such as “the monument in Paris.” Yet even the friend may think they are talking about the Arc de Triomphe without further information. If the person tries to give more descriptions, such as “the steel tower,” then the hearer might think of the Eiffel Tower, but there are no guarantees. The hearer may also think of the steel dome of Galeries Lafayette. Even if the person said, “the structure made by Gustave Eiffel,” the hearer may think of a lesser-known structure like La Ruche. One can imagine that with enough descriptions a person could uniquely pick out the referent for the hearer. Even with an infinite amount of descriptions this may be impossible, since it involves the large presumption that the hearer shares our same ontology of things in the world. The hearer may simply have no conception that the Eiffel Tower even exists, and so may be unable to grasp the referent regardless of the number of descriptions given. Surprisingly, adding more descriptions may even make the referent more ambiguous.

Reference by acquaintance can be done only to accessible entities. How can anyone directly refer to things they are not acquainted with, such as imaginary or historical objects? Many possible referents one may wish to refer to, one is not acquainted with by the constraints of life. One may simply not be able to afford a plane ticket to visit the Eiffel Tower, or it may not be possible to live long enough to visit a distant galaxy or survive visiting the edge of a black hole. References to non-accessible entities must be by description, and hence must be ambiguous.

Since URIs are used to refer to things, and reference is ambiguous, there is a problem with the slogan “a URI identifies one thing.” For the Semantic Web to work, URIs will have to refer to things by description, so a URI can not universally and unambiguously identify one thing. Or can it? Kripke’s “causal theory of naming” states that at least some proper names may be unambiguous (1980). This is precisely the account appealed to by some members of the W3C Technical Architecture Group such as Dan Connolly (2006). Kripke claims that proper names have the same referent regardless of description and in every possible world. Gustave Eiffel is not equivalent to the “architect of the Eiffel Tower” because

he would still be Gustave Eiffel even in the possible world where he did not create the Eiffel Tower. To briefly outline Kripke's theory, a historical causal chain between a current user of the proper name and past users allows the referent of a name to be transmitted unambiguously through time. The meaning of any sentence can then supposedly be dependent in an unambiguous manner on the referents of the names used in the sentence. The historical chain is established by a name being given its original referent through a process Kripke calls "baptism." Baptism is a combination of direct acquaintance with the referent and the action of naming the referent. Baptism is often quite literal. Gustave Eiffel got his name via an actual baptism.

Kripke's account of unambiguous names can then be transposed to the Web with a few minor variations (1980). In this story, a URI is like a proper name and baptism is given by the registration of the domain name, which gives a legally binding owner to a URI. The referent of a URI is established by fiat by the owner, and then can be communicated to others in a causal chain in the form of publishing documents at the URI (or a redirection thereof) or by creating Semantic Web data about the URI. In this manner, the owner of the URI can thereby determine the meaning of the URI. So, if one got a URI like <http://www.example.singandich.org/EiffelTower> and one wanted to know what the URI referred to, one could use a service such as *whois* to look up the owner of the URI, and then simply call them and ask them what the URI referred to.

On closer inspection, the causal theory of naming does not work on the Web. First of all, baptism seems to be a slight variation of Russell's story about establishing reference by acquaintance (1911). Russell's story is slightly problematic in itself, as it requires a strange "semantic Cartesianism" in that it assumes the mind directly associates names with patches of "sense-data" in the world (Luntley, 1999). This viewpoint is Cartesian insofar as it first assumes a discrete boundary between the world and the mind, but then posits some "idea" that can connect a name in the head with the world outside. Regardless of Russell, what the causal theory of names then leaves implicit is that establishing reference by acquaintance requires naming conventions. What Kripke correctly shows is that any naming convention like baptism is social. Yet baptizing then cannot function purely causally, since the naming convention requires communication between more than one person. The only way it could be purely causal would be if every person using the name had some form of direct acquaintance with the referent. The chain between the act of naming and the use of the name depends on reference. Ambiguities and errors can happen. For example, Gareth Evans pointed out famously that African natives may have used the word "Madagascar" to refer to the African mainland but when Marco Polo heard it, he thought it referred to an island off of Africa. What should be clear is that the ambiguity remains, and even Kripke himself says "a name refers to an object if there exists a chain of communication, stretching back to baptism, at each stage of which there was a successful intention to preserve reference" (1980). The notion of success is undefined, but it is clear that the chain is not just one of causation, but communication and so subject to ambiguity since communication is often due to description.

However, Kripke leaves us an important insight, that naming is a social act, although Kripke himself would be unlikely to endorse this view. The social nature of naming applies to URIs on the Web as much as it does to proper names in natural language. The use of domain name registration to "own" a URI is just a social act with a technical infrastructure that happens to support it. It is possible to argue that the domain name system somehow builds the referents of URIs into the social contract implicit in buying a top-level domain name, so that if one is using the Web, then one is in effect part of a name-using community. Under this assumption, if a user were given the URI

<http://www.example.singandich.org/EiffelTower> then the user would be part of the community of the Web and the user is then forced to buy into the owner's claim that the URI refers to the Eiffel Tower. This argument is trivially not true on the Web. The owner cannot communicate via telepathy what the URI refers to. In a decentralized system such as the Web, a user of the URI can usually tell what a URI is supposed to refer to by accessing Web pages through the URI, and Web pages are another form of description for things and so subject to ambiguity. Even if the owner of the URI somehow “knows” what the URI is supposed to refer to, the referent of the URI cannot be communicated unambiguously, for the owner must communicate the referent using descriptions in either natural language or Web pages. The owner of the URI has even further problems, for most of the things they will want to refer to they cannot know through direct acquaintance, and so their own understanding is ambiguous. Practically, many people use URIs where they do not own the top-level domain, and so most people do not “control” what their URI refers to on the domain-name level.

Interestingly, the use of URIs as referring mechanisms reveals a fatal flaw in assuming that any naming convention is universal. One reason that examples used in declaring that naming is unambiguous use famous names like “Cicero” or “Gustave Eiffel” is because the reader of the example is already in the naming-using community of that particular name. For names of not well-known people like “Kavita Thomas” or URIs like <http://www.example.singandich.org/EiffelTower> the famous name convention does not hold, and so it becomes evident that a universal naming convention cannot be assumed. Furthermore, for people there is a clear and legal process of baptism. For things like the “Eiffel Tower” or “the integers” it becomes much harder to find a clear baptism. Any naming convention must be established, not assumed.

REFERENCE IS INHERENTLY AMBIGUOUS

Access is dependent on architecture. It is precisely by providing a space of names for access that Web architecture is useful. In order to be useful, access should be unambiguous. In contrast, reference to natural entities is inherently ambiguous. In this manner, reference on the Web is the same as reference off the Web. This is simply obvious. The Web is a transport mechanism for (what are in the REST sense) representations such as web pages. What a representation represents, and how the names in it refer, has nothing particularly to do with how the representations are transported. In the words of Korzybsky, “the map is not the territory” (1931). Web architecture does not determine what any names, including URIs, refer to. It only determines what they access.

The relationship between access and reference is essentially arbitrary; it is ours to decide and cannot be decided by Web architecture. Since most of the things referred to by names are not accessible, references to them can only be determined by description, perhaps based on other preexisting naming conventions. If a URI is intended to refer to Gustave Eiffel, this reference cannot be established by acquaintance, since Gustave Eiffel isn't accessible today. It has to be done by description and descriptions can never pin down a referent exactly. There will always be some slack, some possible doubt about what exactly is being referred to. As said before, reference by description is inherently ambiguous. This claim may seem to fly in the face of common sense, for it seems clear that language does use names successfully to refer. What we need is a criterion for what “success” in communicating what a name refers to could be.

A name refers successfully when a use of the name is sufficient to communicate a thought about the referent during an act of communication. The actual processes which constitute having a

thought about a referent in human communication are mysterious. The best formal account we have, which also applies to the Semantic Web, says that this is a process of inference. The same considerations apply whether the agents involved are humans speaking English or Web agents drawing conclusions in OWL. It is difficult, perhaps impossible, to make enough inferences so as to be sure that what the recipient understands a name to refer to is exactly the same thing as what the sender had in mind since communication using referring names is always hostage to slips of reference. A formal version of this claim is based on Gödel's theorem: as long as the language used for communication is sufficient to express arithmetic, it will have "nonstandard" models. The point is made better informally by looking at normal human communication. Take the most direct and unambiguous kind of name, a famous proper name of a public entity such as "Paris." This might refer to the central part of greater Paris defined by the arrondissements, or to the larger metropolitan area. It might refer to the state of the city now, or it might refer to the entire history of the city. In the right context, it can be used to refer to the inhabitants of Paris, the buildings in Paris, the customs of Paris, and so forth. Another example: "Everest." How much does Everest weigh? Where are its edges? In most instances it appears that communication can be successful without the details of precisely what referent is being referred to being resolved.

We have left unanswered the question how successful reference is achieved using names. The problem with any direct theory of reference lies precisely in the assumption that reference must be resolved for reference to be meaningful. There is an alternative to both the direct theory of reference and the causal theory of naming that allows for ambiguous names to have meaning and be used successfully while maintaining the ambiguity of reference. The key idea lies in the observation made by Frege that the meaning of any term, including names, is determined by what Frege calls the "sense" of the sentences that use the term (1892). According to Frege, two sentences could be the same only if they shared the same sense. His example uses the two statements "Hesperus is the Evening Star" and "Phosphorus is the Morning Star." To the ancient Greeks these two sentences have different senses even if they had the same referent, the planet Venus, since the ancient Greeks did not know that the referent was the same, for they did not know that "Hesperus is Phosphorus" (Frege, 1892).

Previously we have presupposed that the meaning of a name is determined by whatever thing the name refers to. The meaning of sentence "Dan Connolly works for the World Wide Web Consortium" would be determined by first figuring out what person "Dan Connolly" refers to, and then figuring out what organization the "World Wide Web Consortium" refers to. Only after calculating the referents of the names can one build up a meaning of the sentence, which would then theoretically be precise and unambiguous. We have to be careful precisely because the word "meaning" is one of the most ambiguous, confusing, and debated terms in the English language. From here on out, it would be better to think of the "meaning" of a name to be defined by the "meaning" of sentences that use the name (Luntley, 1999). So in contrast to our previous example, the meaning of the sentence "Dan Connolly works for the World Wide Web Consortium" determines the meaning of "Dan Connolly" and "the World Wide Web Consortium." After someone utters that sentence, we can reasonably believe that "Dan Connolly" in that sentence is not the same "Dan Connolly" that plays football for the New England Patriots. If we hear enough sentences about "Dan Connolly" we can get a better sense of what "Dan Connolly" means. The meaning of the name "Dan Connolly" is not determined by the meaning of a single sentence or set of descriptions, but the normative use of all the sentences in the language that bear upon "Dan Connolly." Assuming that the meaning of a sentence is determined by simply identifying the referents of the terms gets the picture backwards. The possible referents of a name are always ambiguous, but are defined as much as they need to be by the use of the name in sentences. Names simply make no sense by themselves. We don't

even need to be directly acquainted with something to give it a name. We can just use the name in sentences, and the meaning is created by the name-using community that shares those sentences. We can also understand “sentences” to mean not just natural language sentences but also Semantic Web statements and the non-linguistic dimensions of communication. Uttering “Dan Connolly!” in exasperation after a discussion means something different from yelling his name while trying to get his attention in a busy airport. What determines the meaning of sentences is the “sense” of the sentences, which is constructed from the patterns of use of sentences in a language and the behavior those sentences engender. This includes the inferences one can draw from a sentence.

The wrong picture of meaning on the Semantic Web would be that of a giant decentralized dictionary that listed a URI and its referent. Since a URI can refer to non-accessible things, we cannot have the complete list of URI and referents. The URI <http://www.example.singandich.org/EiffelTower> can not give us the Eiffel Tower itself. A URI could be given a “definition” in natural language like the English words “The Eiffel Tower.” However, looking up a URI and getting natural language terms or sentences is just not too much help if one was translating a language one does not already know, such as “the language of a hitherto untouched people” or a hitherto unused form of computing language like the Semantic Web (Quine, 1960). How does one know if the URI would reference only the monument itself or copies of the monument like the one in Paris, Texas? We would have to assume the “principle of charity” that the URI means the simplest thing possible that explains its usage. So the translator can never “really” know unambiguously what the term means. This is Quine's radical indeterminacy of translation, and it defeats any attempt to think of a language as a simple dictionary (1960).

How would we really know what a URI meant? The solution to this problem is to employ the argument of “radical interpretation” from Davidson (1973). Davidson constructs his notion of radical interpretation as a semantic version of Quine's argument of radical translation. Davidson notes that radical translation “deals with the wrong topic, the relation between two languages, where what is wanted is an interpretation of one” (1973). Unlike a dictionary theory of meaning, which is merely the translation of representations (syntactic languages) from one to another, the construction of the meaning of those representations (a meta-theory) is what in turn lets a translation be possible. Some of the construction of this meta-theory, which should take the form of the ability to assign Tarski-style truth conditions to statements, can rely on the grammar of language. “Dan Connolly works for the World Wide Web Consortium and lives in Kansas” can be explained by analysis of the connective “and” and the two sub-sentences. However, at some point we bottom out in sentences that have meaning by themselves even if the referents of each term therein is ambiguous. The main point again is that it is impossible for a single name by itself to mean anything except in its relationship to other names and its use in sentences. So, the meaning does not lie purely in the terms, but in the language itself, where a language consists of both the rules of grammatical construction, which can be formalized, and the meaning of sentences. The only way to learn a language is to be competent in it, to participate in its usage, and to assume other speakers share a rational point of view on the world. In order to have radical interpretation of the language of a visiting alien we must assign “truth conditions to alien sentences that make native speakers right when plausibly possible, according, of course, to our own view of the right. What justifies the procedure is the fact that disagreement and agreement alike are intelligible only against a background of massive agreement” (Davidson, 1973).

In this manner the often-used but little-explained idea of the social aspect of naming

conventions brought up implicitly, although far from endorsed, by Kripke become clear. This social aspect just means accepting a background of massive agreement when understanding names and sentences and accepting the norms of the community in using those names and sentences. If someone said “Dan Connolly thinks that data types in XML should be context independent,” one assumes that this is the same Dan Connolly that works at the World Wide Web Consortium. If someone said, “Dan Connolly is Spiderman,” one might assume that the background assumptions that you and that speaker share are radically different. When you use a language successfully we can usually assume other people share agreement without relying on the impossible task of unambiguously connecting names to referents.

Ordinary discourse is full of ambiguities which are rarely noted because they have no practical importance, but which are rendered vivid by trying to agree about “common sense” well enough to write it down in a formal notation. One example is a disagreement about whether or not a fitted carpet was “in” an office or “part of” the office. Two competent, intelligent adult native speakers of English each discovered, to their mutual amazement, that the other would believe what they thought was an obviously false claim. Over an hour of discussion it gradually emerged, by a process of induction from many examples, that they understood the meaning of “office” differently. For one it meant, roughly, an inhabitable place; for the other, something like a volume of space defined by the architectural walls. These two people never knew, until this event, that they had different mental meanings for “office” and more generally, for “room.” Presumably this was possible because they had never previously engaged in a communicative act in which their conceptual differences made any difference to the communication. Their success notwithstanding, each of them had, in fact, been understanding the English word “office” differently. In this example, “office” is ambiguous, and this can equally apply to other words in natural language. No doubt the same logic can be applied to any language on the Web that uses reference.

Saying more can make reference by description even more ambiguous. Adding richer formal ontologies to a notion does not reduce ambiguity of reference, but increases it by providing for finer ontological distinctions. If all one wants to say about persons is that they have mailboxes and friends, then one can treat “person” as a simple category. Even when a stable situation of mutual reference has been reached it can be upset by the addition of new vocabulary. If one wishes to include persons in a comprehensive upper-level ontology, then more arcane questions about personhood and how it relates to existence must be answered. Suppose two agents agree that “Tim Berners-Lee” refers to a particular person. Still, they might have divergent notions of what it means to be a person. Such divergences are already found in “standard” ontologies. For example, DOLCE requires a high-level distinction to be made between continuants and occurrents, and a person would naturally be classed as a continuant; but other ontologies reject this contrast and subsume all temporal entities under one heading (Gangemi et al., 2002). So are all names of persons rendered ambiguous by the presence of this high-level ontological divergence of opinion, so that we have to distinguish Tim Berners-Lee the continuant from Tim Berners-Lee the four dimensional history? For formal eng purposes, the difference is important, and indeed confusion between these concepts can produce immediate logical contradictions, such as inferring that Berners-Lee is both 52 years old and 7 years old. This error can arise because Berners-Lee was once seven years old, and a continuant retains its identity through time. Do we need Tim Berners-Lee Lite? Do we really want to require all reasoners to agree on the most abstract of philosophical distinctions in order to have successful inference? Many reasoners are simply not concerned with this distinction.

One disturbing result of this is that we may need different versions of the concept of a “person.” This question is not just a question of technically determining what the proper ontological

description of a “person” is. If one wants to reason about people's lifestyles, travel plans, medical histories and legal rights, then one is obliged to distinguish finer categories of personhood that in turn rely on social and legal constructions with many infamously difficult and even contested distinctions. The more one is able to say and is obliged to reason about, the more possible distinctions there are to be made. As the knowledge increases in scope, the more possibilities arise for making distinctions that were formerly impossible.

What makes this kind of consideration particularly acute is the view that URIs should be global and eternal in scope. This makes things worse. It means that if there is any possibility of some such ontological distinction being made by anyone, anywhere, at any time, then in order to avoid ambiguity of reference, the URI must make all the potentially disambiguating decisions ahead of time. This is of course manifestly impossible, because there is always the possibility of some new distinction being made later. It is impossible to achieve unambiguous universal reference of names by using descriptions. So we should not set out to attempt unambiguous reference, nor pose it as a goal or a “good practice.”

IF AMBIGUITY IS INEVITABLE, LET'S MAKE LEMONADE

Since reference by description is inherently ambiguous, it might be better to set out to utilize the inevitable rather than try to minimize it. The “Identity Crisis” is said to result from phenomena of using the same URI to refer to Tim Berners-Lee and his Web site, the Eiffel Tower and a picture of the Eiffel Tower, the weather and the weather report. These are all examples of overloading, using a single name to refer to multiple referents. Overloading isn't always as bad as it is rumored to be, since it is a way of using names efficiently. Natural language is rife with lexical ambiguity which does not hinder normal communication, as the multiple meanings of even common words like “rose” and “bank” show. Even programming languages routinely do it, as for example many programming languages use “+” to mean both integer and real addition. Using overloading does not violate any principle of formal semantics, as Common Logic allows a single name to denote an individual, a function and a relation (Delugach, 2007). In general, overloading is harmless when it is possible to figure out from the information available at the time which of the various referents is intended by that particular use of the name. Often this is easy when the referents are in different ontological categories, as with a person and a web page. On a social networking site to befriend a Web page is odd, but not to befriend a “person.” One way to think about overloading is that an overloaded term denotes a single “complex thing” which has all the usual referents as parts or facets, selected by implicit selectors. This often works surprisingly well. The syntax of logic is enough to select the appropriate referent in the Common Logic formal semantics when a single name could be used to refer to an individual, a function, or a relation. For people and Web sites it seems to work, for usually Berners-Lee is not affected by an attempt to HTTP GET him instead of his Web site, and his Web site isn't going to spend his salary. The normal infrastructures of interaction with people and Web sites are sufficiently robust to disambiguate this kind of overloading when they need to be disambiguated by humans.

As pointed out by Connolly, there can be problems in inference due to ambiguity (2006). While there is not room to go through the details of his example here, we will reconstruct a similar example. Imagine an ontology where one defines Web pages and people to be disjoint. However, because of the decentralized nature of the Web, a person can use the URI that accesses their Web page to make statements about themselves using an ontology for people. Let's assume that Dan Connolly has two URIs. The first <http://www.w3.org/People/Connolly/> accesses his Web page. The second

http://www.example.org/Dan_Connolly# refers to Dan Connolly as a person. Dan Connolly then carefully makes statements about his Web page like “<http://www.w3.org/People/Connolly/> was created in 1994.” He also makes statements about himself, like “http://www.example.org/Dan_Connolly# works for the W3C.” However, because the Web is a decentralized system and there is no way to tell what http://www.example.org/Dan_Connolly# refers to, someone makes a statement on the web like “<http://www.w3.org/People/Connolly/> lives in Kansas.” Since earlier it was defined that Web pages and people are disjoint, and “lives” is a predicate that applies to living people and not Web pages, we get a contradiction when we run an OWL reasoner. While humans can sort out this ambiguity of reference machines can not. However, is this a problem for inference? It actually isn’t a problem with inference per se, it’s a problem with stating ahead of time two things are disjoint, a sort of sneaky use of the Law of the Excluded Middle on the Web. However, assuming one wants to keep that sort of statement possible in Web languages like OWL, then one could claim that the problem is some person is stating incorrectly and implicitly that <http://www.w3.org/People/Connolly/> is a person instead of a Web page. Yet what if Dan Connolly himself attempted to state that <http://www.w3.org/People/Connolly/> lives in Kansas?” Who could stop him?

A Web site accessible from a URI could allow the owner to describe how they want their URI to be used. This would be using the one advantage the Web has over previous languages, the ability to access representations via URIs. One can reconsider the issue of radical interpretation. Given a URI, how can one tell what the owner thinks the URI is supposed to refer to? The answer of Davidson is to look at its systematic and normative use in the entire language, including Web languages like Web pages and Semantic Web statements (1973). Of course one is only going to observe a small portion of its use, so ambiguity remains. One way the owner of the URI can help clarify the situation is by making representations for both humans and machines accessible from the URI itself. If the user of the URI gets these representations that are accessible from the URI itself, they may have a better idea of how the owner of the URI at least intends their URI to be used. If the owner intends the URI to be used for reference to a non-accessible thing, the same URI can be used for both reference to a non-accessible thing and hosting accessible representations about the non-accessible thing. Yet in order to communicate about a non-accessible thing, the fundamental confusion between reference and access returns.

The Technical Architecture Group (TAG) of the W3C took on the “identity crisis,” calling it the *httpRange-14* problem. They phrased it as the question “What is the range of the HTTP dereference function?” In their solution, they define a class of resources on the Web called “information resources,” which they specify as a resource “whose essential characteristics can be conveyed in a message.” (Jacobs and Walsh, 2004). This definition is unclear at best. While a Web page is clearly an information resource, is the text of Moby Dick an information resource? Is the World Wide Web Consortium itself an information resource? Yet what the term “information resource” seems to be groping towards is that some things are accessible on the Web and others are not. It would be better for things that are accessible on the Web to be called by a term like “Web resources.” It is possible that the term “information resource” was chosen to define something that in some possible world could be accessible on the Web. “The text of Moby Dick” could be an information resource, because even if the complete text of Moby Dick isn’t on the Web, one day it might be. However, a “particular collector’s edition of Moby Dick” could not be, since after all, the part that is collectible isn’t the text, but the physical book itself. The actual hackers and webmasters on the streets need a term like “Web resource” that they can use to mean something like “accessible via the Web” (such as by being an HTTP end point) without arguing metaphysics about the “essential nature” of Moby Dick every time they want to add a URI to the Semantic

Web.

Despite their invention of new terminology, the TAG's solution has two desiderata, both of which are respectable. Even if overloading is harmless most of the time, there should be some mechanism that is able to distinguish between the use of a URI to access a thing and the intended use of a URI to refer to a non-accessible thing. Furthermore, even for URIs that are intended to refer to non-accessible things, there should be a way to access descriptions to help users of the URI make sense of the resource. What the TAG resolves is less than ideal. In the case of a non-information resource, a non-accessible thing that is referred to using a URI, an agent should not get any sort of representation without a redirection to another URI resource. This disambiguation is done through the "303 See Other" HTTP header. So the official resolution to "Identity Crisis" is given in terms of HTTP response headers:

1. If an HTTP resource responds to a GET request with a 2xx response, then the resource identified by that URI is an information resource;
2. If an HTTP resource responds to a GET request with a 303 (See Other) response, then the resource identified by that URI could be any resource;
3. If an HTTP resource responds to a GET request with a 4xx (error) response, then the nature of the resource is unknown.

To give an example, let's say our user is trying to access a URI that is "supposed to" refer to the Eiffel Tower like <http://www.example.singandich.org/EiffelTower>. Upon attempting to access that resource with an *HTTP* GET request, since the Eiffel Tower itself can not be an HTTP endpoint, one should not host any representations there. Instead, the user gets a 303 response code that says "See Other" that in turn redirects them to another URI that hosts Web pages about the Eiffel Tower such as our trustworthy <http://www.tour-eiffel.fr/>. When this URI returns the 200 status code in response to an HTTP GET request, we know that <http://www.tour-eiffel.fr/> is actually an "information resource." The URI used to refer, <http://www.example.singandich.org/EiffelTower>, could be any kind of resource. That is, we know what we did before the 303 redirection, namely that <http://www.tour-eiffel.fr/> can be used to access a web page. The agent knows nothing about the original URI, <http://www.example.singandich.org/EiffelTower>. So this method fails actually to deliver on any sort of distinction between those things that are information resources and those that aren't. The reason that the use of the 303 status code can not possibly tell us that the resource redirected from was used for referring, arises because the 303 status code was specified as "See Other" before the "Identity Crisis" was even noticed. There is no reason why it can't be used to simply to redirect from one "information resource" to another "information resource."

In practice, web architecture does not determine what any names, including URIs, refer to. It only determines what they access. The relationship of reference is determined by the users of the URI. The use of 303 redirection seems to presume that if a URI accesses something directly (not through an *HTTP* redirect) then the URI must refer to what it accesses. This presumes, wrongly, that the distinction between access and reference is based on the distinction between accessible and inaccessible referents. There is no law of the universe that prevents a URI that accesses one referent, such as "<http://www.amazon.com>," from also being used to refer to a particular company in everyday speech. It places the responsibility for deciding the relationship between referring and accessing at the wrong end of the communication channel, that of the person who hosts representations accessible at the URI, not the user of the URI. While everyone could do precisely what the TAG recommends by using 303 redirection, there is no architectural way to enforce this doctrine, so instead people will do what is convenient and works in

practice, and 303 redirection is far from convenient.

Pragmatically, there are problems with the TAG's suggested redirection. It uses a distinction in how a text is delivered (an HTTP code) to disambiguate the accessible Web page itself; a category mistake analogous to requiring the postman dance a jig when delivering an official letter. Since the vast majority of names, even on the Web, refer to things which are not accessible, this requires referring URIs to perform an act of redirection with doubtful benefit. As shown earlier, since the URI bears no trace of its delivery to the majority of human Web users that do not monitor or understand HTTP status codes, no disambiguation is achieved for the human. The TAG is correct in noticing this solution could solve the problem of inference brought up by Connolly (2006), but it does so in such a manner that not only makes normally harmless overloading illegal but that does not even make the distinction between access and reference clear. The particular solution requires the use of an arcane redirection technique that most people actually hosting URIs are not familiar with and cannot even deploy, since deploying 303 redirection requires access to the web server many users may not have. It also produces harmful effects by misusing HTTP codes for an alien purpose. The particular code, 303, is only valid for HTTP 1.1 and was originally introduced to solve a completely different problem. As put by the specification, “this method exists primarily to allow the output of a POST-activated script to redirect the user agent to a selected resource,” not to distinguish access and reference (Fielding et al., 1999). The 303 status code was invented due to the over-use of the HTTP 1.0 302 status code to redirect both temporarily and permanently. The 307 and 303 status codes in HTTP 1.1 could disambiguate between the two cases of redirection, with the 303 status code having future requests to that URI being automatically redirected by the browser unlike the 307 status code, which is only a “temporary” redirection. Given this history, it is unclear why 303 is suitable for distinguishing between access and reference. Why not just invent a new HTTP status code? The negative effects of this redirection requirement will continue and achieve little in return.

The main alternative to using HTTP 303 is to have a fragment identifier—the hash—attached to a URI to get redirection for free. So, if one wanted a URI that referred to the Eiffel Tower itself without the hassle of a 303 redirection, one would use the URI <http://www.tour-eiffel.fr/#> to refer to the Eiffel Tower and the URI <http://www.tour-eiffel.fr/> to access a Web page about the Eiffel Tower. Since browsers think the “#” at the end of the URI means a fragment of a document or some other representation, if a user tries to access via HTTP GET a “hash URI” it will not return a “404 Not Found” status code, but instead simply resolve to the URI before the hash. In this way machine reasoners can keep the URI that refers to the Eiffel Tower and a Web page about the Eiffel Tower separate, while a human can access the URI “about” the Eiffel Tower and receive some information about it, in essence by taking advantage of some predefined behavior in web browsers. This solution would solve the inference problem where monuments and Web pages are defined in OWL as disjoint. This is valid because according to the W3C TAG's “Architecture of the Web,” using a fragment identifier technically also identifies a separate and distinct “secondary resource” (Jacobs and Walsh, 2004). Further, the TAG states that “primary and secondary simply indicate that there is a relationship between the resources for the purposes of one URI: the URI with a fragment identifier. Any resource can be identified as a secondary resource” (Jacobs and Walsh, 2004). Yet using hash URIs has the exact same problem as 303 redirection, since it doesn't normatively define any sort of relationship between the two URIs, much less distinguish between access and reference.

It appears that the W3C may very well be contradicting the relevant IETF specification by supporting hash URIs. The URI specification says “the semantics of a fragment identifier

are defined by the set of representations that might result from a retrieval action on the primary resource. The fragment's format and resolution is therefore dependent on the media type of a potentially retrieved representation, even though such a retrieval is only performed if the URI is dereferenced" (Berners-Lee et al., 2005). If the media type explicitly defines what fragment identifiers do, then the user should obey the standard of the media type. Only "if no such representation exists, then the semantics of the fragment are considered unknown and are effectively unconstrained" (Berners-Lee et al., 2005). In other words, only if you get a 404 from <http://www.tour-eiffel.fr/> can <http://www.tour-eiffel.fr/#> mean anything you want. However, if a Web page with the "text/html" media type is returned by accessing the primary (no hash) URI, then according to the HTML specification, "for documents labeled as text/html, the fragment identifier designates the correspondingly named element; any element may be named with the *id* attribute" (Connolly, 2000). In other words, fragment identifiers should be used for named elements in the document, not as a shortcut for distinguishing URIs used for reference and access. This defeats the entire purpose of using hash URIs, since the supposed benefit is that a human can "follow-their-nose" by accessing the primary URI and thereby access some human readable HTML about the URI. In the case where the "application/rdf+xml" media type is returned by the accessible URI, things are different. "In RDF, the thing identified by a URI with fragment identifier does not necessarily bear any particular relationship to the thing identified by the URI alone" so the hash convention can legitimately identify anything, including non-accessible resources (Schwartz, 2004). This seems to defeat the point of returning representations, since unlike rendered HTML, RDF/XML is much more easily used by machines than humans. If people accessed <http://www.tour-eiffel.fr/> and received RDF/XML most would have no idea what to do with it. It is most useful for machine processing, not informing humans.

Strangely enough, the very idea that a media type determines the semantics of the fragment identifier is in conflict with other statements from the W3C. Even if one accepted the "URI identifies one thing" slogan, by using content negotiation, if both the "application/rdf+xml" and "text/html" media types were available for a URI, then the meaning of the URI with fragment identifier would be interpreted two different ways depending on the media type received, and so the URI would not identify a single resource with a global scope. This fundamentally breaks the orthogonality of the specifications, as a single resource can return different kinds of representations, so how a "hash URI" can be used is dependent on media types. The URI specification explicitly says one should not do this, for "whatever is identified by the fragment should be consistent across all those representations" (Berners-Lee et al., 2005). One could imagine the hash somehow being consistent across representations, but if the fragment identifier exists in a RDF document and in the HTML document, the meaning of the fragment identifier will be muddled since it will identify both a portion of a document in HTML and possibly some non-Web accessible thing. In cases where the fragment identifier exists in RDF and not in HTML, it will be a broken fragment identifier for an HTML document and perhaps specified by the RDF, and so inconsistent. If the fragment identifier is non-existent in both the RDF and HTML documents, in RDF the fragment identifier can identify a non-Web accessible resource but not so in the HTML document, where it will just be a broken fragment identifier for a particular document. Regardless, there needs to be a mechanism in HTML for saying that either the given use of a fragment identifier is for non-Web accessible things or that fragment identifiers that are not given by the HTML representation can be anything, including non-Web accessible things. So, this use of fragment identifiers to identify non Web-accessible things, while convenient and much more practical than 303 redirection, is as far from "a URI identifies one thing" as one can get. One can assume that at some point the W3C will fix the relevant specifications to be more inline with their proposed solutions, but the hash URI is no panacea for distinguishing access and reference. While easier for users to deploy than 303 redirection, it still does not distinguish access and reference any better than 303

redirection.

DISTINGUISHING BETWEEN REFERENCE AND ACCESS ON THE WEB

We should first examine our presumptions. Using the same URI to refer to the Eiffel Tower and to access a Web site about the Eiffel Tower isn't even an example of overloading. It only becomes that if we assume that the name which accesses the Web site therefore also refers to the Web site. Why do we make this assumption? It would be possible to lose the assumption that access means reference on the Web. One could state that URIs only refer to accessible things just when the accessible thing is actually assigned that name; and assigning a name is done only by an explicit naming convention, the Web equivalent of pointing to the thing and giving it a name. There are two ways to attach a name to a thing: by being it or by naming a URI that accesses it. The two kinds of naming convention this makes possible are similar, respectively, to wearing a name badge and to having someone point at you and say, "I will call you Spiderman." Unlike reference by description, this kind of naming can be unambiguous under certain conditions. In particular, that the naming convention should be shared and the accessible entity itself has the naming convention attached. In that case, assigning the name to an accessible entity inside the entity itself makes the name "permanent" and global in scope. Named graphs fulfill this, as do properties in the header of HTML document, and both are under the control of the owner of the document. However, we are still dodging the fundamental question, since the class of referable things far outweighs the class of accessible things, and these things may need to be distinguished, we need some actual mechanism for assigning referents to names. Right now there aren't any, because access isn't the same thing as reference.

The solution would be just to have some sort of statement that explicitly stated a URI was being used for reference and associated descriptions with that URI. This "naming convention" needs to be unambiguous, easily deployed, and allow access to representations. In theory, the solution can be solved on a number of levels by allowing the URI, in the HTTP header status codes, or in the representation itself. The basic naming convention could be a RDF predicate called *ex:refersTo* whose subject was the URI that had accessible descriptions of the thing and whose object was the URI being used as a referring name for a thing. The natural inverse would be an RDF predicate *ex:describedBy* whose subject would be a URI used as a name for a thing and whose object would be the URI that had accessible descriptions. It seems intuitively obvious the subject of *ex:refersTo* and the object of *ex:describedBy* should be Web-accessible resources like Web pages, what the TAG might call "information resources." It is not clear if the reverse should apply. Should the object of *ex:refersTo* and subject of *ex:describedBy* be non-Web-accessible things, or what the TAG might call "non-information resources?" While at first this may seem to be the right choice, it would be a premature slicing of the Web into two classes of things. After all, there is nothing to prevent a URI for a Web page from being used as a referring term, or for a Web page to be described by another Web page. In other words, in order to allow flexibility of reference, the object of *ex:refersTo* and the subject of *ex:describedBy* should be anything, whether that thing is Web accessible or not. The question then might return about how we specify that a referent is non-accessible. The obvious solution is to create a class, *ex:nonAccessible*, for non-Web-accessible things. This class can then be applied as needed to particular objects of *ex:refersTo* and particular subjects of *ex:describedBy*.

Interestingly enough, the definition of these predicates would be weaker than *owl:sameAs* but more powerful than *rdfs:seeAlso*. The proposed *ex:refersTo* and *ex:describedBy* are somewhat similar to *foaf:topic* and *foaf:page*, respectively. The obvious advantage of having an explicit RDF predicate is that distinguishing between access and reference would almost always be a distinction

only made on the Semantic Web, so RDF is a much more natural technology than HTTP response headers for making the distinction. Furthermore, this sort of RDF predicate can then be easily embedded in HTML documents using techniques such as RDFa and GRDDL. This allows accessible representations such as Web pages to say what “thing” they describe if the “thing” they refer to happens to have a separate URI without violating any existing specifications. Unlike both the hash URIs and 303 redirection, this resource specifies precisely when a URI is used for access and when a URI is used for reference. Further, since there is no reason to define these classes as disjoint, it will not cause any ad hoc problems in inference but will merely enrich the results.

So, we can now outline how we would in a disciplined manner solve our problem about distinguishing referring to the Eiffel Tower and accessing a Web page with representations of the Eiffel Tower. First, this distinction would be necessary if one was building a directory of tourist attraction Web sites. Since there are multiple Web sites about the Eiffel Tower, one could connect them by having in RDF that all of these Web sites refer to the same tourist attraction, so there is a concrete need to both refer to the Eiffel Tower itself and to refer to the Web page(s) about the Eiffel Tower. This distinction could even be clarified in an ontology that classifies Web pages and monuments as disjoint. In the tourist directory, the Web sites would be grouped by URIs about particular attractions, such as the Eiffel Tower. One mints <http://www.example.tourism.org/EiffelTower/#> to refer to the Eiffel Tower. At <http://www.example.tourism.org/EiffelTower> would be hosted a RDF/XML document that contained the statement “<http://www.example.tourism.org/EiffelTower/#> *ex:describedBy* <http://www.tour-eiffel.fr/>.” This single statement allows the Semantic Web to distinguish reference and access in a manner at least unambiguous to RDF. Furthermore, by hosting an RDF/XML document at <http://www.example.tourism.org/EiffelTower/> we keep the semantics of the fragment identifier of <http://www.example.tourism.org/EiffelTower/#> unconstrained. We can then ask for “back-link” from the various Eiffel Tower–describing Web sites, which then using RDFa can embed their inverse statement, for example: “<http://www.tour-eiffel.fr> *ex:refersTo* <http://www.example.tourism.org/EiffelTower/#>.”

One argument against this solution is that people may not have access to the representation itself. There is no reason why something semantically identical with *ex:describedBy* cannot be using other formats. For example, one could imagine the use of a “link” element in the header of a HTML document to specify *ex:refersTo*. One could also use a revived “Link” HTTP header, as was included in an earlier version of HTTP, IETF RFC 2068 (Fielding et al., 1997). but was left out of the latest version, IETF RC 2616 (Fielding et al., 1999). So reintroducing the “Link” header would not be adding anything new to HTTP, merely using a neglected feature. The main issue with the “Link” header is that it would lack the semantics to distinguish reference and access. In other words, it’s just a link from one resource to another, and makes no statement about whether one resource is accessible or not. The use of a “Link” header could be given proper semantics if it could be used in combination with a profile header or link relations that allowed the type of the “link” to be specified as either for reference or access. The URI for reference and access used in the profile header could be as simple as reusing URIs from the RDF as the profile value or link relation such as “<http://www.example.org/refersTo>” and “<http://www.example.org/describedBy>” (in our previous examples, the namespace <http://www.example.org/> has just been abbreviated as “*ex*”). If we are beginning to edit specifications, one could also dictate that a URI ending in just a fragment identifier indicates that the URI should be used for reference regardless of the media type, thus legalizing the use of hash URIs for reference in RDF. There are cases where accessing and actually editing a representation for adding in the proper RDF statement could be difficult, and so the use of a combination of a “Link” and link relations with URIs could be crucial.

Imagine the organization in charge of <http://www.tour-eiffel.fr/> may not want to disturb their delicate HTML by inserting RDFa statements inside of it. Perhaps their Web page is made by a Web designer who specializes in graphics. However, they have been swayed by their technical engineer that making their Web page part of the Semantic Web would make their site be more useful and popular. The tourist attraction directory is having difficulty attracting visitors, since the visitors cannot make heads or tails of the RDF/XML documents accessible from <http://www.example.tourism.org/EiffelTower/>. Assuming a world where “Link” and URIs with link relations are now valid parts of HTTP, we can fix this problem without altering any representation. We can have a HTML Web page accessible from <http://www.example.tourism.org/EiffelTower/> by having the “Link” header use the URI <http://www.example.tourism.org/EiffelTower/index.html>, which is just a human-readable Web page, and by using <http://www.example.org/describedBy> in the link relation. Then a human user could be redirected to the human-readable Web page via a 303 redirection and it would be possible for them to know why they are being redirected there. Likewise, the webmaster of the Eiffel Tower site can then have a link to the Semantic Web URI by using the “Link” header to specify <http://www.example.tourism.org/EiffelTower/#> and then specifying its semantics by the use of the <http://www.example.org/refersTo> as the link relation. In this way, no HTML has to be changed but the links are still there. A machine could follow this links to discover more information while keeping the URIs for access and reference distinct, while a human could always get a human-readable version.

Regardless of the details, the use of any technology in Web architecture to distinguish between access and reference, including our proposed *ex:refersTo* and *ex:describedBy*, does nothing more than allow the author of a URI to explain how they would like the URI to be used. Ultimately, there is nothing that Web architecture can do to prevent a URI from being used to refer to some thing non-accessible. However, at least having a clear and coherent device, such as a few RDF predicates, would allow the distinction to be made so the author could give guidance on what they believe best practice for their URI would be. This would vastly improve the situation from where it is today, where this distinction is impossible. The philosophical case for the distinction between reference and access is clear. The main advantage of Web architecture is that there is now a *de facto* universal identification scheme for accessing networked resources. With the Semantic Web, we can now extend this scheme to the wide world outside the Web by use of reference. By keeping the distinction between reference and access clear, the lemons of ambiguity can be turned into lemonade. Reference is inherently ambiguous, and ambiguity is not an error of communication, but fundamental to the success of communication both on and off the Web.

Bibliography

Berners-Lee, T.(1994). IETF RFC (Informational) 1630 Universal Resource Identifier (URI). <http://www.ietf.org/rfc/rfc1630.txt>.

Berners-Lee, Fielding, R., and McCahill, M. (1994). IETF RFC (Obsolete) 1738 Uniform Resource Locators (URL). <http://www.ietf.org/rfc/rfc1738.txt>.

Berners-Lee, T., Fielding, R., and Masinter, L. (1998). IETF RFC (Obsolete) 2396 Uniform Resource Identifier (URI): Generic Syntax. <http://www.ietf.org/rfc/rfc2396.txt>.

Berners-Lee, T., Fielding, R., and Masinter, L. (2005). IETF RFC 3986 Uniform Resource Identifier (URI):

Generic Syntax. <http://www.ietf.org/rfc/rfc3986.txt>.

Conolly, D. (2000). IETF RFC (Informational) 2854 The 'text/html' Media Type. <http://www.ietf.org/rfc/rfc2854.txt>.

Connolly, D. (2006). A Pragmatic Theory of Reference for the Web. Proceedings of the Identity, Reference, and the Web (IRW2006) Workshop at the World Wide Web Conference (WWW2006). Edinburgh, United Kingdom. May 22nd 2006.

Davidson, D. (1973). Radical Interpretation. *Dialectica*, 27, 314-28.

Delugach, H. (2007). Common Logic (CL): a framework for a family of logic-based languages. ISO/IEC 24707. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39175.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and Berners-Lee, T. (1997) IETF RFC 2068 - Hypertext Transfer Protocol – HTTP/1.1. <http://www.ietf.org/rfc/rfc2068.txt>.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. (1999) IETF RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1. <http://www.ietf.org/rfc/rfc2616.txt>.

Frege, G. (1892). Uber sinn und bedeutung. *Zeitschrift fur Philosophie and philosophie. Kritik*, 100, 25-50.

Gangemi, A., Guarino, N., Masolo, C. and Oltramari, A., and Schneider, L. Sweetening Ontologies with DOLCE. (2002). *Proceedings of International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. Siguenza, Spain. 1-4 October 2002, pp. 166-181. London: Springer-Verlag.

Jacobs, I. and Walsh, N. (2004). Architecture of the World Wide Web. W3C Recommendation. <http://www.w3.org/TR/webarch/>.

Korzybski, A. (1931). A Non-Aristotelian System and Its Necessity for Rigour in Mathematics. *Proceedings of the American Mathematical Society*. New Orleans, Louisiana. December 28 1931 pp. 747-761. Providence, RI: American Mathematical Society.

Kripke, S. (1980). *Naming and Necessity*. Cambridge, Mass.: Harvard University Press.

Luntley, M (1999). *Contemporary Philosophy of Thought*, Oxford, UK: Blackwell.

Mealling, M. and Daniel, R. (1999) IETF RFC (Experimental) 2483 URI Resolution Services Necessary for URN Resolution. <http://www.ietf.org/rfc/rfc2483.txt>.

Moats, R. (1997). IETF RFC (Proposed) 2141 Uniform Resource Names. <http://www.ietf.org/rfc/rfc2141.txt>.

Russell, B. (1911). Knowledge by acquaintance, knowledge by description. *Proceedings of the Aristotelian*

Society, 11, 197-218.

Schwartz, A. (2004). IETF RFC 3870 application/rdf+xml Media Type Registration.
<http://www.ietf.org/rfc/rfc3870.txt>.

Quine, W. (1960). *Word and Object*. Cambridge, Mass.: MIT Press.