

Exploring HTTP Content Negotiation

Xiaoshu Wang

INESC-ID

R. Alves Redol 9, 1000-029 Lisboa, Portugal
(+351) 21 3100 228

xiao@kdbio.inesc-id.pt

Arlindo L Oliveira

INESC-ID/IST

R. Alves Redol 9, 1000-029 Lisboa, Portugal
(+351) 21 3100 228

aml@inesc-id.pt

ABSTRACT

In this paper, we explored a relatively underused feature of Hypertext Transportation Protocol (HTTP) – content negotiation. With a concrete use case, we illustrated how extending content negotiation with Uniform Resource Identifier (URI) can provide us with many new and exciting possibilities, such as managed URI transparency and contract-first methodology, to improve a data's life in the Web. In addition, we discussed the conceptual issues raised by content negotiation and showed how Fred Dretske's semantic information theory can help us formulate a concrete definition of "information resource" and a more meaningful architectural model of the Web, which, in turn, can help us clarify and probably settle many other issues, such as the new URI scheme, the fragment identifier, and the metadata issue, that have been heatedly debated in the Web. At last, we suggested a design pattern – the AND pattern – that will allow the Web to be incrementally developed and agilely accessed.

Categories and Subject Descriptors

C.2.2 [Computer Communication Networks]: Network Protocols; Theory – *general systems theory*. D.2.11 [Software Architectures]: *Patterns*. D.2.12 [Software Engineering]: Interoperability; H.1.1 [Models and Principles]: Systems and Information.

General Terms

Human Factors, Standardization, Knowledge, Information, Design Pattern.

Keywords

Semantic Web, HTTP, Content Type, MIME, Content Negotiation, Uniform Resource Identifier (URI), Information Theory

1. INTRODUCTION

In the beginning there was information (the Web). The word came later. The transition was achieved by the development of organisms (data, machine agents etc.) with the capacity for selectively exploring this information in order to survive and perpetuate their kind. ([4], p. vii)

We borrowed the opening paragraph of Fred Dretske's book *Knowledge and the Flow of Information* as our leading introduction for two reasons. First, Dretske's view on what information is has helped shaping up our view on what the Web is, which we will explain later. Second, by a few simple word substitutions (with ours in the parentheses), the paragraph outlines the purpose and importance to reach an understanding

of the Web. Obviously, the Web will run regardless of how we think about it. But to think about the Web – to underpin it with a philosophy – should help us enrich the Web by developing web organisms that will survive and perpetuate in it. To help with this transition – from the Web incarnated twenty years ago to the one that we can now understand and explore, let us study one of its living organisms.

2. A USE CASE AND A PROPOSITION

We have some biological pathway data that lives at *//yeastrack.com*¹. If the Web were an ecosystem, our data would be one of its primordial species because it is queried and returned in HTML. To better survive and perpetuate in the Web, our data needs to evolve so that it can be more easily integrated with the rest of the world.

To avoid complicating the matter, let us just consider the simplest, though not necessarily the easiest, use case: how should we serve our data about a gene named "xxx" to the rest of web organisms?

By the architecture of the World Wide Web (AWWW)[30], there are three design decisions that we need to make. The first one is the identification of our data. This can be easily solved (for now) with a URI pattern. For instance, we can designate the URI "*//yeastrack.com/gene/xxx*" to denote our data about a gene named "xxx". Some may argue that such a design is already biased toward the RESTful[6] web service. But let us leave the argument for now and we will come back to it in Section 3.2.

The second decision that we need to make is about our data's interaction with the rest of the Web. By serving our data via HTTP[5], the issue can also be easily solved because HTTP is the de facto standard transportation mechanism for the Web as it is ubiquitously supported in the Web.

The third issue – choosing a data format – is, however, problematic. Among the many XML-based markup languages (ML) for describing pathway data[23], none of them is capable of fully expressing the semantics of our data. This, however, does not come as a surprise to us and it argues for adopting an RDF-based data standard[33]. As a matter of fact, there is a pathway-centered ontology – BioPAX (<http://www.biopax.org/>), which, despite its few design problems[32], could at least be used by us as a starting point. But the issue here is not about whether we should embrace the future (of course we should) but whether we should face the reality. The current state of affairs is: there are just not many web organisms that can consume RDF, let alone those who favor biological pathway RDF. The predominant requests that we have received are whether we can return our data in XML.

¹ We have intentionally omitted the URI scheme to prepare for latter discussion.

To live in the present, our data must evolve progressively. Our data's form, therefore, should be in XML first and RDF later. Although existing MLs can only provide a partial representation of our data, they nevertheless gave it the best chance to continue its life so that it may prosper in the future.

But which ML should we choose? None of the existing MLs can fully express the semantics of our data; nor can it cover the semantics of the others. This suggests that we should not arbitrarily choose one ML over the others because, doing so would lower the vitality of our data as well as those who feed upon other MLs.² To maximize our data's survivability, we should – to the best of our ability – return our data in as many formats as possible. In this way, although not all of our data's semantics will be fed to those univorous organisms – they would not need it anyway because of their simple diet pattern, it is nevertheless possible to the omnivorous species.

But to return multiple ML meets a technical difficulty. All these MLs have the same internet media type (MIME or content type) – “application/xml”, making it impossible to serve various XML documents from a single URI under the current standard web practice. Our earlier designs, therefore, need to be revised. There are two options: we can either multiply our data's identifier or its interaction. Let us explore these two options in the next section.

2.1 Options and Solution

2.1.1 Identifier Multiplication

In this approach, we need to redesign our URI pattern. One possible design is to use “.../gene/xxx/m” to denote our data about “xxx” that will be returned in a particular ML denoted by *m*.³ This approach, however, is not a satisfying solution because it decouples a number of intimately related information by severing their contextual relationship. It is nearly impossible for a web organism to systematically discover the relation between “.../gene/xxx/m” and “.../gene/xxx/n”.⁴

One potential solution is to use the deprecated HTTP LINK header[20], which has been recently endorsed by W3C's Technical Architecture Group (TAG)[29]. But we do not think it is a sensible option for us. First, there is the concern of size. Consider, how should our data adapt to the situation when ML “*m*” has evolved from version 1 to version 2? As evolution succeeds through preserving variation but not by simple substitution, we need to again multiply our identifiers, such as by using “.../gene/xxx/m/*k*” to indicate the *k*-th version of ML *m*. What worries us here is not the size of our URI space, which is unlimited, but that of the HTTP LINK at “.../gene/xxx/m”. As the latter is not simply bound by the number of formats, i.e., *m*,

² XML transformation cannot solve the problem here because some semantics that could have been present in one ML will be missing when served in other MLs.

³ Any other designs, e.g., using URI query string, would be semantically equivalent to this design because they are syntactically transformable.

⁴ The issue here is not about whether we should explicitly assign a URI to an alternative representation of a generic resource[21] but about the linking between them. The same MIME type of alternative representations makes it impossible to front them with one generic URI because the traversal from the generic URI to one of its representative ones may not be possible.

but by the number of any web scenario that would lead to the multiplication of our identifiers, the LINK size will grow by a factor of $O((m \times k \times \dots)^2)$. Such a growth rate makes the LINK-header approach difficult to scale so that it cannot be a long term solution. Yet, even as a short-term solution, the LINK-header approach still has an issue of necessity. At any instance of time, any web organism will be univorous because it can only request one representation at a time. What this means to us is: when an organism makes a specific request for “.../gene/xxx/m”, most of the time it could not care less about the existence of “.../gene/xxx/n”. To compare the Web to our living environment, the LINK-header solution will not be “green” because it produces unnecessary web waste. Most of all, we think that the LINK-header solution would break “the principle of orthogonal specification” as described in AWWW[30] because the sole purpose of network transportation should be about delivering the right message but discovering related ones.

2.1.2 Interaction Multiplication

If multiplying identifier is not a viable option, it leaves us with the only option of multiplying the interaction. As it is unrealistic to expect all MLs to be registered with Internet Assigned Numbers Authority (IANA) as a special MIME type, we have to find ways to extend the HTTP's content negotiation. The current HTTP specification[5], in fact, provides us with an extension mechanism. As defined in section 14.1 of [5], the HTTP's Accept header can be extended with an “*accept-extension*” field by appending sets of name-value pairs after the “*accept-params*”. This leaves us with only one decision to make. That is, a naming convention.

Obviously, we should avoid arbitrary naming schemes because doing so would make our design both proprietary and implicit. Being proprietary in the Web implies that naming conflict is bound to happen; being implicit in the Web implies added difficulties for our data to be found, which we do not desire. The only sensible option, therefore, is to use URI, which can be shared by all web organisms. In our case, the namespace URI of an XML based ML would serve the purpose. For instance, if a client intends to obtain our data about gene “xxx” in PSI-MI[14] version 2.5, the client can simply send an HTTP GET to `//yeastrack.com/gene/xxx` with the following Accept header.

```
Accept: application/xml;q=1;d=
//psidev.sourceforge.net/mi/rel25/src/MIF25.xsd"
```

The token “*d*” is the one letter acronym for “*document-type*”⁵. We opt for the succinct form because we try to make our solution as “green” as possible. Unlike URI, which may, and perhaps quite often, be manually inputted into a software agent, such as a web browser, HTTP protocol will usually, if not exclusively, be handed by machines. As things processed by machine do not need human friendly name, using one character token can save us a few bytes on every HTTP request.

It is worth noting that the *document-type URI* does not have to denote the schema of a format because not all formats have a schema language as XML does. Nevertheless, dereferencing a *document-type URI* should obtain necessary information that will describe and possibly constrain the desired content negotiation.

There are several advantages to the above approach. First, it lowers the cost of document-type standardization. Instead of the compulsory MIME type registration[10, 11], deploying a

⁵ See our later discussion on the conceptualization of document.

document specification on the Web is all that is needed to standardize a class of document. Second, expressing content type in URIs allows content negotiation to be formally modeled. This may lead to new and more meaningful ways to negotiate the content of a resource. Third, as format definition is now dereferencible, it becomes possible for a web organism to automatically handle a novel content type – a problem that we are trying to tackle at [//dfdf.inesc-id.pt](http://dfdf.inesc-id.pt).

Of course, our design of extending content negotiation still has one proprietary convention – the use of token “d”. This is what we wish to be one day accepted as a standard web practice because, as we will show, what is proposed here, i.e., extending HTTP’s content negotiation with URI, will provide us with many new and exciting possibilities to improve our life in the Web.

3. NEW POSSIBILITIES

There are many new possibilities that we can think of. But to avoid prolonging this article, we will discuss two that may help us settle some heated debates in the Web. The first possibility is about making transparent URI, which will help us solve the new URI scheme issue, such as those functionalities requested in Extensible Resource Identifier (XRI)[34]. The second one is about the contract-first methodology, which may help us settle the SOAP vs. REST debate.

3.1 Transparent and Nice URI

On the surface, suggesting a transparent URI contradicts the best practice recommended by the AWWW (section 2.5 of [30]). Here we quote:

Good Practice: URI opacity – Agents making use of URIs SHOULD NOT attempt to infer properties of the referenced resource (from the composition of URI)⁶.

We think, however, what the above recommendation is against is the practice that an agent will derive conclusion in the absence of any information but not the practice that an agent can use to systematically and logically discover the composition of a URI. What we are proposing here is, in fact, a managed approach to URI’s transparency. Hence, we do not think that it contradicts W3C’s recommendation.

Our early design of URI still has an issue. The convention – “.../gene/xxx” – is still proprietary and implicit. To standardize it in the web, we should give “.../gene/xxx” a global scope. The question is how. One potential solution is to use Archival Resource Key (ARK) naming scheme[17]. But, besides a few of its conceptual issues, such as the ambiguous definition of metadata (we will provide our point of view later), the one aspect of ARK that discourages us is the potential verbosity of its syntax. For instance, if we were to provide our data related to a Gene Ontology (GO, <http://www.geneontology.org/>) term in ARK style, the URI would become as follows.

[//yeastrack.com/gene/ark:/http://purl.org/obo/owl/GO%23GO_012345](http://yeastrack.com/gene/ark:/http://purl.org/obo/owl/GO%23GO_012345)⁷

As seen, the globalization of just one component of a URI already makes its syntax verbose. It is not difficult to imagine

⁶ The original wording of AWWW is ambiguous so we have added our interpretation in the parenthesis.

⁷ The ‘#’ character of the hypothetical GO term must be escaped because, otherwise, the string “GO_0012345” will be treated as a fragment identifier and truncated by the Web server.

the situation when more than one of a URI’s components need to be globalized (e.g., see an XRI use case at [35]). Of course, if URI were to be exclusively handled by machines, any form would be acceptable. But the fact is that URI is used as often by humans as by machines.

There is a conflict between a human- and a machine-friendly URI. The former desires comprehensibility and usability; hence it is ideal that a URI is short. But the latter requires explicit global scope; hence it cannot possibly be short. This seemingly irreconcilable difference, however, can be solved by multiplying the interaction. What we need first is a language, which we have named as URI Description Language (URIDL) and is deployed at [//dfdf.inesc-id.pt/tr/uridl](http://dfdf.inesc-id.pt/tr/uridl). What URIDL does is to simply describe the potential scoping information of various URI components. With URIDL, we can redesign our URI pattern as follows:

[//yeastrack.com/gene/QName](http://yeastrack.com/gene/QName)⁸

The previous ARK style of URI would now be shortened as:

[//yeastrack.com/gene/GO_0012345](http://yeastrack.com/gene/GO_0012345)

Without any explicit knowledge, the semantics of the above URI is still opaque. But it can become transparent on request. For example, if a machine agent intends to understand our URI, it can send an HTTP GET to the above URI with the following Accept header.

application/xml;q=1;d=“//dfdf.inesc-id.pt/tr/uridl”

Upon receiving the request, the server at [//yeastrack.com](http://yeastrack.com) would return an URIDL document, where the namespace for the second path component of our URI, i.e., that of GO, can be described. On the other hand, for a machine agent working on behalf of a human, such as a Web browser, the Accept header can be set as follows:

application/xhtml+xml;q=1;d=“//dfdf.inesc-id.pt/tr/uridl”

Upon this request, our server would respond back with an HTML document, where the URI’s scoping information is described in human languages.

As seen, extending content negotiation with URI allows us to mint *transparent and nice URIs*. The *transparency* refers to the explicit global scope of every URI’s components; the *niceness* refers to the short form and comprehensibility.

3.2 SOAP and REST

When Simple Object Access Protocol (SOAP), along with its related protocol stack such as WSDL, WS-Addressing, WS-* etc., and REST are discussed in parallel, the two words are usually connected by the preposition “*versus*”. We purposely choose the conjunction “*and*” in the above heading to suggest that they are not two mutually exclusive technologies and can both be provided by an organization, even at a single URI.

The debate between SOAP and REST, we think, is often conducted on inappropriate footings. First, it is often argued from extreme viewpoints so that the presumed strength of one approach is often considered its weakness, when argued from the other side. Take the contract-first methodology (used in SOAP/WSDL) as an example. It is considered a good practice in terms of controlling the integration consistency over a distributed system but also a bad one in terms of promoting

⁸ QName stands for Qualified Name as defined in <http://www.w3.org/TR/REC-xml-names/#ns-qualnames>.

serendipitous reuse[25]. Neither view is factually wrong; nor are they factually correct; because the facts they use are partial and the effects they seek are different. To this end, we can find relevant evidence from political science, where the central-local relation tackles exactly the same problem. We know – from the facts provided by our human history – that neither complete anarchy nor total authoritarianism is the ideal political system. The key is almost always about the balance, as a matter of degree and scope, between the two extremes.

Second, the debate between SOAP and REST is often based on false premises. For instance, it is often assumed that REST is all about *ad hoc* integration. But this is definitely not true because we can furnish REST with a contract-first methodology as well. First, it is not difficult to create a service description language for REST. For the sake of argument, let us call it RSDL (for REST Service Description Language). Second, we can deploy REST Service Definition Resources (RSDR) on the Web, where RSDL will be used to define a particular service interface. If desired, a RSDR can also serve as a registry for its implementations. Third, at every REST service endpoints, we could open up an RSDL channel, i.e., by content negotiation, over which an implementation of a RSDR will be declared along with its mapping of service parameters to its URI components. If special naming convention is required, an agent can also obtain the necessary information from the resource via negotiating the URIDL content.⁹ With RSDL, the semantics of “.../gene/...” in our URI design can also become transparent, making its integration with the rest of the Web easier.

It is not difficult to see that REST/RSDL can accomplish exactly the same task achieved with SOAP/WSDL. But the difference – perhaps a very important one – is that REST/RSDL allows both *contract-first* and *contract-later* development to take place because it does not require a uniformed URI design. This will in turn help maintain a URI’s stability, which is an important factor to make a living in the Web[26].

In principle, we believe, all technical features of WS-* stack can be implemented in REST, with the obvious exception of those that do not channel over HTTP. Such a statement may make us sound that we are biased toward REST. This is true. But to think it in that way may have missed our point: the debate between REST and SOAP is wrongly framed. REST is a set of architecture design principles that happened to be realized via HTTP in the Web[6]. The proven success of the Web may suggest that the design of any large scale distributed communication systems will more or less follow the same set of principles. In other words, the WS*- stack may eventually become RESTful too. In this sense, the debate should have never been framed as “REST vs. SOAP” because it is comparing apples with oranges. Using the title – “HTTP vs. SOAP” – should give us a more appropriate mindset to conduct the debate. Along this line, we cannot possibly be biased toward either choice because we do not know the HTTP dependence of the organization at debate.

4. CONCEPTUAL ISSUES

In the previous sections, we have shown how we can accomplish many tasks by multiplying a resource’s interaction via HTTP’s

⁹ Naturally, a server may be simply configured to handle a full URI as the parameter. But this is not required. In addition, we only illustrated the HTTP GET scenario. The same principle still applies to POST.

content negotiation. In principle, we believe that extending HTTP content negotiation can satisfy most, if not all, communication needs in the Web. To support our claim, let us quote Butler Lampson’s famous aphorism – *all problems in computer science can be solved by another level of indirection*. Interaction multiplication is simply another form of indirection. Any communication problem between two resources is abstracted into another resource, denoted by a URI, and solved by negotiating over the URI.

Evidently, how we have used HTTP’s Accept header may have been different from what the mechanism was designed for. But, what we have done here is exactly what Dretske has suggested – to explore an information for the purpose of advancing our data’s life in the Web. Besides, our proposed solution neither violates nor requires any change to existing web protocols. In addition, it is in line with the REST model that was originally envisioned.¹⁰

Nevertheless, conforming to technical specifications and architecture styles does not mean that our exploration will not lead to any conceptual debates. In the subsequent sections, we will make an attempt to describe how we have understood the Web.

4.1 URI, Resource and Representation

The architecture of the Web is defined in terms of three essential concepts: *URI*, *resource*, and *representation*. A URI is a symbolic *thing* – a kind of *thing* that *denotes* or *references*¹¹ another *thing*, which is called *resource* in the Web. Obviously, *things* have meaning, which is the relation between one *thing* and another. To obtain meaning, however, requires a communication system because, in the absence of such a system, everything becomes unobservable and irrelevant to others. In the physical world, the system is space-time, where meaning is delivered in force and energy; in biological world, it is sense, where meaning is delivered in light, sound, and pressure etc; in the Web, the system is the network transportation, where meaning is delivered in *representations*.

The above description of *URI*, *resource* and *representation* should be consistent with how they are described in the current AWWW[30]. But to make subsequent discussions clear, we will narrow the definition of *resource*. *Resource* is here used to refer those *things* that are neither *symbolic* (so that Resource ≠ URI) nor *representative* (hence, Resource ≠ Representation). In addition, a *resource* must have an established identity, i.e., an explicit and canonical URI, in the Web. The word *thing* will be used to replace the general notion of resource.

It is worth noting that making the above distinction is not to suggest that neither *URI* nor *representation* can be modeled as *resources*. They certainly can because it is simply a matter of abstraction – with a URI. But within any given context, a *thing*

¹⁰ See Roy Fielding’s response[7] to Pay Hayes’ articulation for our model[12] on TAG’s mailing list.

¹¹ In AWWW[30], the word “identify” is used to describe the relation between *URI* and *resource*. But the word “identify” has at least two interpretations: (1) to cause to be or become identical, and (2) to establish the identity. We believe the intension of AWWW is the second one because a URI cannot possibly become identical to the resource that uses it to establish the resource’s identity in the Web.

(resource) cannot possibly be the same as another thing that references it. Nor can it be the same as what represents it.

4.2 Information Resource and Document

In the current AWWW, there is a fourth concept – *information resource*, which is also informally known as *document*[1]. We will use the word “information resource” in this section and “document” in subsequent sections, which reason will be obvious as the discussion progresses.

Information resource is defined to be those things “that all of their essential characteristics can be conveyed in a message[30].” However, not only the clarity of the above definition but also the essentiality of such a concept to the Web has been controversial. “What information resource is” is at the heart of many debates, most notably the httpRange-14[27]. In [31], we have argued how the above ambiguous definition cannot possibly be objectively followed in practice. In here, we will argue its philosophical shortcomings. To this end, Dretske’s writing on what information is could be instrumental.

There is one way of thinking about information. It rests on a confusion, the confusion of information with meaning. Once this distinction is clearly understood, one is free to think about information (though not meaning) as an objective commodity, something whose generation, transmission, and reception do not require or in any way presuppose interpretive processes. One is therefore given a framework for understanding how meaning can evolve, how genuine cognitive systems – those with the resources for interpretive signals, holding beliefs, and acquiring knowledge – can develop out of the lower-order, purely physical, information-processing mechanisms.([4], p. vii)

According to the above view, just the wording of “information resource” could already start out as a confusion. Resource (*thing*) is an inherently static concept and by nature *meaningful* because a completely meaningless thing lives in absolute solitude and, therefore, virtually non-exist. Information, on the other hand, is an inherently dynamic concept because it is often associated with an event, from which knowledge may transpire. To say *information*, therefore, presupposes two *things* – a source and a recipient, but to say *resource (thing)* presupposes only one *thing*. Hence, the simple apposition of the two words already foretells an identity crisis.

Nevertheless, it is not too wrong to suggest that there are two kinds of things in the Web – one is *informative* and the other not. As a matter of fact, the Web was started out with such a distinction. The informative things were traditionally denoted by URLs and the others by URNs.[3] However, as soon as the Web started its march to the Semantic Web, it was realized that the distinction between URL and URN is only arbitrary and inconsequential. Once again, it is due to the confusion of meaning with information. But this time, it is a variant of it – the confusion of reference with access.[13] Reference is the subject of semantics whereas access is the means of obtaining information. Unless URNs are used to denote things that cannot possibly be connected to the Web, URN-things can always be accessible in the web and, therefore, informative. In fact, these absolute URN-things do not even exist, at least within the confines of human knowledge.

If the AWWW’s definition of “information resource” simply stops at “can be conveyed in a message”, there will be no debate and no controversy. Anything else would be nothing more than a play of words and a parade of self-conceit. However, in telling

us what *an information resource* is with the ambiguous wordings, such as “all”, “essential” and “can”, and in telling us how a non-information resource should behave[27], and in telling us that non-cooperative behaviors may be potentially punished in the future[28], the AWWW’s definition of *information resource* makes the Web impossible to work with. The reason is clear: unless there is a complete and indisputable set of knowledge on everything in the universe, what is “all”, “essential”, and “can” is always subject to debate and change.

4.3 Two Models of the Web

On a deeper level, the debate about the essentiality of *information resource* reflects two contrasting views on what the Web is. In the first view, the Web is considered to be a web of *documents* talking about *things* (the Shadow Web model coined by McCool in [18]) In the second, it is a web of *things* talking with *documents* (Figure 1). Let us see how the two views will affect our thinking about the Web and our life in it.

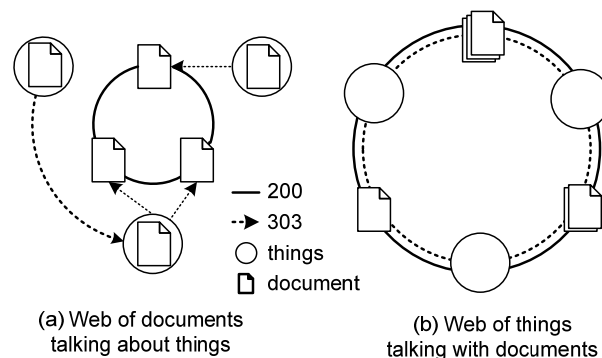


Figure 1 Two architectural views of the Web

4.3.1 A web of documents talking about things

The rearing of this view may come from history. As the Web was initially conceived to be a web of documents – those written in HTML and linked through HTTP, it seems right to suggest: as a natural progression, a web of things should be built on top of a web of documents. However, our general use of the word “document” has been ambiguous because, subconsciously, we often think that a “document” retrieved from a URI is the “document” denoted by that URI. But this is an obvious case of psychological identification but not a physical one because a document cannot possibly be the same as another one that represents it.

Nevertheless, putting aside the issue of network’s reliability, practicing psychological identification in the Web may not matter much if there is a one-to-one relationship between resource and representation. In fact, this one-to-one relationship is the prerequisite for *the web-of-document* model to stand. Of course, we do not know if this – the first architecture view of the Web – is what is behind the definition of information resource and the resolution of httpRange-14. But the latter two certainly have helped reinforcing the former. Take the definition of *document* as example. If they are what they are defined for, i.e., that *all of its essential characteristics can be conveyed in a message*, we can of course HTTP-GET a document – *all in one representation*. On the other hand, if *non-documents* cannot have it *all*, they must not have *any* (representation) because otherwise *documents* would have no privilege over *non-documents* in the Web. This not only makes any definition of document (as resource) a moot point but also makes *the-web-of-documents* virtually non-exist. However, by mandating non-

document things to 303-redirect (per the bylaw of httpRange-14), a web of 200-documents (but not exactly a web of all documents) is created, upon which a web of other things could be built, though rather inconveniently.

Content negotiation, however, strikes a serious blow to the above model because what if a *resource/document* can but may not convey the *all* in *one* of the possibly many representations? This will once again blur the distinction between document and non-document things and subsequently jeopardize *the-web-of-documents* model. Of course, to make the model work, the *one-representation* can be thought as a semantic- rather than a syntactic-*one* so that all representations of a document become mathematically transformable. To make this work, however, requires all languages – both human and machine ones – to have the same expressivity on every possible subject. This is neither true in reality nor likely to be true in the future. For instance, how can we, if ever, to capture the essence of a picture in a language? The only option to make the one-to-one model work will be through control. That is, to make a policy on what kinds of contents that a resource can and cannot be negotiated. If this were indeed to happen, we must seriously ask ourselves: is this what we want, or what we want the Web to be? First, it will bring unnecessary hardships to the life of web organisms. Think, for example, how should our data at *//yeastrack.com* live and behave without a freely extending content negotiation? Second, it hinders innovation. Most suggested possibilities in this article may not be possible. Third, it makes the Web cumbersome because a resource must always respond in an all-out fashion as opposed to do it discriminatingly according to a client's special request.

As always, of course, we should sacrifice ourselves for the benefit of community. But the question – and a very important one – is: what could the Web have possibly gained had we made all the above sacrifices? From what we know and can possibly imagine, nothing – except a vaguely defined vocabulary and an arbitrary model.

4.3.2 *A web of things talking with documents*

What makes the first model fail, in fact, is neither its grip on *document*, which is an essence of the Web, nor its use of the word *information*, which is a correct instinct. The model fails because it misaligns *information* with *resource*. Perhaps, we have all forgotten that the Web is at the first place a communication system. As it is with any communication system, it is the nature of signal, but not that of its producers or receivers, that defines the system's characteristics. In this sense, the correct alignment of *information* should be with *representation* but not with *resource*. To follow Dretske's advice[4] that, in order to provide a semantic theory of information, the word *information* should be used to refer to the semantic or propositional content of a signal¹², we think that the word *document* should be more meaningfully defined to be the semantic content of a *representation*. By this definition, *document* is neither *representation*, which is the signal that carries it, nor *resource*, which is the *thing* that either produces or digests it.

In addition, *document* is not *meaning*. A *document* is a (but not *the*) piece of *information* about its producing *resource*. What a *resource* is, i.e., its *meaning*, is an entirely different matter and,

in fact, completely irrelevant to what a *document* is because the meaning of a *resource* depends on the cognition of a document's recipient and that of its producer. In a broader sense, any *resource* is *meaning a priori* and any *document* produced belief in another *thing* is *meaning a posteriori*. The purpose of the web, therefore, is to promote the sharing of these two kinds of meanings by making them easily accessible through the flow of information (document).

The above notion of *document* should fit more naturally to our common sense of "document". What we see and recognize from a browser, for instance, is never a *representation* but a *document* because the former in the Web is by nature a byte stream that we neither perceive nor cognize. Of course, this new definition of document still makes it rather inconvenient to talk about it because identifying a document requires two URIs – one of its source and the other of its *document-type*¹³. Nevertheless, the inconvenience of talking about information is neither a new nor a web-specific problem because, otherwise, there would have not been so many theories on information (see review in [9, 19]).

Evidently, we do not choose Dretske's theory for convenient reason. Nor do we in fact choose it for being the only correct information theory, about which we do not know much and could not care less. "A good problem", as David Hilbert pointed out in [15], "is a problem rich in consequence, clearly defined, easy to understand, and difficult to solve, but still accessible" (paraphrased by Luciano Floridi in [9]). In this light, Dretske's framework sure seems the right one for the Web because it gives us *the-web-of-things* model which, not only makes the web organisms easily integrated with each other, but also makes the Web easily integrated with the rest of world.

Let us use a simple example to illustrate the last point. Imagine the apple that we have placed in front of us (*u:s*) and named it *an:apple* (let "*an:apple*" and "*u:s*" be URIs). There are several information systems that connect *an:apple* with *u:s*. There is the light that gives *u:s an:apple*'s color and shape, and *vice versa*; there is the air that gives *u:s an:apple*'s scent, and *vice versa*; and there is the Web that gives *u:s an:apple*'s birth place, drug (pesticide) history, etc. and *vice versa*!

Some may immediately cry afoul over our use of "vice versa": *an:apple* can NOT see, smell, let alone access the internet. But, let us refute them with "子非鱼, 焉知鱼之乐?" Our intension here, though, is neither to show off our Chinese (being the first author's native language) nor to direct this article to more philosophical issues than it is necessary. Rather, we are here to illustrate Dretske's point: information is objective but meaning is not. What *an:apple* is as a reality – that is, what makes *an:apple an:apple* but others not – is an ever eluding thing to know. But we can think – as a personal or public-accepted belief – that *an:apple* can NOT see, smell, and access the internet. But we do not know that for a fact because we do not know apple's language. This is exactly the same situation that we – you, the readers, and *u:s*, the authors – are in with regard to a third thing "子非鱼, 焉知鱼之乐?" As an *information* from *u:s*, the sentence carries some meaning of *u:s* but incurs (perhaps) nothing in you. Yet, what the sentence is as itself is unknown because we do not know what the sentence, the *information*, means to the sentence, the *thing*. Hence,

¹² In Dretske's account, Shannon's communication theory[22] is a quantitative theory of signals but that of information.

¹³ The meaning of our proposed *d*-value should be much clearer now: it should define a class of documents.

information, as an objective entity, is always out there in spite of how it is thought. Conversely, how one *thing* thinks of another is independent of how the latter's information is acquired. For instance, even without our translation¹⁴ of “子非鱼，焉知鱼之乐?” you may still find it in some other way. Our information paths to the sentence are definitely different, but our conceptualization about it – that is, its meaning in us – may nevertheless be ended up the same.

With the above illumination on the subject, let us now ponder the question: is there any essential difference between the light, the air, and the Web as information systems? And, is there any essential difference between wavelength, scent molecule, and document as informations? We think not. The Web, in the view of the second architectural model, is simply part of the natural world – as natural as it gets.

Any information system is in fact built from a *web of thing* model; what makes them different is the information that flows in the system. We can, for instance, build a *web of things talking with apples* quite easily. All that is required is our knowledge about one – but not all and not necessarily one essential – aspect of apple, such as our simple ability to tell sweet apples from sour ones. In fact, the Web is built upon another web of similar attribute. Only this time, it is not sweet-sour apples but on-and-off bits. The bit-web is, in turn, built upon an electron-web and a photon-web, and so on. It is through the bridging of all these information-webs that things become more accessible and meanings, and more meanings, evolve.

4.4 Information Triads

4.4.1 Knowledge-Information-Data (KID)

In the last section, we have mentioned that any resource can be considered to be *meaning a priori* and document produced belief *meaning a posteriori*. Now, let us use the word “data” to refer to the former and “knowledge” to the latter. This should give us the familiar knowledge-information-data (KID) triad.

Knowledge and data are, however, relative concepts; they are interchangeable. Whether something is a piece of knowledge or data depends on its position in an information processing line. Both knowledge and data are subjective entities, in the sense that they are purely about meanings. As meanings are expressed in languages, knowledge and data do not have a physical structure. Information, on the other hand, is objective and must have a physical structure. It follows that a structure-less thing cannot be information.

It is worth noting that the above use of the word “data” does not conflict with our conventional view of data, which usually carries some physical structure. What a data is, in fact, is about its content but not about its form; the former is its meaning while the latter its information. Once a data is fed into a processing unit, such as a computer program, and its form (structure) being analyzed, it becomes information and may output knowledge, which can in turn serve as the data for the next information processing unit. In other words, it is the data – as *information* – that is being analyzed but it is the data – as *meaning* – that drives an algorithm.

¹⁴ “You are not fish, how can you know the happiness of fish?” It is a famous quibble recorded in the book of Zhuangzi (see more at <http://en.wikipedia.org/wiki/Zhuangzi>).

4.4.2 Symbol-Information-Referent (SIR)

Meaning is the relation of symbols. Of the most fundamental kind of meaning is the equivalence. When we named the apple in section 4.3.2 with the symbol *an:apple*, we have, in fact, created a meaning – *an:apple is the-apple*. There are three symbols in this assertion; they are from three different symbol spaces used by three different information-systems. “*An:apple*” is a symbol in the document-web; “is” is a symbol in the English-system; “the-apple” is a symbol (e.g., as a geodesics) in the space-time system. Obviously, all three symbols must be subsequently projected to the symbols in our mental space in order for us to comprehend its meaning. But without the “is”, there will be no meaning. Both *an:apple* and *the-apple* would be just symbols. But with it, *an:apple* becomes a reference with *the-apple* being the referent. To assert the “is”, however, requires information. For us, the authors, the information could be a simple body gesture; for you, the readers, the information could be this document.

With the above understanding about symbol and references, we can formulate a more fundamental information triad – Symbol-Information-Referent (SIR) – to define an information system. The referential realm of the symbol space necessarily defines the system's boundary and expressivity; and the kind of information defines the system's characteristics.

Anything is by nature a SIR system, with its self-identity serving as the *symbol*, its content the *referent*, and its form the *information*. But a closed self-reference SIR system is useless to others because its meanings cannot be passed across. The system “3”, for instance, will be meaningless to us unless its content is projected to a symbol, e.g., as a number, in our mental system. Hence, in order for meanings to evolve, symbols of various SIR systems must be either bound (by asserting the equivalence) or shared. Only by symbol-binding and sharing can two SIR systems be combined into a larger SIR system, from which more information can be acquired and more meanings can evolve. In our earlier example, for instance, the light, the air and the Web forms a larger information system with our binding of *an:apple to the-apple*, which tells us a more complete story about *the-apple*. Of course, to verify the binding requires information, such as a wavelength, a scent molecule, a body gestures or a document. But whether one accepts or rejects the binding of symbols depends on whether the information incurred meaning is consistent with one's personal knowledge. To put it plainly, without information, nothing gives. But with it, nothing is a given.

4.4.3 The SIR triads for the Web

Naturally, as a man-made SIR system, the Web must define its own information and implement transportation mechanism to deliver it. But being man-made does not and should not suggest its model be any different from the naturally occurring information system. In Table 1, we have defined a few web systems in terms of the SIR triad.

Table 1: The SIR triads for the Web

System	Symbol	Information	Referent
Semantic Web	URI	Document	Resource
Physical Web	URL	Representation	URL Endpoint
DNS	Domain	A Record	IP Address
The internet	IP Address	Packet	Machine

There are few things worth mentioning about Table 1. First, the distinction between the physical Web and the Semantic Web is

obviously arbitrary. But to make such distinction emphasizes where the Semantic Web should be focused on. In addition, it decouples the Semantic Web from its implementation so that a Semantic Web application can (in principle) be executed on any document-delivery system that will take the URI's symbol space, be it HTTP-Web, SOAP-Web, or Postal Office System. Second, the SIR-triad is the model for capturing the essentiality of an information system, where REST is a model for capturing the essence of good implementation. The two models obviously complement each other. But they are nevertheless different because they address different concerns. Third, we stopped our writing of SIRs at the machine level. Nevertheless, based on the principle introduced above, it is not difficult to write it out all the way to "the lower-order, purely physical, information-processing mechanisms" as Dretske have envisioned. This last point, to echo our earlier quoting of Hilbert in section 4.3.2, is the consequence (a very rich one we believe) of adopting Dretske's framework. This is also the reason why we have accepted his take on information because the Web is now "clearly defined, easy to understand, and difficult to solve, but still accessible."

5. OTHER ISSUES

As Guy Fitzgerald pointed out in [8] (p. ix), one of the most important contributions that philosophy can make to the world of information system is "to highlight the various assumptions that underlie our action." The above conceptual detour was aimed at just that because, once we understand the distinction between *resource* and *information* and their wrongly assumed one-to-one correspondence, many other seemingly difficult questions can be easily answered with clarity.

5.1 URI Scheme Issue

There are some efforts, such as Life Science Identifier (LSID, which seems now defunct) and XRI[34], that attempted to create new URI scheme, arguing that the de facto standard http-URI scheme are not suitable for their need. We believe that TAG's decision to discourage these efforts[24] has been correct. Creating a new URI scheme is the equivalent of creating a new SIR system, hence risking the danger to fragment the Web.

Most the so-called identifier issues, we think, have been wrongly argued in the past. Identifier has only one design issue. That is, if it is sufficiently structured to cover the desired referential realm. In this regard, any URI, http- or not, is sufficient because the URI's host component binds its system to the internet, which is a necessity, and the rest can be bound to any other things in the universe. All the so-called identifier issues are, in fact, maintenance and trust issues of a particular SIR-system's implementation but not issues of identifier design.

Take the often argued persistent identifier issue as an example. "It is purely a matter of *service* and is neither inherent in an object nor conferred on it by a particular naming syntax." [17] A persistent identifier should denote a persistent symbol-binding between two SIR systems. Since there are quite many SIR systems involved in the Web, there are many kinds of persistent bindings. For instance, if *an:apple* were designated to be a persistent identifier, which one of the following bindings would it denote? (1) The binding between *an:apple*'s referent, i.e., *the-apple*, and others things that *the-apple* relates. In other words, would *an:apple* denote a never rotten apple? (Maintenance of the natural world) (2) The binding between *an:apple* and *the-apple*? (Maintenance and trust of human SIR system) (3) The binding between *an:apple* and a URL endpoint. (Maintenance and trust of human and DNS system) (4) The binding between

an:apple and its document. But this confuses resource with information and we know that the latter requires two, but not one, URIs to be identified. Unfortunately, most persistent identifier's efforts have failed to clearly define what kinds of persistency their identifiers intend to denote. As there is more than one kind of persistency that needs to be addressed, one identifier syntax or scheme simply cannot be a complete solution. It is, therefore, only sensible that we leave the semantics of URI's composition opaque by design but transparent by request (see section 3.1).

Nevertheless, the continual effort and support¹⁵ for creating new URI scheme suggests that there is a deep conceptual gap between the understanding of TAG and that of general public. The reason is obviously due to the intimate relationship between http-URI and HTTP. To bridge the gap, we are here making another proposition. Let us create a *scheme-less URI* and let it be the http-URI sans "*http*". Since the scheme-less URI is essentially a URN, which we only need one set in the Semantic Web, we can convert the hard question "Do we need new URI scheme?" into a very easy one "Do we need new URL scheme?"

There are several advantages to the scheme-less URI. First, it is backward compatible with our existing practice of using http-URI. This is the reason that we have used "*//yeastrack.com*" in our earlier use case. We believe that most readers, if they will, can find its information without much difficulty. Second, the scheme-less URI makes the URI scheme more meaningful as it will now specify an information path to a resource. This will make the principle *reference does not imply dereference* (section 3.5 of [30]) much more evident and easy to follow. Third, it solves existing problems. For instance, the identity of an http-resource has often been compounded by the use of security protocol, i.e., https. With the scheme-less URI, the issue naturally dissolves. Forth, the scheme-less URI allows the Semantic Web to be further cleanly separated from the physical Web because they not only use different *information* but also different symbol spaces. Last, the scheme-less URI may settle the URI scheme issue once for all. Because most, if not all, desired identifier's functionalities (e.g., those raised in XRI[34, 35]) can be solved through managed URI's transparence (section 3.1), there will be no need to suggest any new URI scheme unless it is coupled with a new transportation protocol. Obviously, content negotiation is currently only supported by HTTP. But we believe that not many systems will mind, or can even avoid, using HTTP as a bootstrap protocol in the Web.

The next obvious question is: how should the Semantic Web (i.e., the scheme-less URI) be bound to the physical Web (i.e., a *schemed URI*)? To check out reality, we typed *//yeastrack.com* into three browsers: IE 7.0, Firefox 3.0.3 and Chrome 0.2. The former two use the scheme "http" and the latter "file". This result seems to suggest that the scheme-less URI will cause access problem without an explicit binding. But our thinking here is however different. First, imagine what if we have named our file system, or in fact any other aspects of our life, according to the scheme-less URI: our whole world would be much more organized and meaningfully connected. Second, all three browsers failed¹⁶ to meet our expectation is not due to the lack of information but their disregard of it. Had they made their

¹⁵ XRI 2.0 failed narrowly to become an OASIS standard. See <http://lists.oasis-open.org/archives/xri/200806/msg00001.html>.

¹⁶ The table will turn the other way around if we typed *//localhost/c:/work* into the browsers.

guess based on URI's host component, they would have succeeded. Hence, the problem, if there is one, is not about the scheme-less URI but about us – the human as a SIR system. Although we all desire convenience in life, we cannot be too lazy in acquiring it. To use the Web, we should have some basic knowledge about it. Knowing the most popular mean to acquire information from an internet domain is via HTTP and that from a localhost is via “file” should be as elementary as knowing the alphabets to use a language. Our point is: the binding of scheme-less URI to schemed-ones should not be defined but be driven by the popular demands of web organisms. Obviously, it is in the best interest of a resource owner to ground his or her resources to the most popular physical Web. But, on the other hand, we should not hinder innovation by discouraging them from inventing new ways to deliver information.

5.2 Fragment Identifier Issue

The current URI specification[2] (section 3.5) states that “the fragment identifier component of a URI allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information.” As multiple documents may be obtained from a single resource via content negotiation, it seems to suggest that there is a flaw in the URI specification.

But once we know that (1) a URI, fragmented or not, denotes meaning (resource) and (2) a resource's meaning is independent of the path to its information, which could be either a complete or a part of a document, we should know that the fragment identifier issue is not about the URI's specification but about our practice about it.

The current AWWW (section 3.2.2) recommends that “representation providers must not use content negotiation to serve representation formats that have inconsistent fragment identifier semantics.” But there are two ways that we can follow the advice. One, we can avoid potential inconsistency by not mixing fragment identifiers at all in variant documents. Two, we can embrace it by making the meanings of all shared fragment identifiers consistent across variant documents. It is the second practice that we recommend because it avoids unnecessary URI duplication that hinders the integration of various *document-webs*. For instance, should an HTML document be provided at RDFS's namespace (at a matter of fact, we think it should), the content of HTML element with an “id” attribute of “Resource” should describe the definition of *rdfs:Resource* in some natural languages. In this way, a person who is illiterate to RDF can still find out, e.g., via a browser, about what *rdfs:Resource* is. And this knowledge she acquires by herself will be consistent with what an RDF-agent will tell her later.

5.3 The Metadata Issue

There are quite a few efforts, such as LSID, HTTP LINK[20], and ARK[17], that intend to establish a standard mean to obtain the metadata (or its variant word “description”) of a resource. Obviously, their intension cannot be faulted. But their approach nevertheless should be. In our common sense, metadata is the data about data. Hence, to know what a metadata is presupposes some knowledge about what the data is. But within the Web, we do not know the latter. For instance, what would be the metadata of *a:thing*? An often heard answer is: whatever *a:thing*'s owner thinks it is. But this is trivially true because no one is expected to acquire information that she already knows. As an information recipient, what she wants to know is: in which form will the metadata be delivered? A metadata in SQL, for instance, is useless to a SQL-illiterate human or an RDF

agent. Thus, the word “metadata” can only be meaningfully defined with regard to language and formats. But this is what content negotiation does and it makes any metadata-effort a redundant one.

It is the same (redundancy) reason that we do not think the rationale behind the LINK-header approach[20] is logical. In the Web, there is no such thing as legacy resources but only resources that refuse to evolve. But a refusal to evolve implies neither LINK-header nor content negotiation is needed. On the other hand, if a resource does evolve, it should evolve via content negotiation but not via an HTTP header because otherwise the *principle of orthogonal specification* would be violated by putting data semantics into the transportation layer.

We believe what makes people to take the faulty notion of “legacy resource” and “uniformed access to metadata” is, once again, rooted from the-web-of-document model, which presupposes a one-to-one relation between resource and representation.

There are, however, two possible exceptions, where metadata can be a sensible definition in the web. They are: (1) the data about the composition of a URI and (2) the data about a resource's available document types. The word metadata makes sense here because the above two pieces of knowledge are what we always know about the Web as a SIR system. In this light, the short-hand notation “?” and “??” used in ARK[17] may be introduced to URI specification so that “*a:thing*”, “*a:thing?*”, and “*a:thing??*” would respectively denote *a resource*, *a URI* and *a resource's document-types*. This would, in fact, make URI's referential realm self-complete. And such a syntactic completeness would allow us to model the semantics of both a URI and its referent without raising any ambiguity issues. In addition, it allows us to refactor the content of *document-types* (i.e., those defined in the transparent content negotiation[16]) into the more appropriate place – the Semantic Web – from the physical Web that it is currently in.

6. THE “AND” PATTERN

The Web, as it started out its life journey twenty years ago, was already a Semantic Web. Only at the time its language for expressing semantics was the informal (in terms of logic) natural languages. This is what the current Semantic Web initiative is aimed at complementing with RDF and OWL. Nevertheless, we must not take RDF as a one-stop solution to every problem. First, RDF simply cannot be the one-stop solution because general logical formalisms are undecidable. Human intelligence must be infused at one point or another into machine agents that are tailored for some specific tasks. Second, even for the semantics that RDF is capable of expressing, the RDF's formalism may not be desired. Expressing a large data set in RDF (or XML in that matter), for example, is unlikely to be a sensible approach. RDF/OWL, therefore, may be the ultimate mean in terms of automated information processing; it is nevertheless not the ultimate goal of the Semantic Web, which should be about *sharing* data and knowledge (as resources).

How much a resource is shared, therefore, should reflect the resource's vital index in the Web because an unshared resource has essentially no web-life in it. To improve a resource's sharing is the equivalence of improving its life. The key, and perhaps the only key, is to maintain a URI's stability. Not only should we not abandon a URI that we already have, but should we also not unnecessarily duplicate URIs to denote our resources. To achieve this goal, we must think about the following three questions. (1) How will our resource handle

change? (2) How will our resource be used by others? (3) How will our resource be serendipitously discovered? In [32], we have discussed how the first two issues may affect an ontology's design and deployment in the Web. In the use case given in this article, we discussed how a general data source should evolve under the same URI. To meet the third criteria, we should think about what kind of general search engines that are, and will be, available. At the present, there are quite a few and, in fact very successful, search engines based on human languages. Hence, we should always serve our data in HTML. In the future, we can envision a few RDF based search engines. Thus, we should also serve our data in RDF, if possible. The HTML+RDF pattern should allow general search engines to usher more specialized web organisms into our data world so that our data can be selectively accessed in more efficient and meaningful manner. To satisfy those specialized agents, we should also serve our data in XML and in SQL and in Binary and in SOAP/WSDL and in RSDL, etc., – one piece at a time as long as there is demand. Most of all, all of them should be served under the same (scheme-less) URI, whenever it is applicable. This is “The AND pattern” that we recommend because it allows the Web to be incrementally built and agilely accessed.

In light of “the AND pattern”, if the web has a shadow[18], it would be a shadow over information but not over resource (meaning). The meaning of a thing should always be made explicit and transparent regardless who or what is trying to understand it because: what else do we need in life other than the pursuit of its meaning?

7. ACKNOWLEDGMENTS

This work was developed while the first author was supported by a research contract financed by the Ciência 2007 program of Fundação para a Ciência e Tecnologia.

8. REFERENCES

- [1] Berners-Lee, T. (2006) *What do HTTP URIs Identify?* <<http://www.w3.org/DesignIssues/HTTP-URI2>>
- [2] Berners-Lee, T., et al. *Uniform Resource Identifier (URI): Generic Syntax*. IETF RFC 3986 (2005)
- [3] Berners-Lee, T., et al. *Uniform Resource Identifiers (URI): Generic Syntax*. IETF RFC 2396 (Obsolete) (1998)
- [4] Dretske, F.I. *Knowledge and the Flow of Information*. MIT Press, Cambridge, MA (1981)
- [5] Fielding, R., et al. *Hypertext Transfer Protocol -- HTTP/1.1*. IETF RFC 2616 (1999)
- [6] Fielding, R.T. *Architectural Styles and the Design of Network-based Software Architectures Information and Computer Science*, University of California, Irvine, Irvine, California, (2000)
- [7] Fielding, R.T. (2008) *TAG Mailing List*, <<http://lists.w3.org/Archives/Public/www-tag/2008Apr/0223.html>>
- [8] Fitzgerald, G. *Forward*. in Russel Winder, S.K.P., Ian A. Beeson ed. *Philosophical Aspects of Information Systems*, Taylor & Francis, (1997) 258.
- [9] Floridi, L. *Open Problems in the Philosophy of Information. Metaphilosophy*, 35 (4) 554-582 (2004)
- [10] Freed, N. and Klensin, J. *Media Type Specifications and Registration Procedures*. IETF RFC 4288, (2005)
- [11] Freed, N. and Klensin, J. *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*. IETF RFC 4289, (2005)
- [12] Hayes, P. (2008) *TAG Mailing List*, <<http://lists.w3.org/Archives/Public/www-tag/2008Apr/0139.html>>
- [13] Hayes, P.J. and Halpin, H. *In Defense of Ambiguity. International Journal on Semantic Web and Information Systems*, 4 (2) 1-18. (2008)
- [14] Hermjakob, H., et al. *The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. Nat Biotechnol*, 22 (2). 177-183. (2004)
- [15] Hilbert, D. *Mathematische Probleme. Nachrichten von der Königl. Gesellschaft der Wiss. zu Göttingen*. 253-297. (1900)
- [16] Holtman, K. and Mutz, A. *Transparent Content Negotiation in HTTP*. IETF RFC 2295 (experimental), (1998)
- [17] Kunze, J. and Rodgers, R. *The ARK Identifier Scheme*. IETF Internet-Draft, (2008), <<http://tools.ietf.org/html/draft-kunze-ark>>
- [18] McCool, R. *Rethinking the semantic Web. Part I. Internet Computing, IEEE*, 9 (6). 88, 86-87. (2005)
- [19] Mingers, J.C. *An Evaluation of Theories of Information with Regard to the Semantic and Pragmatic Aspects of Information Systems. Systems Practice*, 9 (3) 187-209 (1996)
- [20] Nottingham, M. (2008) *HTTP Header Linking* <<http://tools.ietf.org/id/draft-nottingham-http-link-header.txt>>
- [21] Raman, T.V. *On Linking Alternative Representations To Enable Discovery And Publishing*. W3C TAG Finding, (2006), <<http://www.w3.org/2001/tag/doc/alternatives-discovery.html>>
- [22] Shannon, C.E. *A Mathematical Theory of Communication. Bell System Technical Journal*, 27. 379-423, 623-656. (1948)
- [23] Stromback, L. and Lambrix, P. *Representations of molecular pathways: an evaluation of SBML, PSI MI and BioPAX. Bioinformatics*, 21 (24) 4401-4407 (2005)
- [24] Thompson, H.S. and Orchard, D. *URNs, Namespaces and Registries*. W3C TAG Finding, (2006) <<http://www.w3.org/2001/tag/doc/URNsAndRegistries-50>>
- [25] Vinoski, S. *Serendipitous reuse. IEEE Internet Computing*, 12 (1) 84-87 (2008)
- [26] W3C Style: *Cool URIs don't change*. <<http://www.w3.org/Provider/Style/URI>>
- [27] W3C TAG Issue (2002) *What is the range of the HTTP dereference function?*, <<http://www.w3.org/2001/tag/issues.html#httpRange-14>>
- [28] W3C TAG Mailing Archive (2007) <<http://lists.w3.org/Archives/Public/www-tag/2007Oct/0050.html>>
- [29] W3C TAG Minutes <<http://www.w3.org/2001/tag/2008/05/20-minutes#item06>>
- [30] W3C Technical Architecture Group Jacobs, I. and Walsh, N., (2004) *Architecture of the World Wide Web, Volume One*. <<http://www.w3.org/TR/webarch/>>
- [31] Wang, X. (2007) *URI Identity and Web Architecture Revisited* <<http://dfdf.inesc-id.pt/tr/web-arch>>
- [32] Wang, X., et al. *Ontology Design Principles and Normalization Techniques in the Web*. in *Data Integration in the Life Sciences*, Springer (2008) 28-43.
- [33] Wang, X., et al. *From XML to RDF: how semantic web technologies will change the design of 'omic' standards. Nat Biotechnol*, 23 (9) 1099-1103 (2005)
- [34] XRI TC *OASIS Extensible Resource Identifier (XRI) TC*. <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri>
- [35] XRI Wiki: (2008) *Boeing XRI Use Cases* <<http://wiki.oasis-open.org/xri/BoeingXriUseCases>>