CoReWeb: From Linked Documentary Resources to Linked Computational Resources

Alexandre Monnin, Nicolas Delaforge, Fabien Gandon

CoReWeb: From linked documentary resources to linked computational resources

Alexandre Monnin
Université Paris 1 Panthéon Sorbonne
(Phico, EXeCO) /IRI/INRIA (Wimmics)
4 rue Aubry le Boucher
75004 Paris
Alexandre.Monnin@malix.univ-paris1.fr

Nicolas Delaforge
INRIA (Wimmics)
INRIA Sophia Antipolis Méditerranée
2004, route des Lucioles - BP 93
06902 Sophia Antipolis Cedex
Telephone number, incl. country code
Nicolas.Delaforge@inria.fr

Fabien Gandon
INRIA (Wimmics)
INRIA Sophia Antipolis Méditerranée
2004, route des Lucioles - BP 93
06902 Sophia Antipolis Cedex
Telephone number, incl. country code
Fabien.Gandon@inria.fr

ABSTRACT
The naive documentary model behind the Web (a single HTML Web page retrieved by a client from a server) soon appeared too narrow to encompass all cto account for dynamic pages, content negotiation, Web applications, etc. The Semantic Web raised another issue: how could we refer to *things outside* of the Web? Roy Fielding's REST style of architecture solved both problems by providing the Web its post-hoc "theory", making it a resource-oriented application. Recent evolutions (AJAX, HTML5, Linked Data, etc.) and envisioned evolutions (Web of devices, ubiquitous Web, etc.) require a new take on this style of architecture. At the core of the Web architecture and acting as a unifying concept beneath all its facets we find the notion of resource. The introduction of resources was very much needed for the Web to remain coherent; we now have to thoroughly redefine them to espouse its evolutions through time and usages. From the definition and the characterization of resources depends our abilities to efficiently leverage them: identify, publish, find, filter, combine, customize them, augment their affordance, etc.

Keywords
Web, resources, architecture of the Web, webarch, hypertext, hyperprocess, documentary resource, computational resource, rules.

1. INTRODUCTION
More and more often, the Web stands between us and the world. The Web of documents and data augments our perceptions of reality; the Web of applications and services, our grip on reality through the tasks we accomplish. It becomes at the same time both unavoidable in our daily activities and hardly manageable. On the Web, a resource is said to be anything and as the Web grows, everything around us is becoming a Web resource indeed.

This issue was already prevalent with the so-called Web of document. The first naive model behind the Web (a single HTML Web document retrieved by a client from a server) soon appeared too narrow to encompass all existing cases: dynamic pages, applets and scripts, content negotiation, Web applications, etc. The computational aspect, which at first appeared as an exception to the metaphor of the Web-as-a-universal-library, became the rule. In addition, the Semantic Web itself raised another issue: how could one refer to *things outside* of the Web? Roy Fielding's REST style of architecture solved both problems by providing the Web its post-hoc "theory", making it a resource-oriented application.

Recent evolutions (AJAX, HTML5, Linked Data, etc.) or even envisioned evolutions (Web of devices, ubiquitous Web, etc.) require a new take on this style of architecture. At the core of the Web architecture, acting as a unifying concept beneath all its various facets we find the notion of a resource. The introduction of resources was very much needed for the Web to remain coherent; we now have to thoroughly redefine them to espouse its evolutions through time and usages. From the definition and the characterization of resources depends our abilities to efficiently leverage them: identify, publish, find, filter, combine, customize them, augment their affordance, etc.

Justin Erenkrantz' definition of a resource as "a locus of computation" in his work on CREST (an computational update of REST) and the implications of plastering these loci all over the world around us will constitute our starting point in this article.

It also seems that among the different elements of the Web, the Web of linked data (i.e., linked meta-data or structured data) is to play an important role here. To manage the diversity of resources we can rely on another kind of diversity: the diversity of metadata. We believe that by overlaying a Web of semantic descriptions over the landscape of resources and by managing these linked data by the semantics of their linked schemas, the Web is giving itself a distributed and extensible paradigm to model its open pool of resources and to process these models. For this reason, we will lay the theoretical foundations of our work in this paper with the hope of getting closer to producing an ontology of resources based on Semantic Web formalisms in order to address many issues that are generally considered solely with regards to URIs.

2. WEB RESOURCES: TURNING THE PAGE OF THE DOCUMENTARY WEB

2.1 Giving names on the Web
In this first part, we wish to demonstrate that it is possible to account for the putative transition between a Web of document towards a Web of applications strictly *from an architectural point*

of view. Far from being just an historical account of the development of the interactive Web, with careful analysis of the introduction of JavaScript, the DOM, Ajax-based applications, etc., our endeavor will rather be one that aims to show that the basic concepts behind the Web and the constraints they entail were enough to undergo and even foster these evolutions.

At the heart of the original architecture of the Web [4] we find three basic concepts.

The first basic concept is the **URL** [6] or **URI** [7] [18]. Over time, the URI (Universal Resource Identifier) came to be thought of as a format of unique identifiers for naming and indicating any "resource" on the Web (this understanding of URIs stems from the REST style of architecture according to which parts of the Web were reinterpreted to cope with predicaments found in previous standards). If, in addition, such an identifier gives a path to obtain a representation of a resource, then it is also a URL (Universal Resource Locator) one of these famous Web "addresses" that everyone now knows about, even if, originally, they were not to be handled directly by users - e.g. "http://www.inria.fr/" is the URI(L) of INRIA home page. We could immediately note here that although these so-called addresses were initially not intended to be really used by humans, they are now part of our daily communications up to the point that there exists a market where they're valued and exchanged. Also, rather than addresses, which is actually a different concept, URLs should be understood as belonging to a subset of URIs, those URIs that are deferenceable. After all, URLs do not just locate representations, they retain the relation of identification between URIs and resources and add another relation, of access, to representations.

The second fundamental concept is the **HTTP protocol** [12] which allows for instance a client (e.g. a Web browser) to request a representation of the resource identified and "located" by a URL and get in return either the data of the resource representation or an error code indicating a problem, e.g. the famous 404 error indicating that the page you requested was not found. We should stress that the HTTP protocol does not only enable one to GET a representation but also to POST a new one, PUT an updated version or DELETE it.

The third fundamental concept was the **HTML** language to represent, store and communicate the representation(s) of the famous Web pages. It has ever since been complemented by other languages using an XML syntax to exchange any kind of structured data or document, one of the dialects of XML being a syntax for RDF, the linked data framework and core graph model of the Semantic Web.

All three basic concepts of the Web are especially important given that any current extension of the Web, including the **Web of data**, is fundamentally based on the first two concepts to identify the subject of data exchanged (URI) and transfer the data (HTTP). Indeed, the keystone of the architecture of the Web of data is the same as the classic Web: namely, the standard URI naming mechanism. However, unlike the documentary Web in which relationships are formed between anchors in hypertext documents, relationships in the Web of data are typed links (where types themselves are identified by URIs) between arbitrary resources (also identified by URIs). By relying on (HTTP) URIs for naming, on the HTTP protocol for data transfer, on the RDF graph model (instead of HTML) to describe and link resource, and on shared schemas, the recommendations of the Semantic Web outline an architecture for the world-wide interconnection of data sources and models.

2.2 Identity crisis

Yet, much work was needed to reach a shared agreement over the most basic building blocks of the Web. Standards for identifiers, for instance, evolved over time, from the first UDI draft [4] and URI specification [5] during the pre-W3C era (when the fundamentals of the Web were not yet clearly distinguished from their implementations) to the first standards concerning URLs and URNs (non-dereferenceable proper names), up until the latest URI RFCs. The work accomplished by Roy Fielding with the REST style of architecture [13] [14] was instrumental in reshaping the understanding both of Web identifiers and the HTTP protocol. It is also in Fielding's thesis that resources are defined for the first time. An immediate practical result consequence of REST was the fusion of what had previously been sundered between URLs and URNs back into URIs in 1997-1998 [29] [6]. French sociologist Laurent Thévenot [31] summarizes the agency of standards by explaining that they are "forms" that aim to generalize, extend, stabilize and equate a given technical reality. This is exactly what the Web achieved through REST and the recommendations it inspired.

Around the same time (19997-1998), other standards, the first explicitly dedicated to the Semantic Web, appeared. This conjunction is not really surprising considering that the Web had reached an unprecedented state of maturity. A "new" problem then seemed to arise. Formerly known as the httprange-14 [27] [19] – now issue-57 [20] – it consisted in understanding how one could distinguish between URIs that identify so-called "documents" and those that identify "things"[1]. This distinction itself was rephrased in terms of "information resources" (IR) and "non-information resources" (NIR) – with no real investigation with regards to whether or not these distinctions were tantamount to one another.

Basically, the httprange-14 may be summed up as an attempt to find the technical means to distinguish between IR and NIR by relying on the HTTP header sent by a server to a client in a typical HTTP negotiation. Actually, it is difficult to discuss the httprange-14 from a purely technical point of view since it has become marred with conflicting interpretation over time. What the httprange-14 actually says is that a 200 header will be followed by a representation, a 303 by a URI that identifies a second resource and is supposed to give access to a representation through a 200 header, and both 4XX and 5XX responses do not give access to anything.

Table 1: Summary of the Http-range14

HTTP code	Result	Indication
200 (OK)	(HTTP) representation	IR (and NIR?)
303 (See Other)	URI	Any kind of resource

[1] "Well, things and their descriptions are not the same and when people started using URIs to make assertions (using RDF, on the *Semantic* Web) they wanted to be able to say both http://cities.example.org/oaxaca has a radish festival every year on December 23rd and http://cities.example.org/metadata/oaxaca.html was written by Raphael Sabattini" [32].

4XX 5XX	Error message	Impossible to guess anything

One could infer, just by looking at the columns titles of **Table 1**, that the httprange-14 eschews in proving anything since the second column only contains HTTP-representations and URIs in the case of redirection (or error messages) while the third column, indicating what can be inferred from the previous one, is left open to interpretation. If resources are just "shadows" or "concepts" [14], then both information and non-information resource cannot be distinguished in terms of their potential accessibility: only representations being accessible by definition, not resources[2]. Hence, the first two rows of the third column will technically contain both IR and NIR.

However, this has not been the default interpretation. What the httprange-14 was supposed to provide was a clear separation between IR and NIR. The technical solution advocated failed to achieve that goal for the aforementioned reasons. Therefrom, a normative reading of the header responses was promoted instead of the more circumscribed technical solution first envisioned. Whenever a 200 header is served, says that reading, what we get is an IR. NIR are served indirectly, through a 303 header, by redirecting to an IR whose representations are then accessed by a client. The debate then focused on the relevance of this construal, mainly motivated by the need felt to determine whenever a URI identifies a document or a "thing" (our answer being that in both cases it identifies a *resource*, in accordance with the fundamentals of webarch).

Instead of just a technical relation, redirection thus became a good practice advocated in the publication NIR. While httprange-14 had completely failed as a purely technical tool, a normative reading was still possible. Many are still deterred by the difficulty of implementing redirection on a broad scale. That is why a new issue was opened by the TAG:

> At their meeting in 16th July 2007 [1] the TAG resolved to create a new issue, HttpRedirections-57, as a response to a community request [2] that we give further consideration to the use of the HTTP 303 status codes *and* other possible mechanisms of obtaining a description of a resource (typically a non-information resource) where the referenced resource is not capable of providing representations of itself.[3]

Echoing the AWWSW report, the issue really is about URI "definition", especially within RDF context:

> "When a URI appears in an RDF statement, how can the reader of that statement determine the author's intended meaning? What RDF triples characterize that meaning? Where does the meaning come from? How should the meaning be determined, particularly in the context of the HTTP protocol, for an http URI? Can we codify a suite of nose-following methods for semantic web use -- a recipe one can follow in order to obtain a canonical graph (or "definition", "description resource", "URI documentation") for a URI?"[4]

Rather than following that trail and search for additional ways of materializing the "meaning" of a URI, we would like to make sense of the existing Web by showing its fundamental coherence, accounting for both the Web of document and the Web of applications (current webarch discussions focusing more on the RDF side of things). This will require of a close examination of what is called a "resource", a task that can no longer be deferred for the purpose of reaching a solution.

3. ONLINE COMPUTATIONAL LOCI: FROM LIBRARIAN REFERENCES TO LOCUS OF COMPTATIONS

Everything was there from the start; in fact the Web was never purely documentary. At least if we are to take seriously the fundamentals of its architecture (and by doing so, lots of problem would simply not appear in the wild).

Looking at the definition of a resource, one can distinguish between three elements: a resource; the state of a resource; and the representational state. We shall examine each of these three elements in turns.

a) Resources

According to RFC 2396 [7], a resource can be anything. Roy Fielding called it a "shadow" or a "concept", thus making a strong distinction between resources and documents (even a digital one, understood, ultimately, as a binary set of 1 and 0 physically hosted somewhere). By definition, resources can never be accessed and are only manipulated through their representations (see [14], one section of paramount importance in their paper is fittingly entitled "Manipulating Shadows").

b) States of a resource

Resources have states. While resources remain the same (or at least *should*, since that is a normative statement which is contradicted on a daily basis), they also carry different results over time in terms of the representations that can be served to give information about them. One must thus distinguish between a resource and its state(s). This echoes the well-known distinction between *rules* and their *applications*. Alexandre Monnin [23] has previously suggested to understand resources as rules, thus specifying Fielding's claim that resources are concepts (it should be noted that concepts are often treated as rules in the philosophical literature). Assimilating the resource to a rule

[2] This is not always understood, as evidenced in RFC 3986 [18] where one can read "A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or *physical resource* [our emphasis]." No resource is physical *per se* yet if a resource is "*the semantics of what the author intends to identifies* [our emphasis]" [14], as defined in REST, then it can be said that what one intends to identify this way is a physical thing, though the resource itself won't. This distinction has been put forward to full scrutiny in order to better understand resources in [23] It is interesting to note that the httprange-14 was somehow theoretically fixed before the advent of the Semantic Web when as soon as the notion of resource was contrived to make sense of the Web departing from an understanding of "Web pages" as static documents. It is the focus on documents that again was the cause of the identity crisis. The fact that resources can be anything (as long as it is identified by a URI) also made it possible to build the Semantic Web on top of URIs as a mean to identify any resource (not just document) and link to it.

[3] Cf. [20].

[4] Cf. [3].

allows to better understand how and why states are produced. Basically, a resource generates states: over time (Web pages evolve, just as the result of search engine queries or application results in general) or punctually, through content negotiation (abbreviated as "conneg").

Of course, some cases seem at odd with this construal. Is Tim Berners-Lee a rule? Of course not. But a rule/resource being a means to identify Tim Berners-Lee, it will always depend on the way one individuates that "thing". It could be either "the founder of the Web", "the overall Director of the W3C" or "a man born of X and Y" (this is actually the Kripkean way of identifying people through across possible worlds despite the claim that rigid designators are adverse to definite descriptions), etc. Eventually, these are three different resources, or, in other words, three different objects, three different ways to pick-up something.

It is especially important make this distinction *since nothing warrants that a resource will adequately correspond to a "real thing" in the world simply because it has been published on the Web*; even more so since the goal of the Semantic Web is **not** to find a way out of this issue[5]. Resources need not always correspond to definite description but at least they must have enough content to specify what "an author intends to identify" [14]. This identification is thus possible by means of rules, corresponding to resources on the Web.

Even if the Semantic Web is to be conceptualized as a Web of "entities" (a characterization we borrow from the OKKAM project[6] [9], [30]), many of these entities are in fact the result of a complex publishing process that begins with people who edit Wikipedia and agree by consensus to identify something somehow. This is at least how DBpedia[7], one of the most successful applications of the Semantic Web, works.

We must accept once and for all the fundamentals of webarch. Fortunately, the architecture of the (Semantic) Web is no theory of truth. By contrast, it happens to be fuelled by a very different notion, *trust*. A paramount factor of trust is who the publisher of a resource is, whence the importance of *provenance* on the Web. All these elements, that were traditionally associated with the *epistemic* dimension of knowledge and dissociated from the *ontological* dimension, are now clearly intermeshed on the Web. For instance, as a telling fact that should not surprise us, it should be stressed that the definition of a resource given by Roy Fielding and Richard Taylor [14] doesn't shy away from mentioning the *intention* of an author – perhaps better described as one or more publishers in this context. A resource is thus always, at least partly, an intentional object, or rather what we'd call an *institutional object*, to better cope with the public nature of publication on the Web and its technical environment, both aspects corresponding to what is hereinafter referred to as the *editorial* and *computational* commitments.

c) The representational states of a resource

States remain abstract, just as resources, not accessible as such. What can be accessed is the HTTP-representation of the state of a resource. It can also be of various formats and many representations can be served for a given resource. While the latter need not be identical, they should at least be all faithful to a given resource. In other words, all of them must be *computable* as acceptable states (i.e., applications of the rule) of a resource (i.e., rule).

If my resource is "the original text of Shakespeare's MacBeth", a French translation in HTML will not do as faithful representation. This case illustrates a simple yet important fact: even the Web 1.0 was a Web of resources. Something that hasn't changed today, despite the advent of the Web of applications.

We thus adhere to Justin Erenkrantz' definition of resources as "loci of computation" as exposed in his work on the CREST style of architecture [11]. With a slight difference, since we also firmly believe that such a definition is true for the Web in general, not just the Web of applications. Erenkrantz' words fit very well within the general picture we try to draw where resources are rules when he uses the expression "network continuation" to describe them, thus underlying the dual aspect of stability and change[8] that essentially characterizes them.

4. WEB OF LINKED COMPUTATIONAL RESOURCE

It is commonly admitted to attach a version number to the Web, like 1.0, 2.0, 3.0, squared, etc., so that people may eventually come to think that there are several implementations of the Web. This is clearly misleading. Actually, we are still not using the full potential of what Tim Berners Lee had originally envisioned in the early nineties. In fact, rather than characterizing the Web,

[5] As Larry Masinter explains in a presentation entitled "Philosophy" of the Web", delivered at PhiloWeb 2012, WWW 2012 workshop in Lyon, France, http://www.slideshare.net/PhiloWeb/larry-masinter-philoweb): "Naming is printing money". One just has to remember that money can also be counterfeit, and the Semantic Web has not been designed to sort between genuine and counterfeit.

[6] http://www.okkam.org/

[7] http://dbpedia.org/About

[8] A major modification to this equilibrium would be the introduction of a new HTTP method to improve the red-write aspect of the Web, namely the PATCH one as described in a proposed standard, RFC 5789 [10]: "The PATCH method requests that a set of changes described in the request entity be applied to the resource identified by the Request-URI". Despite the lack of clear enough separation between resources and representations (for simplicity reasons and probably as an heritage of WebDAV conceptualization, though it certainly has the adverse consequence of partially excluding content negotiation. Rather than treat resources as modifiable files on a server, fitting mechanisms should be developed to apply updates on one kind of representations and spread it to others so as to preserve conneg). The idea behind this RFC and earlier proposals (including RFC 2068) is of great magnitude since it explicitly opens up the possibility that modifications applied to representations explain how resources may change over time (for instance, the URI identifying a given question about X on Stacked Overflow once it has been properly answered will thereafter identify a-question-about-X-that-has-been-answered-and-is-now-closed, prompting a very different attitude that translates in the set of potential actions made available to users by publishers (answering will no longer possible as perusing becomes the main available task, archiving will become the publisher's goal, etc.). In an interactive Web of applications prompting responses from users, such possibilities may become the norm, thus making it necessary to reassess what counts as "cool URIs".

these notations seem to betoken the (limited) grasp that we still have of it.

Up until recently, many industries were not ready to bring the Web to its full potential, nor were many computer scientists. Therefore when we talk about "documentary resources", one should really understand "the documentary application and understanding of Web resources".

Figure 1 emphasizes the differences between the first Web of documentary resources versus the current Web of computational resources. Of course, this distinction is mostly didactic. In practice, things are not so neatly and conveniently separated and, as we said, most of what we discover these days was here from the start. But for our purpose it is convenient to have a look at the "evolution of Web understanding". This approach allows us to highlight how much *practices* changed the structure of the Web.

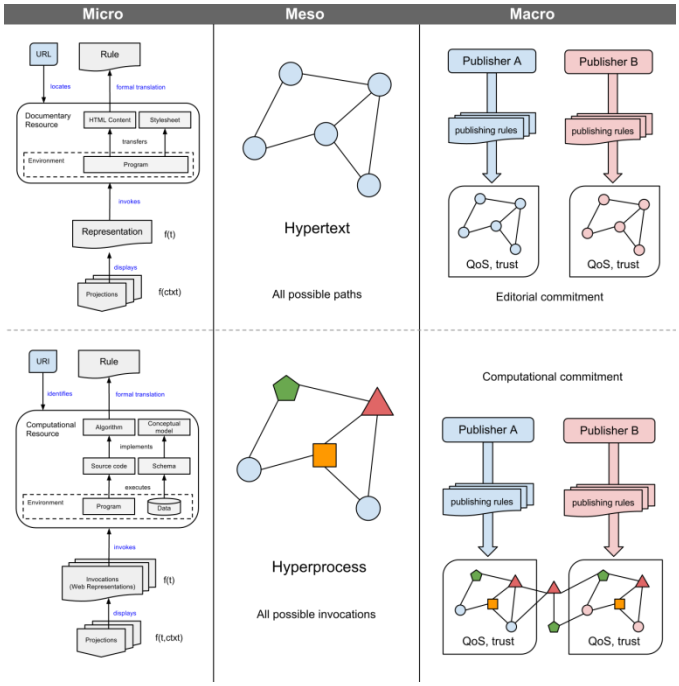

Figure 1: From Hypertext to Hyperprocess on a micro, meso and macro level.

In this section we propose a three-level analysis of the "Web of Computational Resource" (CoReWeb). The micro level focuses on the resource itself and its inner mechanisms. The meso level is about relations and interactions between computational resources. The macro level highlights the relations between the editorial policy of a publisher and the way he manages his Web resources.

4.1 Resources and other rules

Web resources are often published as part of bigger sets of resources that have in common to be named and managed by the same publisher. We consider that an *editorial policy* can be summarized as a structured rule set. Some of these rules are generic, others are specific and can inherit or be related to broader ones. From this, we assert that **any Web resource formally expresses the intersection of several of these publishing rules**. In other words, a Web resource is situated at the intersection of a number of publishing rules. A URI then gives access to a representational state that is the result of this intersection and its closure, while it is often perceived as identifying ony the most specific rule involved in generating the aforementioned representational state (otherwise known as "the" resource).

Indeed, the very way by which Web resources are cut out depends on their being distinguished from one another and included in a common set, an editorial ecosystem generally known as a "website" – even though such a notion bears little sense according to webarch. Actually, the set-theoretic approach, as found in the W3C recommendation POWDER ([1], [2]) allows to treat websites and RESTful Web services or data stores the same way: as "irisets" (in facts, sets of resources rather than IRIs, but the former are only manipulable as sets of IRIs[9] – groupings of resources identified by IRIs/URIs).

To borrow an analogy from linguistics, the "signified" in Saussure's theory is specified by relations of difference. By contrast, resources do share some common traits: they link to one another, to external resources, as mashups include parts of other resources, follow a given publishing policy being organized under specific categories, hierarchy, etc. Or, alternatively, in the case of Semantic Web resources, follow various axioms, share sets of properties and objects, etc. Yet, eventually, each must have a specific content distinguishing it from its neighbors. A resource is precisely this modicum atom of content that is supposed to remain stable, at least as much as possible, especially from a publisher's point of view, whereas representations as well as editorial policies do endure modifications (albeit allegedly much less often regarding the latter).

Here it may be useful to appeal to the distinction proposed by T.V. Raman [24], between "Web components" and "Web container":

> "(...) the need to provide a single point of access to oft-used information led to portal sites that aggregated all the information onto a single Web page. In this context, the various items of information can be viewed as lightweight Web components. The environment in which these components are hosted (such as the software that generates and manages the Web page) can be viewed as a Web container. Thus, common actions (such as signing in) were refactored to be shared among the various Web applications hosted by the Web container, a piece of software managing the user's browsing context."

Those rules reflect the editorial policy of a "website". For instance, this includes whether actions such as sharing content on a social network or using one's account to sign up or log in to a third-party website as well as being given the possibility to push the Facebook "like" button or Google's "+1" are made available. Such cases correspond to the integration of modular *components*, the grouping of which (and other editorial rules previously mentioned) gives rise to a Web *container*. Components and containers[10] may or may not be identified for themselves (a Facebook component might have one or more URIs while, by contrast, the decision to link a page to other pages "inside" a given container will not).

In any case, both containers and components are akin to *non-necessary rules* which add to a resource specific content enough

[9] See [9]: "A Resource Set is defined in terms of the IRIs of resources that are its members."

[10] We are using those words in a broader sense than as mere equivalents of "portlets" and "servlets". Many examples are given in section 4.1.2 ("Meso level").

details to compute concrete http-representations (the software used, HTML code, Web server headers and configurations, CSS style sheets, the JavaScript it includes, the JSP or PHP tags it uses, etc.). In other words, the policies or pieces of code that will generate a desired effect without belonging to the core-definition of a resource – i.e., without being confused with what a URI specifically identifies.

On the Web, attending to editorial policies and rules can either be done by one or many people. Since these tasks can be separated and often are in concrete situations, it is crucial to have them clearly distinguished from the inception.

4.1.1 Micro level

Technical evolutions have impacted both servers and clients. At the beginning, browsers were the only Web clients but now, we have many devices and applications that are able to connect to the Web and to get data and services from it.

Web servers were originally designed to propose a hypertext experience of "filesystem-like" remote services. Since the common gateway interface (CGI) their structure became increasingly complex. Nowadays, servers are able to negotiate with clients to adjust the response so that most of the content is generated on the fly. Any Web server is also compatible with at least one programing language that can trigger the processing of very sophisticated tasks that sometimes involve other remote services.

This point has important consequences on what is downloaded from those servers. One of the defined rationales behind documentary resources is that people have tried to preserve the *causal pathway between a reference and an informational content*, because it was constitutive of all our "real world" documentary reference systems. The "transition" from documentary resource to computational resource made more obvious that this artificially preserved causal relation had been broken. Now the downloaded content is what [17] called a "Web representation" of the resource, and can change each time a resource is invoked. The documentary location has been replaced by a *locus of computation*, or what we would call *a space of invocations***.**

Times at which "pages" were written with authoring tools like Adobe Dreamweaver or Microsoft Word now seem long gone. Today, blogs and wikis have permeated the Web and old-fashioned authoring practices are withering. From the server point of view, it is much more complicated to host a blog than a set of HTML files and CSS style sheets. Online editing tools involve scripting language capabilities, database and adequate security policies with possibly multiple ports opened to connect remote services, authentication API keys, etc.

To enable the Web of Data, the W3C have made slight changes in the specifications of Web architecture. URLs are now considered as (dereferenceable) URIs. From a linked data perspective, every URI minter/resource publisher is indeed strongly encouraged to make them dereferencable, so that it is possible to navigate between RDF concepts in the same manner as between pages. The 303 HTTP code is used to inform the "concept browser" that the resource he is asking for is not "informational". Hence, the technical distinction between Web pages, Web services and RDF concepts is no longer a valid one (our subsequent use of the received expression "Web pages" – or rather HTTP-representations – is entirely motivated by this observation).

URLs were initially locating documentary resources. CGI and REST have turn URLs into RPC passing parameters to scripts or web services. Now every URL is, and in a sense has always been, a URI. URI are identifying protean resources that can turn themselves in any format required by the client. Such are the *computational resources*.

Like with any program, to manipulate a computational resource, one has to *implement* an *algorithm* with a *programing language*, a *conceptual model* and *data*. Each of these parts has a strong impact on those Web representations a user can browse or a program parse.

As said before, a resource is a formal translation of necessary and non-necessary publishing rules but these rules themselves can change, the implementation can evolve to match a new technological context, a bug can be fixed, a new feature added, the database can also be updated with fresh data, etc. There are many reasons for Web representations to change and that is the real communicative power of the Web: an editor can instantly adapt the whole editorial chain synchronously in accordance with any informational or technological constraints.

The growth of Web communication in the last fifteen years resides mostly in the quickness with which information can pass from the state of data stored in one or more remote databases to a Web representation. Thus, the ease of update of the publication chain on a global scale induced by the architecture of the Web constitutes its greatest value and its biggest breach with previous editorial practices.

4.1.2 Meso level

As we have seen, through HTTP, any computational resource is likely to refer to other resources or communicate with them. This capability was exploited to add dynamicity and real-time content to Web pages, but it also has many applications in the Web of data.

4.1.2.1 Extending the pages communication capabilities

In 1995, Java applets were the first practical manner to asynchronously load remote content into Web pages. One year later, Microsoft introduced the *iframe* element designed to allow webmasters to include one Web page into another one. In 1999, the first XMLHTTP ActiveX control appeared with IE5. Now every browser proposes asynchronous communication capabilities and this technology, commonly known as AJAX for "Asynchronous JavaScript and XML", is very widely used.

Many widgets do use AJAX to connect a remote Web server and include real-time changes into the displayed content of a Web page. Real-time charts of stock exchange ratings, news tickers, Google maps, Google trends are just a few examples of applications using AJAX.

But with HTML5 and the brand new Websocket JavaScript API, things are going even further. Whereas AJAX is asynchronous (connections are closed after the server response is received), websockets provide persistent connection capabilities to Web pages, a feature that used to be characteristic of low level programming languages. Other evolutions like IndexedDB and WebGL APIs contribute even more to transform Web Pages into complex Web Applications [23]. Persistent connections enable the development of real-time applications, such as collaborative real-time painting or 3D games.

4.1.2.2 Public APIs, Dashboards, Widgets, Mashups

With the spread of Service-Oriented Architectures and the standardization of RPC (Remote Procedure Call) protocols, the

Web offers a wide pool of public services any Web developer can draw from to build innovative applications. These services can either be requested directly, or, more often, they provide widgets that should be integrated in Web pages.

Since 2005, many dashboard applications have emerged, like Netvibes[11], but quickly Google[12], Yahoo![13] and Microsoft[14] released their own dashboards providing a large variety of widgets like calendars, mail, contacts, todo lists, RSS readers, financial or weather survey tools.

Entire frameworks, like Life Ray[15] have been developed to build such platforms where the user can compose his own page made of portlets[16].

Now dashboard applications seem to wither in favor of more flexible widgets that can integrate into any page. It is impossible to reference them all here, so we will limit ourselves to some typical examples:

- *Data visualization*

Using either REST or SOAP protocols, it is now common to compose complex processing chains made of multiple remote service calls. The most typical combination is to provide a data stream to a visualization service and to integrate it into a Web page. As an example, one can mention Wordle[17], Many Eyes[18] or Google Maps[19].

- *Mashups*

A mashup is the result of the combination of several sources of information like RSS feeds. Yahoo! Pipes[20] is the best-known mashup application and his cousin, DERI pipes[21], includes semantic features. Other examples include 123People[22], a personal information aggregator and the Twitter API which gave birth to lots of applications like Bubble-T[23], Polemic Tweet[24]...

- *URL shortener*

With the Twitter's 140 characters restriction, URLs were often too long to be posted. To that purpose shortening services have appeared like TinyURL[25] and Bit.Ly[26]. Both provide a public API to get a short URL from a longer one. These very simple services are among the most used on the Web and within many Twitter clients.

- *Translation services*

For those who wish to get their Web page automatically translated in any language, Microsoft, Yahoo! and Google have published their solutions: Bing Translation API[27], Google Translation[28] and Yahoo! Babelfish translation service[29]. The final representation visualized by the user is thus the application of their web service to an initial Web representation that could itself call on many other Web resources.

- *Currency conversion*

It can be useful to delegate to a remote service the task of currency conversion according to current exchange rates. This is the purpose of web services like Exchange Rate API[30] or Open source exchange rates[31].

4.1.2.3 Web services orchestration and choreography

For people wishing to build much more complex services compositions from middleware architectures, to model Business processes as compositions of atomic tasks and to execute these compositions as single processes, several standards have been released by the W3C allowing what is called "service orchestration" and "service choreography"[32] [21] [28]:

> An orchestration specifies an executable process that involves message exchanges with other systems, such that the message exchange sequences are controlled by the orchestration designer. A choreography specifies a protocol for peer-to-peer interactions, defining, e.g., the legal sequences of messages exchanged with the purpose of guaranteeing interoperability. Such a protocol is not directly executable, as it allows many different realizations (processes that comply with it).[33]

Therefore resources are not only related to each other by navigation or composition links. They are nested into a much more complex interaction network mostly based on remote procedure calls and data exchange between servers. Consequently, qualifying the Web as a hypertext seems a little bit outdated. That is why we would prefer the term *hyperprocess* (actually, REST, by turning webarch into a resource-oriented architecture already

---

[11] http://www.netvibes.com/

[12] http://www.google.com/

[13] http://my.yahoo.com/

[14] http://live.com

[15] http://www.liferay.com/

[16] http://www.jcp.org/en/jsr/detail?id=286

[17] http://www.wordle.net/

[18] http://www-958.ibm.com/software/data/cognos/manyeyes/

[19] https://maps.google.com/

[20] http://pipes.yahoo.com/pipes/

[21] http://pipes.deri.org/

[22] http://www.123people.com/

[23] http://dev.fabelier.org/bubble-t/

[24] http://polemictweet.com/

[25] http://tinyurl.com/

[26] https://bitly.com/

[27] http://msdn.microsoft.com/en-us/library/ff512419.aspx

[28] https://developers.google.com/translate/v2/getting_started

[29] http://babelfish.yahoo.com/free_trans_service

[30] http://www.exchangerate-api.com/howto

[31] http://josscrowcroft.github.com/open-exchange-rates/

[32] In [15], we find an attempt to account for the client-server dialog mechanism both in the context of Web pages and Web services in a logical way in order to type the processes involved; in other words, so as to be able to determine whether "two processes that interact may be checked *before* the interaction". The Curry-Howard correspondence ensures that these logical types correspond to Web processes. While especially relevant to our own computational approach, by treating all URIs as URLs, and URLs as pointers in computing languages, it has the severe drawback of being oblivious to the fact that the Web is a publishing platform whom identifiers have two functions, none of which can be ignored.

[33] Source : http://en.wikipedia.org/wiki/Business_Process_Execution_Language#The_BPEL_language

had the immediate effect of discarding this notion of hypertext making it fully inappropriate for the Web; despite the enduring popularity of the word, it remains largely deprived of meaning in this context).

4.1.3 Macro level

On the one hand, and fortunately for Web users, the increasing complexity of server infrastructures, was progressively outsourced under the responsibility of specialized companies that provide hosting and administration services at low cost. The improvement of virtualization and monitoring technologies has also greatly simplified such system administration tasks.

On the other hand, it is more and more difficult for publishers to ensure a good *quality of service* throughout the entire processing chain. The technological stack and the processes involved in publishing a resource have become so complex and so distributed that it is becoming harder and harder to ensure a strict editorial commitment because as the Web grows in diversity, this commitment has turned into a computational one.

From the societal point of view, content publishers whose main activity was to produce content and to guarantee the quality of information now have to deal with various new constraints owing to the specificity of the medium. Beyond the increasing rate of publication, publishers must also face new stringent public expectations in terms of technical quality of service and interoperability.

Facebook, Twitter, Delicious and Google have imposed their "social ranking" tools ("I like" button, Google "+1", "Retweet") to publishers who must embrace these technologies otherwise they risk losing customers. Publishers must also consider the growing number of devices that people use to access information: smartphones, tablets, Kindle, television... The outsourcing of network infrastructures and servers adds another intermediate in the decision chain, which further complicates delivering a good quality of service. Browsers now even include calls to the cloud to delegate part of the rendering...

In summary, the gradual evolution from *hypertext* to *hyperprocess* has progressively added to the constraints of an *editorial commitment* those of a *computational commitment*.

5. CONCLUSION – TOWARD UBIQUITOUS HYPER-RESOURCES

The Web was already very rich with regards to the variety of the multimedia resources it hosted and linked to, and this richness is still increasing. With the advent of the mobile Web and the Internet of Things, we are going toward Web-augmented reality, ubiquitous Web and a Web of things or objects.

But while the Web is augmenting our reality, the objects and places of our lives, the latter are in turn increasing the number and variety of Web resources. This evolution will come with a price, namely an increase in the complexity of Web resources and their dependencies.

The architecture of the Web of data and the models of the Semantic Web may provide a way to match the diversity of online resources by means of a framework of metadata designed to annotate Web resources and exploit the semantics of their schemas to process them intelligently. Metadata and their schemas could be the keystone of the new resource-centric Web applications, their integration and interoperability.

It is conceivable that tomorrow, he who controls metadata on the Web controls Web resources, and through them a lot of things.

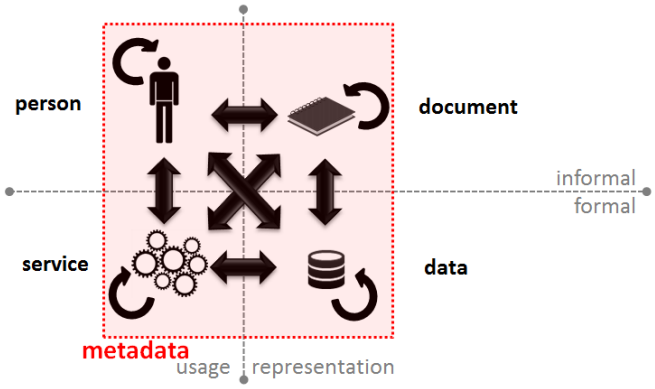

Figure 2: Synthetic view of the resource-centric Web architecture and the cross-cutting importance of metadata (as found in [16]).

6. REFERENCES

[1] Archer, P., Smith, K., and Perego, A. (Eds.). Protocol for Web Description Resources (POWDER): Description Resources. W3C (Sept. 1 2009). Retrieved from http://www.w3.org/TR/powder-dr/#operational

[2] Archer, P., Perego, A., and Smith, K. (Eds.). Protocol for Web Description Resources (POWDER): Grouping of Resources. W3C (Sept. 1 2009). Retrieved from http://www.w3.org/TR/powder-grouping/

[3] "Architecture of the World Wide Semantic Web" Task Group. *AwwswFinalReport*, W3C Wiki, (2012). Retrieved from http://www.w3.org/wiki/AwwswFinalReport

[4] Berners-Lee, T., Groff, J. F., and Cailliau, R. 1992. Universal Document Identifiers on the Network. *CERN, February*.

[5] Berners-Lee, T. 1994. RFC 1630 - Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web (June 1994). Retrieved from http://tools.ietf.org/html/rfc1630

[6] Berners-Lee, T., Masinter, L., and McCahill, M. 1994. RFC 1738 - Uniform Resource Locators (URL). (Dec. 1994). Retrieved from http://tools.ietf.org/html/rfc1738

[7] Berners-Lee, T., Fielding, R. T., Irvine, U. C., & Masinter, L. (1998, August). RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax. Retrieved from http://tools.ietf.org/html/rfc2396

[8] Berners-Lee, T., Cailliau, R., and Connolly, D. 2000. A Little History of the World Wide Web. *w3.org*. Retrieved from http://www.w3.org/History.html

[9] Bouquet, P., Stoermer, H., and Vignolo, M. 2011. Web of Data and Web of Entities: Identity and Reference in Interlinked Data in the Semantic Web. *Philosophy & Technology*. DOI:10.1007/s13347-010-0011-6

[10] Dusseault, L., & Snell, J. 2010. RFC 5789. PATCH Method for HTTP. IETF (March 2010). Retrieved from http://tools.ietf.org/html/rfc5789

[11] Erenkrantz, J. R. 2009. *Computational REST: A New Model for Decentralized, Internet-Scale Applications* (PhD Thesis). University of California Irvine. Retrieved from file:///F:/Articles/Erenkrantz-2009-CREST-Dissertation.pdf

[12] Fielding, R. T., Gettys, J., Mogul, J. C., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (Eds.). 1999. RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1 (June 1999). Retrieved from http://tools.ietf.org/rfc/rfc2616.txt

[13] Fielding, R. T. 2000. *Architectural Styles and the Design of Network-based Software Architectures* (PhD Thesis). University of California, Irvine. Retrieved from http://www.ics.uci.edu/%7Efielding/pubs/dissertation/fielding_dissertation.pdf

[14] Fielding, R. T., and Taylor, R. N. 2002. Principled design of the modern Web architecture. *ACM Transactions on Internet Technology (TOIT)*, *2*(2), 115–150.

[15] Fouqueré, C. 2011. Ludics and web: another reading of standard operations. In A. Lecomte and S. Tronçon (Eds.), *Ludics, Dialogue and Interaction*, Lecture Notes in Computer Science (Vol. 6505, pp. 58–77). Springer.

[16] Gandon, F., Faron-Zucker, C., and Corby, O. 2012. *Le web sémantique - Comment lier les données et les schémas sur le web ?* Dunod.

[17] Halpin, H., and Presutti, V. 2011. The identity of resources on the Web: An ontology for Web architecture. *Applied Ontology*, 6(3), 263–293.

[18] Hinden, R., Carpenter, B., and Masinter, L. 1999. RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax. (1999, December). Retrieved from http://tools.ietf.org/html/rfc3986

[19] ISSUE-14: What is the range of the HTTP dereference function? - Technical Architecture Group Tracker. (n.d.). Retrieved from http://www.w3.org/2001/tag/group/track/issues/14

[20] ISSUE-57: Mechanisms for obtaining information about the meaning of a given URI - HttpRedirections-57 - Mechanisms for obtaining information about the meaning of a given URI. (n.d.). Technical Architecture Group Tracker. Retrieved from http://www.w3.org/2001/tag/group/track/issues/57

[21] Nickolas Kavantzas, David Burdett, Gregory Ritzinger, Tony Fletcher, Yves Lafon, and Charlton Barreto (Eds.). 2005. Web Services Choreography Description Language Version 1.0. W3C Candidate Recommendation 9 November 2005. W3C. Retrieved from http://www.w3.org/TR/ws-cdl-10/

[22] Monnin, A. 2012. L'ingénierie philosophique comme design ontologique : retour sur l'émergence de la « ressource ». *Réel-Virtuel*, *4*. Retrieved from http://reelvirtuel.univ-paris1.fr/index.php?/revue-en-ligne/3-monnin/

[23] Monnin, A. 2012. The artifactualization of reference and "substances" on the Web. Why (HTTP) URIs do not (always) refer nor resources hold by themselves. *American Philosophical Association Newsletter*, *11*.

[24] Raman, T. V. 2009. Toward 2 W, beyond web 2.0. *Communications of the ACM*, *52*(2), 52–59.

[25] Raman, T. V., & Malhotra, A. (Eds.). 2011. Identifying Application State - TAG Finding 01 December 2011. W3C. Retrieved from http://www.w3.org/2001/tag/doc/IdentifyingApplicationState

[26] Rees, J. 2012. The TAG Member's Guide to ISSUE-57 Discussion. *w3.org* (2012, March 27).. Retrieved, from http://www.w3.org/2001/tag/2012/04/57guide.html

[27] Rhys Lewis (Ed.). 2007. Dereferencing HTTP URIs (Draft Tag Finding 31 May 2007). W3C (May 31 2007). Retrieved from http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14

[28] Ross-Talbot, S., and Fletcher, T. (Eds.). 2006. Web Services Choreography Description Language: Primer. W3C Working Draft 19 June 2006. W3C (June 19 2006). Retrieved from http://www.w3.org/TR/2006/WD-ws-cdl-10-primer-20060619/

[29] Sollins, K., & Masinter, L. 1994. RFC 1737 - Functional Requirements for Uniform Resource Names (Dec. 1994). Retrieved from http://tools.ietf.org/html/rfc1737

[30] Stoermer, H. 2008. *OKKAM: Enabling Entity-centric Information Integration in the Semantic Web* (Thesis). DIT - University of Trento (Jan. 1 2008). Retrieved from http://eprints.biblio.unitn.it/archive/00001394/

[31] Thévenot, L. 1986. Les Investissements de Forme. In L. Thévenot (Ed.), *Conventions économiques*, Cahiers de Centre d'Etude de l'Emploi (pp. 21–71). Paris, PUF.

[32] Thompson, H. S. 2012. An introduction to naming and reference on the Web. HTML presented at the Philosophy of the Web seminar, La Sorbonne, Paris. (Jan. 28 2012). Retrieved from http://www.ltg.ed.ac.uk/~ht/PhilWeb_2012/