

# W5: The Five W's of the World Wide Web

Massimo Marchiori<sup>1,2,3</sup>

<sup>1</sup> The World Wide Web Consortium (W3C)

<sup>2</sup> MIT Lab for Computer Science, Cambridge (USA)

<sup>3</sup> University of Venice, Italy

`massimo@w3.org`

*Keywords:* WWW, Trust, Semantic Web, HTTP.

## 1 The five W's

The World Wide Web is a Web of information. Information can be more or less qualified, more or less usable, more or less usable by automatic processors. Information of the most different kinds, that can be reused for a lot of purposes. So how do we treat this information, how do we give some order, and possibly help its intelligent reuse?

Journalism has had the same problem since its inception: you have to report and classify a bit of information, but here “information” is as wide as the information we have nowadays in the WWW. So, what's the way out? One way out, which proved to be quite successful, is to use the so-called five W's, which are five axes that somehow identify the information event. These are the well-known and self-explanatory:

- WHAT
- WHERE
- WHO
- WHEN
- WHY

So, what about reusing this five W concept for the information present in the Web?

Historically, the five W's have already been used explicitly inside some application, for instance in XML dialects (cf. [5]), but what about reasoning about them at the most abstract level? Can they help, for instance in building a better Semantic Web?

## 2 Trust

The problem of Trust is a fundamental one in computer science, and in practice in the WWW. In order to talk about trust, we can try to give a more or less formal definition that we can later reuse to define some terminology. So, in general we can define:

**Definition 1. (Trust Scenario)**

A trust scenario is a quintuple  $(\mathcal{T}, \mathcal{R}, \mathcal{U}, \mathcal{S}, \tau)$  so defined:

- A "trust property"  $\mathcal{T}$  (that can be computationally intractable).
- A "test property",  $\tau$  (that is usually computationally tractable).
- A "universe"  $\mathcal{U}$  of entities (e.g., software agents, persons, etc.).
- A number  $\mathcal{R} (\in [0, 1])$ , indicating the "real" probability that  $\tau$  implies  $\mathcal{T}$ .
- A mapping  $\mathcal{S}$  from  $\mathcal{U}$  to  $[0, 1]$ , indicating that the "subjective" probability for an entity  $e \in \mathcal{U}$  that  $\tau$  implies  $\mathcal{T}$  is  $\mathcal{S}(e)$ .

Note that a trust scenario usually is not fixed but depends on an *environment*  $\mathcal{E}$ , which can contain the information on how to compute the probabilities, and that can be itself dependant on a number of factors, like time for instance.

In the following, when talking individually about entities and test properties, we shall always mean them within an understood environment and trust scenario (an " $\mathcal{E}\text{-TRUST}$ ").

Having defined what a trust scenario is, we can now use it to somehow formally define when problems with trust occur, i.e., when we have deception:

**Definition 2. (Deception)**

Deception occurs for an entity  $e$  when

$$\mathcal{R} \ll \mathcal{S}(e)$$

So, in general, we can say that in a trust scenario deception occurs when there is an entity such that deception occurs for it.

The severity of a deception could of course be quantified in various degrees, both locally for an entity  $e$  (e.g. by using the gap measure  $\mathcal{S}(e) - \mathcal{R}$ ), and globally by measuring its diffusion in the universe  $\mathcal{U}$  (e.g., in case of a finite universe, by averaging the local gap measure, or by fixing a threshold and measuring how much of the universe has a deception higher than that).

### 3 The Cost/Benefit

In the WWW, resources do not come for free, but there is a cost for creation and modification. Every solution for the WWW may bring some *benefits*, but usually also implies new creation/modification of information, and this *cost* must be taken into account, because that could be a big obstacle to the widespread adoption of such solution. Therefore, the parameter to take into consideration for success is the ratio *cost/benefit*. The cost/benefit (for instance, to diminish deception of some trust scenario), must be sufficiently high for users to adopt the solution and to build critical mass, so to create a possible network effect.

## 4 The WWW

We consider the World Wide Web in its approximation of “universal information space” where there are certain resources that are retrieved by dereferencing a certain URL. In other words, more technically, we just consider the Web under the assumption that the HTTP GET method is the only one to be used<sup>4</sup>

So, we can view the WWW as a “dereference map”  $\delta$  from URLs  $\rightarrow$  byte streams, with the intended meaning that  $\delta(u) = s$  if and only if, in the real WWW, there is a machine such that retrieving (GET) the URI  $u$  gives as a result the byte stream  $s$ .

When we later add semantics and meaning (depending on the particular application we use), we are essentially using an interpretation (let’s say  $\mathcal{I}$ ) of such web objects, that can give us more knowledge. That is the one that can allow, in trust scenarios, to lower deception.

Most of the times W3C sets up a standard (for example, for the Semantic Web),  $\mathcal{I}$  is refined.

## 5 The light five W’s

It is of utmost importance to minimize the cost of representing additional information in the WWW. This means that we should strive to obtain the information given by the five W’s in the most economical possible way, almost “zero-cost” if possible. Is there such a way? The answer is yes, at least for four or the five axes:

zero-cost WHAT == the resource (at least the message-body)

zero-cost WHERE = yes, the URI of the resource (Content-Location or Request-URI)

zero-cost WHO = yes, the URI authority (Host)

zero-cost WHEN = yes, the time when the resource was transmitted (Date)

zero-cost WHY = no.

In the following, when applicable, zero-cost W’s are understood.

## 6 The W1

What does it mean for a standard or for an application to be “Web”? In many cases, such standard/application doesn’t take into account the mapping  $\delta$ , but just takes into consideration the *message-body* (cf. [2]) of the image of  $\delta$ , in some cases integrated with the information about their MIME type. Simply speaking, this is tantamount to considering “web pages”.

Restated, such standards/applications are posing the WHAT axis equal to such web pages.

---

<sup>4</sup> In fact, this approximation gathers, at least architecturally, a good part of the WWW, as GET is architecturally a “universal operator” (in the sense of category theory) for most of the HTTP methods that collect information.

This is the starting point, and we can therefore define a first kind of World Wide Web:

$$W1 = \text{WHAT}$$

The current architecture of the Semantic Web stays in the W1 (where WHAT = message-body).

The problem is that, to build a reasonably effective Semantic Web (or in any case, to increase the semantic content, therefore diminishing deception) can have a very high cost.

## 7 W2 to W5

Another possible approach is to extend the W1 using the information provided by the other W axes.

Therefore,  $\mathcal{I}$  (the W1) can be increasingly integrated with the zero-cost WHERE, WHO and WHEN, giving three flavors of W2 ((WHAT, WHERE), (WHAT, WHO), (WHAT, WHEN)), two flavors of W3 ((WHAT, WHERE, WHO), (WHAT, WHERE, WHEN), (WHAT, WHO, WHEN)), and one W4 (WHAT, WHERE, WHO, WHEN).

## 8 Into Action

The W's give a kind of temporal modal logic: WHERE == world , WHO == world, WHEN == time. As common to modal logics, statements expressed in the same world can usually combine seamlessly, using the operators that the interpretation  $\mathcal{I}$  provides; as WHERE specializes WHO, this means that choosing a W2 or W3 with a WHO (and without a WHERE) will generally allow many more inferences than choosing a W2 or W3 with a WHERE.

On the other hand, the WHEN component is troublesome, as it represents a time instant, and so in general composition becomes practically impossible. Therefore, in order to allow a more useful use of WHEN, we can relax the composition rules, which is equivalent to change our interpretation of the timed logic.

For instance, one possible choice could be to employ some assumption of *local time consistency* (cf. [4]), therefore assuming that web resources stay somehow stable within some time intervals. This changes the interpretation of WHEN from a single instant to a time interval, allowing more inferences to take place. The price is that the approximation given in the choice of the stable time interval will likely make the deception increase, so there is a tradeoff. However, this tradeoff can be mitigated by using appropriate probability distributions of the "local stability" of a resource (therefore, passing to fuzzy/probabilistic reasoning).

Another choice is change the definition of WHEN, which is now rather simplistic (Date), and add for example the information about cacheability of the resource, and the expiration date: this gives right away a timed interval structure, which can be quite useful. The price to pay is that appropriate cache

information can have a cost. However, the benefits are quite high, because this information not only can help produce many more useful inferences in a W2, W3 or W4, but help in general the performance of the WWW (the primary reason in fact why cache information is present...). So, this approach might be worth exploiting,

Finally, of course, more sophisticated approaches are possible, where some or all of the information in the WHERE/WHO/WHEN/WHY axes is refined by integration with the information in  $\mathcal{I}$ . This intermediate solution can be the right way to overcome the limitations of the simplest W2, W3 and W4 solutions, while still keeping reasonably low the cost/benefit ratio.

## 9 Skews

The approach that we have seen so far is based on principles, but it has to be noted that other complementary views must be taken into consideration, when analyzing for instance trust scenario. Problems may occur, coming from malicious attempts to increase deception over time: in such cases, it is not uncommon to use all possible means: many trust problems on the Web usually occur because of so-called *information-flow skews*. A skew occurs when there is a treatment of the information flow in the WWW that departs from the high-level standard architecture of the Web, and that the user cannot see. There are at least three main skews that we can categorize:

- The *Visual Skew*
- The *Navigation Skew*
- The *Protocol Skew*

The *Visual Skew* occurs when not all the data flow goes back to the user, and can be synthesized with the slogan

“What is you see is not what you get”

In practice, this skew exploits the possibility that how a resource is rendered on the screen/medium (and so, what the user perceives) can be much different from what is actually in the resource.

One of the classic cases where Visual Skew shows its appearance is the so-called *search engine persuasion (sep)* (cf. [3]), also sometimes known (improperly) as search engine spam. Sep is the phenomenon of artificially “pumping up” the ranking of a resource in search engines, so to get a higher position (with all that means in terms of visibility and advertisement). Most of the techniques used in sep just profit in various ways of the visual skew, so to apparently present to the user a certain resource, which is quite different under the surface.

The *Navigation Skew* occurs when not all the WWW navigation is specified by the user. For instance, if we click on a link (i.e., request a resource on the WWW), we expect that we are just fetching the corresponding page. But this is not true: for example, frames and images are automatically loaded for us. This

apparent facility, however, leaves the door open for the navigation skew, as it means essentially that the authors of a resource can make us click on the page they want (!). Well-known examples of use of the navigation skew are banner ads and pop-up windows, all employing this skew in its various flavors. But even worst, the navigation skew makes possible applications that are potentially quite dangerous for users, like *tracking systems* (a la DoubleClick and Engage). Typically, such privacy-risky applications might employ a combination of skews (for instance, using so-called “web bugs”, images that use the navigation skew to send data, and the visual skew to hide, therefore resulting invisible).

The *Protocol Skew* occurs when the WWW protocols (e.g. HTTP) are abused (for instance, turning a stateless connection into a connection with state). For instance, the HTTP information flow in some cases should be from server to user (i.e., if we request a page, its only the server that gives us information). But this architectural principle is not always followed in reality, as for example many sites tend to collect so-called “clickstream” information (what you requested, when you did it, what is your computer internet address, etc). Again, this skew allows to collect information “under the rug”, and can therefore become quite a problem for the user’s privacy. Such problem can be worsened a lot when abuse of this skew is performed via aggregation: for instance, use of dynamic links (URIs that are generated on the fly) together with appropriate use of other clickstream information can make such tracking easily work not just for a single click, but for an entire session.

Therefore, every practical use of W1, W2, W3 or W4 have to take into account the potential danger, that “light” solutions can be necessarily prone to a higher risk in terms of possible deception

## References

1. T.Berners-Lee, R.Fielding, L.Masinter, *Uniform Resource Identifiers (URI): Generic Syntax*, IETF RFC, 1998.
2. R.Fielding, J.Gettys, J.Mogul, H.Frystyk, L.Masinter, P.Leach, T.Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, IETF RFC, 1999.
3. M.Marchiori, *Security of World Wide Web Search Engines*, Proceedings of the Third International Conference on Reliability, Quality and Safety of Software-Intensive Systems (ENCRESS’97), Chapman & Hall, 1997.
4. M.Marchiori, *The Quest for Correct Information on the Web: Hyper Search Engines*, Proceedings of the Sixth International World Wide Web Conference (WWW6), 1997.
5. M.Marchiori, *The XML Documentation Markup*, W3C, 1999.