International Journal of Computer Applications (0975 8887)
Volume 108 - No. 17, December 2014

A Foundational Axiomatic Modeling of the Web

Massimo Marchiori
University of Padua
Via Trieste 63
35121 Padova, Italy

ABSTRACT

The Web needs Semantics, let's give it Semantics. This mantra is at the origin of the Semantic Web effort, which tries to enrich the classic Web with more refined layers of information. However, despite its good intentions, the Semantic Web didn't succeed yet, and has remained unexpressed in its potential. In this article we try to analyze the possible reasons, and also the tension that the Semantic Web has with other information approaches, for instance XML or HTML. We emphasize the need for consideration of the more comprehensive social environment, that has instead so far been neglected in the mainstream web reasoning arena. Then, we introduce a new axiomatization of the Web, that can be used as a first foundational base in order to define and reason on information in the Web world.

General Terms:

World Wide Web, Automated Reasoning, Semantic Web

Keywords:

Algebraization, Axiomatization, Formal Models, Deduction, Inference, WWW

1. INTRODUCTION

The need for better information handling on the Web is evident, and has brought to several efforts in order to tame the amount and kinds of information available nowadays on a global scale. The most notable effort in this direction is the Semantic Web, a concept first envisioned by the inventor of the Web, Tim Berners-Lee (see for instance [2]), and then further developed in a variety of ways and standards. But this effort, despite good progress (see for instance [8] for a quantitative analysis) has still to reach success, and mass adoption is seen as far if not hard to achieve at all. In this paper, which is a revised and expanded journal version of [13], we try to analyze what can be the reasons for such slow adoption, and also start providing new possible directions, via an algebraic axiomatization for the Web that can seen as a foundational attempt to successfully reason in-the-large, beyond the boundaries that are common of today's solutions.
The Semantic Web, already in its foundational layer, the Resource Description Framework (RDF, cf. [11] and more generally [15]) has been controversial, and subject of several criticisms coming from other web information worlds. For example, another mainstream effort to structure information has been XML (cf. [4]), and the RDF and XML camps have often battled each other, with communities of people thinking the other side of the fence is doing things "the wrong way".
Given XML's relative success, and the current dual lack of success of the Semantic Web/RDF, it is normal that the latter has been often criticized, using the following "fundamental question":

Q: What can you do with RDF that you can't do with XML?

The fundamental question is both tricky and crucial. This question has been source of embarrassment, and of misunderstandings, for both worlds, and has somehow contributed to the lack of proper understanding of the potential of the Semantic Web in the context of the bigger XML world.
We were saying the question is tricky. The classic general answer which is given is:

Q: What can you do with RDF that you can't do with XML?

A: Semantics!

This usually leaves the XML-World unsatisfied, because this is in fact a very fuzzy answer. Saying that with RDF you can do semantics, equals more or less to say that with the Semantic Web you can do... semantics, which doesn't sound too good to critical eyes. So then, the "socratic dialogue" goes on, and the XML-World usually replies with

XML-World: What do you mean?

More or less, the debate between RDF-World and XML-World then goes on like this:

RDF-World: With RDF I can do X.

XML-World, Well, I can do X with XML too, so what?

RDF-World: But, with RDF I can do Y.

XML-World, Well, I can do Y with XML as well, so what?

(and so on, and so on...)

The point is that the answer is in fact quite easy, and it is one that few people in RDF-World would dare to mention explicitly:

Q: What can you do with RDF that you can't do with XML?

A: Nothing!

This comes trivially from the fact that RDF is XML, and therefore, there's no magic in RDF: RDF is just a dialect of XML, and as such, there's nothing RDF can do ”more” than XML: the question, posed this way, is just bogus.
But so, does this mean the XML world is right, and that the Semantic Web is superfluous?

2. CLOSED VERSUS OPEN WORLDS

The answer to the previous question is not that easy: it really depends on what level of precision we want to analyze. It is certainly true that with XML you can do anything you want, but that doesn't prevent RDF (and the related tower of technologies) to be a successful dialect/specialization of XML, like there are many around. But specialization for what, precisely?
XML has been labeled as the best invention after peanut butter: versatile, flexible, powerful. However, there is one thing for which XML, at least apparently, doesn't work so well: aggregating information.
XML's strength is its specialization capabilities: given an information locale, everybody can easily write a local dialect to express that information. In other words, XML works extremely well in the *closed world* context: an environment where there is a centralized vocabulary control. However, there is another scenario, which didn't fit the original design of XML: the *open-world model*, where there is no centralized vocabulary control. In such scenarios, everybody can develop its own local dialect, and then the big problem is how to exchange information between the different vocabularies, integrating various information sources that have no control over each other. Like for the Tower of Babel, where the multitude of languages has been the disgrace of Humanity, in the open-world model the different languages can provoke heavy interoperability problems (what linguistics call very appropriately the Lost in translations effect).
RDF, more or less consciously, was designed with this fundamental goal in mind (besides the related ”give more semantics” mantra): reducing almost to zero the complexity of aggregating information (which, essentially, becomes a merge of graphs). The connections among information pieces are established via the URIs: so, when merging graphs, nodes are considered equal if they have the same URI. Therefore, URIs become the fundamental key to distinguish web object. This choice is compatible, and actually stems from, one of the very first Web Axioms stated by Tim Berners-Lee (the so-called Universality 2 axiom, cf. [1] and compare with the later [10]): meaningful resources on the Web should be identified by URIs.
Thus, RDF is (also) XML, but RDF has been designed to work in the open-world model: while XML works better in the closed-world model, RDF does in the open-world model.

3. THE UTILIZATION MODEL

So, a first important point that distinguishes ”generic” XML from RDF is the complexity of information aggregation. While being an important point, that alone doesn't give the whole picture.
In fact, the Web is, as a whole, semantically speaking, a huge open-world model: so, how come that the Semantic Web didn't rapidly gain success? Something must have gone wrong, and to trace that, we need to start back from the original definition that Tim Berners-Lee gave of the Semantic Web: an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation”. Computers... and people! What about the people?
What is missing in the equation is the *utilization model*, i.e., the complete benefits (goals) that the new technology is supposed to provide.
Saying that RDF ”works better” in the open-world model is a simplistic assertion, as we haven't quite defined what ”better” means. If better means aggregating information, the assertion is correct. But aggregation alone isn't what the Semantic Web promise to do (if it were so, the benefits alone wouldn't be quite clear): the goals of the Semantic Web are more ambitious, and for that reason, the original idea of the Semantic Web includes the well-known ”Semantic Web tower” (see for example [2]), i.e., a full tower of technologies that better describe the operational model, and therefore help clarifying the benefits.
So for instance, aggregation of information isn't much helpful if we don't have a clear working model that allows us to benefit from that feature. In order to exploit information aggregation we can then for example also include a logic into the picture: a logic allows to make deductions, and so in principle augments by far the benefits of having aggregated information on the Web. Initial step into this direction have been done with the RDF Semantics, RDF-Schema, OWL, and this line has been continuing more recently with the work of W3C's RIF effort, devoted to specify a Rules Interchange Format that will allow even more flexibility in ”programming” the rules shaping web information. This is all consistent with the big view of the Semantic Web Tower.

4. THE COSTS

But then, there is also the other side of the coin, the dual part that has to be considered every time that we want to analyze the behavior of a successful technology: the *cost factor*.
The overall cost is in general a complex thing to compute, but roughly, the cost of a technology can be seen as the sum of two components: the *technological cost* (the cost for the machine), and the *social cost* (the cost for the people). We can summarize the concept this way:

$$\text{Cost} = \text{Technology} + \text{Society}$$

Both aspects, technology and society, are equally very important, and this is the paradoxical aspects that is often neglected: that a technology does not belong to the technological realm only, but also to the societal one. What have happened so far is that the societal cost of the Semantic Web hasn't been object of much attention, and the whole design has been centered on the technological cost, making best efforts to ensure that the technologies in the Semantic Web Tower would have a relatively low technological cost. But in the overall Semantic Web operational model, the scenario is much bigger than just the computational complexity of a logic: it includes the much wider scenario of the Web, its information flows, machines, and the people. Therefore, we need to rethink the situation and not just wear the eyeglasses of the technologist, caring mostly about the computers (classic semantic web stack). Sure, there is the need to monitor and balance the technical cost, but also to consider at least another dimension for the social cost (what has been called the *P axis*, P as Perception/People, in [12]).
Only when we have a complete measure for the cost we can proceed to measure the cost/benefit ratio (shortly, C/B), which is a major indicator of success, especially in environment like the Web.

4.1 Costs and Benefits

The C/B ratio provides a uni-dimensional space that can give a rough estimate of the chances of success of a technology (ranging from 0, the optimum, to infinity, the worst). Minimizing the

cost/benefit can happen in a variety of ways, depending on the balance between C and B. In the Web, the important thing to take into account is the dynamics of C and B within the web environment and the users. For instance, in a web-wide application cost usually grows at least linearly with the size of the web (or of the sub-web/community taken into consideration), which can be extremely dangerous. On the other hand, B also in such applications usually depends on the size of the user base, which is very low in the starting adoption phase. Therefore, if we are not careful the corresponding dynamic system will not lead to a success situation, because the too high C/B in the initial phases will prevent an evolution that makes the C/B decrease and reach a wide enough user base. So, in order to produce a network effect, either the initial cost has to be extremely low, or the initial benefit has to be very high.
This cost benefit analysis is crucial to the success of a Web Technology, and so to the shapes that the Web can and will take in the future. In order to better understand some of its complexity, we show four scenarios and see where and how cost is a key factor.

4.2 The Cost of Web Addresses

The previous C/B discussion naturally leads to consider: what are the costs of the Semantic Web? An interesting exercise is to measure the technological cost for the semantic web architecture (e.g. in the Semantic Web tower). The analysis will then reveal, in fact, a nice result: the technological (computational) cost is usually low/moderate according to where one sits in the Tower (although interestingly, even in this respect, computational cost has started to grow a lot, see for instance the logic behind the higher layer of OWL). However, when one views at the historical progression of the Semantic Web (still ongoing. . .), the situation is that there is an overgrowing set of specifications: RDF Model & Syntax / RDF Schema / RDF/XML Syntax revised / RDF Vocabulary Description Language / RDF Concepts and Abstract Syntax / RDF Semantics / OWL (OWL-DL / OWL-Full)/ SPARQL / RDF-A / Rules. . . and the list is still growing.
So, what has been happening here? Will the user be able to sustain the social complexity that these layers are going to produce? The answer, for the moment, is in front of everybody: not yet. The overall cost seems too high for the moment. And this comes from a variety of factor, given that as said, the scenario to consider is much bigger than what has been formally analyzed so far (computational complexity of logics): the Web, the people, information flows.
For instance, let's just revisit the basic association mechanism of the semantic web: aggregation via URIs. An old gag that used to be around in the semantic web circles was the following:

Q: How many Semantic Web scientists does it take to change a light bulb?
A: Ten. One to screw the bulb, and nine to agree on what a light bulb is.

This gag is significant for the suggestion it is giving: it's hard to all agree on a concept. If URIs are meant to be identification names, they are the centralized part of an otherwise decentralized and distribute environment, the web. But how to achieve consensus without control? In other words, there is a significant *social problem* with URIs when they are used as universal aggregators of information. This gives raise to the *URI Variant problem*: in general, there can be many variants (URI) for the same concept. The URI Variant problem is particularly bad in view of the *URI Variant Law*: utility of a URI can decrease exponentially with the number of its variants (in other words, the worst-case is exponential).
This is not enough, because the social problem is not just on what common URI to agree, among many. There is also the other side of the coin, which is much more difficult: how to agree on the semantics of a specific URI. This is sometimes called the *URI meaning problem*: in other words, for two different people in the web the same URI might well mean different things (after all, there is no centralized interpretation for URIs). This problem is rather severe, because it does not simply affect computational complexity (like the URI variant), but deeply touch the relationship between the web and the people who interact with/in it.
So, all in all, what seemed a strength of the Semantic Web, i.e., almost zero cost for aggregating information, is now revealing deeper faces: while the direct technological cost is indeed very low, there is an underlying social cost that is in fact quite high.
Therefore, this extra variable of the social cost, makes the original simplistic observation, that information aggregation in the Semantic Web is very easy and effective, not quite true any more, and emphasizes the lack of a precise operational model and consequent cost/benefit analysis that have occurred so far.

4.3 The Cost of Navigation

Social costs are not limited to URIs, of course, but they can pervade the same data model. Data structures can have a rigid architecture, or lean towards a more liberal framework, therefore going from the areas of structured data, passing thru the intermediate realm of semi-structured data, and ending in the opposite extreme, the area of unstructured data. Within this wide spectrum, we find for instance in small-size data management on one extreme (structured) spreadsheets and the table model, and on the other extreme (unstructured) things like Zig-Zag, the innovative (for the time) concept by the hypertext pioneer Ted Nelson (cf. [14]). In large(r)-size data management, going on with the parallelism, we find relational databases and the relational model (structured), then we can proceed with XML (semi-structured), then ending with RDF (unstructured).
It is therefore interesting to follow the parallelism, and note that the previous unstructured models (like Zig-Zag) didn't have much success, while the more structured ones did. What are the main reasons? This can be explained by using a so-called *short blanket principle of data handling*: if you stretch the flexibility aspect (benefit), you lose in efficiency (cost).
Note that here efficiency doesn't just mean computational efficiency, but efficiency in-the-large, also for the user. In fact, studies by the author have shown that one can quantify the degree of lost in navigation (that is to say, informally, the capability by the user to grasp the data structure, and to navigate without errors in it): the lost in navigation effect increases (not surprisingly) from structured to semi-structured to unstructured. What is more surprising is that there is quite a gap when passing to unstructured models like RDF and graph-like ones. In other words, the amount of flexibility that these kinds of models give has a very high price that the user needs to pay. This can explain more formally why unstructured data didn't gain so far the wide success they were expected to. What this also means is that, in order to lower the cost/benefit ratio, there is the need for extra efforts to raise the benefit.
On a related side, the gap occurring in-between structured and semi-structured data is comparatively rather small, which might also explain why technologies like XML managed to gain success, despite the initial dominant position of structured data approaches.

4.4 The Cost of Privacy

Let's sweep out of the XML / Semantic Web scenario, and for the sake of illustration, try to see some other examples of more specific technologies and their related cost/benefit.

A first pair of examples is interesting, and comes from privacy technologies developed by the author: P3P and APPEL.
P3P (standing for Platform for Privacy Preferences) is the world standard for Privacy on the Web ([6]). Analyzing the P3P specification will easily show that the P3P technical cost is very low (in fact, not surprisingly, as this was a crucial requirement). As far as the Social Cost is concerned, it is moderate for site maintainers: the moderate complexity essentially stems from building the privacy policy for the site, although this can be ameliorated by specific tools, and in any case approximate policies can be written that are much easier; complementary, publishing the privacy policy is very easy and has a very low cost.
Now let's turn our attention to the benefits side. The benefit is moderate for users (this can be evaluated by using the many privacy surveys available), whereas, interestingly enough, it was very high for site maintainers. The reason? When Internet Explorer passed from version 5 to version 6, it actually incorporated the P3P technology, and in a very stringent way: sites not P3P compliant had severe problems and their cookies were essentially blocked by the browsers. This crucial step provoked a huge rise in the benefit of implementing P3P (even if just at the site maintainers side), and therefore boosted the C/B ratio of P3P, despite cost wasn't low (this C/B boost can be also verified by using the statistical P3P dashboards published by Ernst&Young on the subject). Next, protests by web site owners had Internet Explorer remove this strict policy, and as a result the C/B ratio considerably diminished, slowing down P3P adoption.
Now, we want also to consider the other side of the coin, as we said initially that we were going to consider a pair of technologies: P3P and APPEL. APPEL [5] is the companion technology to P3P: the acronym stands for A Privacy Preference Language, and it is a language that enables users (via their browser, for instance), to program on a fine level whether or not to enter a web site, according to the privacy level the site itself provides.
The technological cost for APPEL is moderate, as it can be easily seen. On the other hand, the social cost is high: users need to get knowledge of the privacy possibilities, and to adequately shape a set of preferences. This was too much, given both the relative user interest in privacy (versus content, for instance), and the complexity of programming/shaping a fine level behavior. On the other hand, the benefit here was also relatively small, as the additional privacy control wasn't enough more than for instance some easy pre-defined levels (that Internet Explorer in fact implemented). As a result, the C/B ratio never got sufficiently low, and APPEL didn't fly (in fact, it was never promoted to W3C Recommendation, and remains a proposed technology).
On a wider historical perspective, it would have been much better for W3C, if still wanting to target fine privacy user control, to use a more general-purpose reasoning tool (so to at least try to increase the benefit), either in the Semantic Web area (viewing privacy policies as RDF data), or in the more classic XML data handling area (using technologies like XQuery for instance).

4.5 The Cost of Blogging

Leaving aside privacy, another good example to consider is a mainstream one: *blogs*. Blogs have been so successful that it's interesting to see what the reasons are, especially given that, historically, they had been greeted with much scepticism. Why? Because, from the technological viewpoint, blogs are a relatively trivial technology. This is the reason why in the technological world, blogs had initially received so little attention: trivial technology, no real innovation, and why should people bother about writing a diary online? Things went differently than those critics planned. Sure, the technological level of blogs might well be low, but what really matters is the C/B ratio. Let's see things deeper. The benefit, at least initially, was moderate (now the blog networking/echo effect has raised this initial value to high). The technical cost for blogs is very low. And, even more important, the social cost is extremely low (what is easiest than writing?). As a result, in the initial step a moderate benefit was more than adequately counterbalanced by very low costs, which made the C/B ratio low enough to make blogs spread. A paradigmatic social example that well illustrates the blog C/B power is one of the Times magazine *European Heros* for the year 2005: Beppe Grillo. Grillo (now turned into a politician) was at the time a famous Italian comic actor, which at a point of his career became totally against computers, arriving, in his shows, to smash a computer with a hammer. Then, he discovered blog technology. The entry level was so easy, that he could approach the technology without the complexity that had lead him to previously hammer down computers. And, in his own words, he gave that thing a try, even if the benefits weren't clear at all to him. Nowadays, Grillo's opinion on the Internet, thanks to the blog, have radically changed, and he is an Internet evangelist (!), with his blog (cf. [7]), at the time of the Times award ranked as number one blog in Europe and first no-USA blog, and even nowadays still number one in Europe and among the top ones in the world.

4.6 Other Biases

The same ”technological bias” seen above for blogs equally applied, historically, to many social systems, dubbed as poor technological innovation: think for instance Facebook or even worst, Twitter. Twitter is even more extreme to some extent because its basic technology was (and still is to some extent) radically simple, if not trivial. Its same text limitation, the famous 140 character upper limit, originally came from phone text limits. But overall, the cost of using the system has always been tremendously low, like blogs and even more just because of this text limit. Which is the reason why, years later and with Twitter now an established giant of the web information space, the text limit has stayed: not any more for technological reasons, but merely to keep the overall cost factor low.
Relatedly, we have seen the same bias in another information battle: the one between XML and HTML. To some extent this fight has mimicked the one between RDF and XML: paradoxically and ironically, the whole dialogue in Section 1 could be rewritten by simply substituting RDF and XML with XML and HTML respectively. XML, the allmighty information structuring language, defeated by HTML? Something looks wrong here, but the paradox is only apparent: what has happened is that eventually the cost factor has been weighted, and HTML has got its revenge: structuring information via XML doesn't provide, at the moment, so many additional advantages in-the-large *to the people*, with respect to good old HTML. This is why the HTML route has become so successful, as witnessed by the recent declaration of HTML5 as a W3C world standard (cf. [9]).
So all in all, innovation pertains to people, not to technology only, and people need to be factored in when considering the overall picture.

5. OPERATIONAL MODELS

Trying to grasp the operational model in which the Semantic Web operates is a significant task: as seen, entering people in the loop makes the C/B analysis far from trivial, and can lead to important

insight. But as we have said, the overall operational scenario is bigger, involving the Web, its information flows, machines, and the people. A natural goal would be to have a more formal way to discuss the operational model, so to be able to do more precise and scientific kinds of analysis. This also implies that we need to fill in what are the information flows present in the Web: before even starting to formally discussing properties of an operational model living in the Web, we ought to have a formal model that we can reason upon, describing the information flows mechanics that underline every higher-level handling of information.
Going alone the line of a more formal approach, the reason why it's not easy to answer the question "what is the Semantic Web" is that the semantic web is essentially a way to build a *shared information system*. "Shared" here means essentially that the distributed nature of the Web enables information to be spread out on the Web, and then *composed* together. Therefore, the problem here is that to define how the semantic web works, we need to take into account in the big picture not just the specific language (XML dialect), like RDF, that is used to encode knowledge. We also need to take into account the way information is composed, that is to say, we need to consider in the picture the Web and the related process model. In other words, the fundamental equation to take into account, when wholly describing the semantic web (from the technical viewpoint), can be roughly stated as

Semantic Web = Semantics + Web

So far, just the "Semantics" part has been considered, and the "Web" part has mostly been left out, or considered as an appendix.

6. TOWARDS A WEB ALGEBRA

We start the merge of the Semantics and Web part by introducing an algebraic axiomatizaion of the Web, that generally models the Web as an information space and attemps to properly formalize information injection and extraction in an extended and layered way.

6.1 Infoshares and Infostructures

First and foremost, we need to set the stage for a general logical setting, where reasoning takes place:

DEFINITION 1 (INFORMATION SYSTEM). *An* information system *is a poset* $(S, \vdash_S)$.

So, given an information system $(S, \vdash_S)$, and two elements a and b of S ($a \in S \ni b$), we say that "b follows from a", or also that "from a we can infer b", when $a \vdash_S b$.
Then, we go on with the definition of the more general environment where the various shared reasonings can take place:

DEFINITION 2 (INFOSHARE). *Given an ordinal* k*, a* k-infoshare *is composed by:*

—*A* gateways set Λ*, which is a semi-lattice.*
—*A* shared spaces set ***SSS***
—*For each* i*,* $0 < i \leq k$*:*
  —*An* information system $\mathcal{A}_i$
  —*A map* $TAKE_{\mathcal{A}_i}$: $\Lambda \times \textbf{\textit{SSS}} \to \mathcal{A}_i$
  —*A map* $PUB_{\mathcal{A}_i}$: $\Lambda \times \textbf{\textit{SSS}} \times \mathcal{A}_i \to \textbf{\textit{SSS}}$

We will usually omit the index k in a k-infoshare, and just talk of an *infoshare*. Also, given an infoshare, we will refer to members of its gateways set as "gateways", and to members of its shared spaces set as "shared spaces". In the gateway set, we indicate with $\perp$ the non-interference relationship (i.e., we write $\lambda \perp \lambda'$ to state that two gateways do not intersect).
En passant, note that in the above definition the $\text{TAKE}_{\mathcal{A}_i}$ and $\text{PUB}_{\mathcal{A}_i}$ are total (always defined): a relaxation of these assumptions is provided later in this paper.
An infoshare alone is little more than an algebraic container of the information flows: in order to have an information algebra, we need to enrich the structure with more axioms, that give some basic rules on the behavior of the information handling:

DEFINITION 3 (INFOSTRUCTURE). *An* infostructure *is an infoshare such that the following axioms hold:*

—***Incrementality**:*
$\lambda \geq \lambda' \Rightarrow TAKE_{\mathcal{A}}(\lambda, S) \geq TAKE_{\mathcal{A}}(\lambda', S)$
—***Locality**:*
$\lambda \perp \lambda' \Rightarrow TAKE_{\mathcal{A}}(\lambda', PUB_{\mathcal{B}}(\lambda, S, x)) = TAKE_{\mathcal{A}}(\lambda', S)$
—***Echo**:*
$TAKE_{\mathcal{A}}(\lambda, PUB_{\mathcal{A}}(\lambda, S, x)) = x$
—***Update:***
$\lambda \geq \lambda' \Rightarrow PUB_{\mathcal{A}}(\lambda, PUB_{\mathcal{B}}(\lambda', S, y), x) = PUB_{\mathcal{A}}(\lambda, S, x)$
—***Separation**:*

$$\lambda \perp \lambda' \Rightarrow$$
$$PUB_{\mathcal{A}}(\lambda, PUB_{\mathcal{B}}(\lambda', S, y), x) = PUB_{\mathcal{B}}(\lambda', PUB_{\mathcal{A}}(\lambda, S, x), y)$$

—***Freedom**:*
$\forall \lambda \in \Lambda.\ \exists \lambda' \in \Lambda.\ \lambda \perp \lambda'$

The axioms that constitute the infostructure can be described as follows:

—Incrementality: this states that the more gateways we use, the more information we get. This is what makes *partial evaluation* possible (sound), and so makes reasoning scalable in-the-large.
—Locality: this expresses the fact that if two gateways sets are in non-interference (w.r.t. the semi-lattice structure of the gateways set, i.e., they don't have a common component) then even if some publication occurs in a set, the information that can be taken out of the other set stays unchanged. So, separate areas/teams can safely work independently, and legacy information is preserved if appropriate separate gateways are used.
—Echo: if something is published, and then retrieved, there is no loss of information. This expresses safety of the publication process: every bit of information that is put in the SIS, can be retrieved, with no losses.
—Update: if something is published, this overrides previous publishing in the gateways. So, this axiom corresponds to the possibility to update old/incomplete/incorrect information.
—Separation publishing at two gateways sets that are in non-interference is independent on the order. This allows asynchronous processing between different teams and different applications: they don't have to "wait on each other" to publish information. With no cycles lost, this enables for a true parallel information system.
—Freedom: given a gateway set, there is always another gateway set that doesn't interfere. In simpler words, there's always room to expand and put new information.

Of course, the proposed modeling is an approximation of the simple information system, and more refined axiomatic systems can be introduced, depending on the level of details we want to analyze and model. Nevertheless, even such a simplified model can give quite some hints. For example, a consequence of the fact that the $\text{TAKE}_{\mathcal{A}_i}$ functions are total (i.e., always defined) is the *information compression property*, which says, roughly speaking, that the amount of information is independent on the gateways (even more informally, one can "compress" information coming from many gateways into fewer gateways). In the case of the Web (with URL sets as gateways), this means that one can also aggregate information within a single page (so, the semantic layer and URLs are somehow orthogonal). It is interesting to note that RDF has the compression property (and so, it is indeed an aggregator language), while XML (cf. [4]) does not, due to the in-famous unique root constraint. This shows that aggregation of information wasn't at all one of the design priorities of XML. In fact, given that compression is an essential feature for data handling, the compression property has later been re-introduced "under the hood" in the XML world, either using dummy root elements to do aggregation, or using the concept of XML collection (like for instance in the XQuery language, cf. [3]).

6.2 The Open-World Scenario

As an example of use of infostructures, we can for example start to model a simplified open-world scenario, where there are several systems that publish and retrieve information: what are some conditions that ensure a certain degree of consistency of the information, i.e., that allows safe aggregation between different information worlds? To answer this question, we can introduce a relationship that information system can satisfy:

DEFINITION 4 (INFO-EXTENSION). *Given two information systems $\mathcal{A}, \mathcal{B} \in$ **IS**, we say that $\mathcal{B}$ is a* info-extension *(briefly,* i-extension*) of $\mathcal{A}$, and write $\mathcal{A} \lhd \mathcal{B}$, iff*

$$\forall \lambda, \lambda' \in \Lambda, S \in \textbf{\textit{SSS}}, x \in \mathcal{A}. \quad TAKE_{\mathcal{A}}(\lambda, S) \vdash_{\mathcal{A}} x \Rightarrow TAKE_{\mathcal{B}}(\lambda, S) \vdash_{\mathcal{B}} TAKE_{\mathcal{B}}(\lambda', PUB_{\mathcal{A}}(\lambda', S, x))$$

The following two easily proved lemmata show that $\lhd$ provides an ordering structure:

LEMMA 5 ($\lhd$-TRANSITIVITY). *The relationship $\lhd$ is transitive.*

LEMMA 6 ($\lhd$-REFLEXIVITY). *The relationship $\lhd$ is reflexive.*

Before going on, we now need a few definitions.

DEFINITION 7. *Given an infoshare, we can construct the corresponding term algebra, which we call the infoalgebra. Then, given a term τ in the infoalgebra, we denote with:*

—*Active(τ) the set of all the $\lambda \in \Lambda$ that appear as an argument of any PUB in τ.*

—*Passive(τ) the set of all the $\lambda \in \Lambda$ that appear as an argument of any TAKE in τ.*

—*Before(τ) the initial element $S \in$ **SSS** used to build τ*

—*After(τ) the element $S' \in$ **SSS** obtained by "evaluating" τ, i.e. by applying all the functions TAKE and PUB present in τ.*

By easy axiomatic induction we then have the fundamental:

THEOREM 8 (PASSIVE-ACTIVE). *Suppose to have a term τ in the infoalgebra (say, τ), and the involved information systems form an info-extensionn chain, with $\mathcal{A}$ as top. Then we have:*

$$TAKE_{\mathcal{A}}\left(\bigvee Passive(\tau), Before(\tau)\right) \vdash_{\mathcal{A}} TAKE_{\mathcal{A}}\left(\bigvee Active(\tau), After(\tau)\right)$$

Note the importance of the Passive-Active theorem, that essentially shows that info-extensions provide a well-formed layering structure: in an infoshare there is a top $\mathcal{A}$, then $\mathcal{A}$ "stays the same" (remain consistent), unregarding of what has been happening in the infoshares below. This regulation of the information flows allows for a successful integration of information.
It is interesting to compare the information flow behavior of info-extensions with the current information modeling of the Semantic Web Tower, where no general information extension concept had been defined, for lack of a basic operational model to start with. As a result, extensions have been first interpreted as strict logical extensions, and then somehow "patched" so to ensure consistency of information in a web-free embedding modeling. Info-extensions, on the other hand, provide for a more flexible general mechanism that allows more flexibility, allowing to deal more nicely with scenarios where one can't enforce a one-solution-fits-all, but where there will be many competing solutions (at the same time, and also over time), possibly sharing common knowledge grounds.

7. BEYOND TOTALITY

The fact each PUB function in an infostructure is supposed to be a *total* function is quite an assumption. This, because from the Echo axiom, if a $\text{PUB}_{\mathcal{A}}$ is always defined (as, $\text{TAKE}_{\mathcal{A}}$ is always defined too) then information in $\mathcal{A}$ can't know about the gateway it came from. In other words, $\mathcal{A}$ is in a sense "separated" from the gateway structure, which is good as it makes for a simpler modeling, but also limitative in some scenarios.
The assumption of totality for the publication functions can be be relaxed, and a more sophisticated modeling taken into consideration. For instance, the infostructures axioms can be rewritten using a *Def*() functor that checks whether or not the specific instance of a publication function is defined:

—**Incrementality**:

$$\lambda \geq \lambda' \Rightarrow \text{TAKE}_{\mathcal{A}}(\lambda', S) \geq \text{TAKE}_{\mathcal{A}}(\lambda, S)$$

—**Locality**:

$$\lambda \perp \lambda' \wedge Def(\text{PUB}_{\mathcal{B}}(\lambda, S, x)) \Rightarrow \text{TAKE}_{\mathcal{A}}(\lambda', \text{PUB}_{\mathcal{B}}(\lambda, S, x)) = \text{TAKE}_{\mathcal{A}}(\lambda', S)$$

—**Echo**:

$$Def(\text{PUB}_{\mathcal{A}}(\lambda, S, x)) \Rightarrow \text{TAKE}_{\mathcal{A}}(\lambda, \text{PUB}_{\mathcal{A}}(\lambda, S, x)) = x$$

—**Update:**

$$\lambda \leq \lambda' \wedge Def(\text{PUB}_{\mathcal{A}}(\lambda, S, x)) \Rightarrow \text{PUB}_{\mathcal{A}}(\lambda, \text{PUB}_{\mathcal{B}}(\lambda', S, y), x) = \text{PUB}_{\mathcal{A}}(\lambda, S, x)$$

—**Separation**:

$$\lambda \perp \lambda' \wedge Def(\text{PUB}_{\mathcal{A}}(\lambda, \text{PUB}_{\mathcal{B}}(\lambda', S, y), x)) \Rightarrow \text{PUB}_{\mathcal{A}}(\lambda, \text{PUB}_{\mathcal{B}}(\lambda', S, y), x) = \text{PUB}_{\mathcal{B}}(\lambda', \text{PUB}_{\mathcal{A}}(\lambda, S, x), y)$$

—**Freedom**:

$$\forall \lambda \in \Lambda.\, \forall S \in \textbf{SSS}.\, \exists x \in \mathcal{A}.\qquad \exists \lambda' \in \Lambda.\, \lambda \perp \lambda' \wedge \textit{Def}(\text{PUB}_{\mathcal{A}}(\lambda', S, x))$$

More sophisticated modeling would also include the time variable (which now is implicit in the axioms), making the axiomatization more similar, in spirit, to a modal logic (although, somehow complicating the analysis).

8. CONCLUSIONS

Information handling on the Web is a tough cookie. The whole Semantic Web area is a good attempt, but overall has fallen short. Mass adoption has been extremely slow, and we have identified what can be the missing parts. What is needed is a much wider perspective, including the definition of operational models that take into account the technological structure (the Web, the machines) as well as its interactions with the societal structure (the information flows, the people). Here we have just started to scratch the surface, but it is obvious that a better science of successful information handling for the Web should come from a refined analysis that underlines models, axioms, principles and humans. Such analysis would help a lot in shaping successful standards and solution for web information, and also help to understand what are the real principles to focus on.
For instance, coming back to the original "XML versus RDF" clash described at the beginning, it is interesting to note how once identified an operational model, alternative possible solutions naturally arise: nobody prevents for instance to lower the cost for people that publish (PUB) information, by directly using XML instead of RDF. If this solution is chosen, then suitable interpretations of XML documents can be provided, so to define an infostructure (on this point, remember the basic discussion on the compression property in Subsection 6.1). Then, infostructures can be assembled by using infoextensions, in a scenario where information flows grow like a tree, with several possible branches and leaves. Given the relative drop in the cost factor, that avoids information duplication/reshaping in RDF, this might prove to be quite a viable approach for the future. In fact, this approach can be brought even further, by considering not just XML but also XHTML and HTML, so making the initial cost factor for publication of information radically drop: sure, the initial benefits would be lower too, but to start the network effect, after all, we maybe need to talk (and listen) to humans first, rather than just to machines. And nobody prevents from then having more specialized semantic web solutions, in the spirit of RDF and the Semantic Web Tower, grow as a branch of a bigger *Web Tree*.
Finally, note that the axiomatic system that we presented here concerns an initial, foundational modeling of the web environment. The next steps are to build upon this, and take into account the human and social aspects of the population dynamics, formalizing the concept of Cost and Benefit discussed in the previous Sections 3 and 4. We need to establish an overall reasoning environment where the whole picture is considered, and the formal actions are not just those within the web, but also those outside the web. We need to integrate and model people themselves or, to start with, some suitable approximation of people: cost and benefit.

9. REFERENCES

[1] Tim Berners-Lee. Universal resource identifiers – axioms of web architecture. http://www.w3.org/DesignIssues/Axioms.html, December 1996.

[2] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[3] Scott Boag, Don Chamberlina, Mary Fernandez, Daniela Florescu, Jonathan Robie, and Jerome Simeon (eds.). XQuery 1.0: An XML query language. http://www.w3.org/TR/xquery/, November 2010.

[4] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and Francois Yergeau (eds.). Extensible markup language (XML) 1.0 (fifth edition). http://www.w3.org/TR/REC-xml/, November 2008.

[5] Lorrie Cranor, Marc Langheinrich, and Massimo Marchiori. A P3P preference exchange language 1.0 (APPEL1.0). http://www.w3.org/TR/P3P-preferences, April 2002.

[6] Massimo Marchiori (ed.). The platform for privacy preferences 1.0 (P3P1.0) specification. http://www.w3.org/TR/P3P/, April 2002.

[7] Beppe Grillo. Beppe Grillo's Blog. http://www.beppegrillo.it/, 2014.

[8] Michael Hausenblas, Wolfgang Halb, Yves Raimond, and Tom Heath. What is the size of the semantic web? In *Proceedings of I-Semantics (2008)*, 2008.

[9] Ian Hickson, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O'Connor, and Silvia Pfeiffer (eds.). HTML5 – a vocabulary and associated APIs for HTML and XHTML. http://www.w3.org/TR/html5/, October 2014.

[10] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, volume one. http://www.w3.org/TR/webarch/, December 2004.

[11] Ora Lassila and Ralph Swick (eds.). Resource description framework (RDF) model and syntax specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, February 1999.

[12] Massimo Marchiori. The semantic web made easy. http://www.w3.org/RDF/Metalog/docs/sw-easy, 2003.

[13] Massimo Marchiori. How to talk to a human: The semantic web and the clash of the titans. In *Proceedings of the 16th international conference on Logic-based program synthesis and transformation*, volume 4407 of *LNCS*, pages 1–14. Springer, 2007.

[14] Ted Nelson. Zig-Zag® Software. http://xanadu.com/zigzag/, 1999.

[15] The World Wide Web Consortium (W3C). Resource description framework (RDF). http://www.w3.org/RDF/, 1997.