Procedural versus Declarative Semantics on the Web: Microformats, GRDDL, and RDFa

Harry Halpin

University of Edinburgh

Abstract

The alignment of the hypertext Web with the emerging Semantic Web should be a critical issue for both the further evolution of the Web, as the Semantic Web relies on a critical mass of deployed data in order to reach its full potential. With the rise of microformats, a compelling “lower-case” semantic web has been arising organically. Recently, a large number of emerging standards from the Semantic Web community, in particular GRDDL (Gleaning Resource Description from Dialects of Language) and RDFa have emerged as different methodologies for bridging the gap between microformats, hypertext, and the Semantic Web. GRDDL solves this problem by allowing an XML document link to a procedure to that transforms the document to RDF. In contrast, RDFa solves the problem by allowing document authors to declaratively embed RDF in hypertext documents. While these technologies are movement to alignment between the Hypertext Web and the Semantic Web, they revive a debate in artificial intelligence over the relative merits “procedural” versus “declarative” semantics, a debate that leads to fundamental issues at the heart of the Semantic Web.

Key words: procedural semantics, declarative semantics, GRDDL, RDFa, microformats, Web 2.0

Email address: H.Halpin@ed.ac.uk (Harry Halpin).
URL: http://www.ibiblio.org/hhalpin (Harry Halpin).

Preprint submitted to Elsevier 2 July 2007
1 Introduction: Universal Information Space

The founding specification of both the hypertext Web and the Semantic Web is the URI, the “Uniform Resource Identifier” (2). Tellingly, the specification had as its original title “Universal Resource Identifier,” as the purpose of the Web was not just a hypertext system but a “universal information space.” In this regard, the Semantic Web is the natural extension of the original Hypertext Web, as it further allows the Web as a totalizing system to incorporate any sort of information, not just hypertext documents.

Metcalfe’s Law states that value of a network is proportional to the square of the number of its users, and while minor variations of it with higher or lower growth are still under debate, the general rule has been shown to hold. As put by Jim Hendler, a variation of Metcalfe’s Law for the Semantic Web would state that value of the Semantic Web in terms of applications goes up exponentially given the amount of data on the Semantic Web (7). So far, the Semantic Web has pursued a line of “separate but equal” evolution, hoping that authors and applications would natively produce RDF, and so ignoring the already massively deployed Hypertext Web. Since the current Web of at least 12 billion documents currently dwarfs the Semantic Web of 2 million documents, this strategy needs to be revised (4).

2 Web 2.0: Universal Computation Space

In the mean time, a phenomenon entitled “Web 2.0” arose, characterized by the use of the Web as a platform for applications and resurgence in user-created content on the Web. While disdain exists in Semantic Web circles for the “Web 2.0” epithet, it is a valid phenomenon: as opposed to the Web as a universal information space, it is the the use of the Web as a platform for applications, and so a universal computation space. Web-based applications are the natural consequent of the major browsing vendors have finally implemented a single universal programming language correctly (Javascript) whose use of the DOM (Document Object Model, an abstract methodology for accessing XML as a tree-structure) lets these applications dynamically update not just the entire “tree” of the the DOM, but the “branches” of the DOM via XMLHttpRequest. This providing the latency needed for cross-platform applications, not just “applets,” to be run off the Web. This fueled the creation of easy-to-use web-applications for users to author their own content, which in turn led to the social explosion of user-created content that has led to the growth of the Web 2.0, a growth lacking in the Semantic Web. This feedback cycle between web-based applications and user-generated content is not incompatible with the Semantic Web.
What is the “breaking point” for the full realization of the Web as a platform for applications? The breaking point is the lack of a universal data format. The reliance on the XML and JSON (Javascript Object Notation) is problematic as these data formats are not perfect for “mash-ups,” which are combinations of data from heterogeneous data sources of Web. One solution is to use RDF as a data format, because RDF’s use of URIs as foreign keys and whose flexible “link and node” data-model make it perfect for mash-ups. However, most data is trapped in idiosyncratic hand-coded HTML and in HTML produced on demand from databases behind firewalls. Although “liberating the data” makes for good sloganeering, it is unclear how what the next step is. There are two opposing methodologies for extracting data “semantics” from the Web. The first is that authors should make the semantics of their data explicit. This viewpoint characterizes Semantic Web standards that lets one make semantics explicit through the use of schemas, ontologies, and rules. The opposing viewpoint is that applications should scrape the implicit data. The largest proponent of this methodology has been companies such as Google that specialize in machine-learning and natural-language processing techniques that discover the latent semantics of existing documents. As Peter Norvig of Google put to Berners-Lee at AAAI 2006, “We deal with millions of Web masters who can’t configure a server, can’t write HTML. It’s hard for them to go to the next step.” Or is it?

3 Microformats: The Lower-Case Semantic Web

*Microformats* easily embed ‘semantics’ in web-pages using existing tags from HTML, re-discovering the *rel* link in HTML and using *span* and *div* classes to embed semantics, as shown in Figure 2 (11). While sometimes advertised as the “lower-case” alternative to the Semantic Web, microformats are actually proof that content authors can and will add explicit semantics to their data if the learning-curve and ease of deployment is low. Microformats are nearly isomorphic to already widely adopted non-Web standards, exemplified by the mapping of the vCard standard to the hCard microformat or the mapping of the iCal standard to the hCard microformat. As given by the centralized repository at [http://www.microformats.org](http://www.microformats.org), microformats range over social networking, licenses such as Creative Commons, and collaborative tagging. From “Web 2.0” sites like Upcoming.org, Flickr, Yedda, LinkedIn, and Eventful to more established corporations such as Yahoo!, many sites are deploying microformats and a number of plug-ins exist to help authors and browsers extract standard data format from a microformat, allowing one to interoperate existing programs with microformat data.

Microformats put semantic data into HTML, but provide no standard way to get the data out. Since microformat do not have a uniform data model,
each microformat has its own extraction algorithm. Microformats cannot be validated easily and there are no standardized extraction algorithms for combining multiple microformats in the same document (10). Microformats are domain-specific: a user can not make a microformat for just anything, but must use a few domains that are well-known, and while the process for creating new microformats is open and informal, many users will not want to engage the process if they do not feel their microformat is of universal usage to the Web community. A chemist could not easily use the microformat approach to mark-up a document with for chemical compounds, since no existing microformat for chemicals exist and the process may not find such a microformat to be for widespread use on the Web. Almost all the failings of microformats are taken into account by the Semantic Web. First, the Semantic Web provides a common-data format for creating self-describing vocabularies in any domain using a methodology that does not require any centralized process. Furthermore, Semantic Web data easily recombines or “mashes-up,” solving a major hurdle for users of microformats, and unlike most “mash-ups” due to their use of URIs, the Semantic Web naturally scales to the Web itself.

Despite these advantages, the rate of growth of microformats is large enough that the Semantic Web ignore microformats at their own peril: the current size of the Semantic Web is estimated to be at 2 billion documents (4), while the estimated size of the “microformat” web is estimated to be around 500 million.\(^1\) Although the Semantic Web is larger, it has also been around in some form for nearly a decade, while microformats have only been deployed for only two years. If one wants for the “network effect” of Metcalfe’s Law to apply to the Semantic Web, the Semantic Web would do well to incorporate microformat and further align itself with massively deployed the Hypertext Web.

4 GRDDL: from Hypertext to RDF

The Semantic Web needs to learn from microformats that users will explicitly mark up their data if they are given an straightforward and low-cost methodology for doing so. GRDDL (Gleaning Resource Descriptions from Dialects of Languages), a W3C Recommendation, is a simple way to bootstrap RDF out of XML and in particular XHTML data by explicitly labelling procedures, usually XSL transforms, from RDF to XML (3). It allows microformats to be considered a syntactic short-cut for writing full-fledged RDF. GRDDL consists of three components (3):

(1) **GRDDL-aware agent**: a GRDDL-aware agent is a software agent able

\(^1\) Estimate from Alexa as of July 2007.
to identify the GRDDL transformations and run them to extract RDF.

(2) **Source Document:** an XML document which references at least one GRDDL transformation for a GRDDL-aware agent to use to extract RDF from it.

(3) **GRDDL Transformation:** a GRDDL transformation is an algorithm which, when applied to a compliant source document, allows a GRDDL-aware agent to extract RDF from this document.

GRDDL is a methodology for applying a function to an XPath Data model that results in a RDF graph. GRDDL itself is described formally by a series of rules written as SPARQL fragments which are summarized below:\(^1\):

(1) Given an XPath root node \( N \) with root element \( E \), if there exists a local name attribute with the name 'transformation' and the namespace URI of GRDDL,\(^2\) then the resolution of the GRDDL value is a GRDDL transformation \( TP \). This rule can be applied multiple times, leading to many GRDDL transformations for a single document.

(2) If an information resource \( IR \) is represented by a XML document, the application to GRDDL transformation \( TP \) applied to \( IR \) gives a RDF graph \( G \), then \( G \) is a GRDDL result of \( IR \).

(3) If \( F \) and \( G \) are GRDDL results of \( IR \), then the graph merge of \( F \) and \( G \) is also a GRDDL result of \( IR \).

One drawback of GRDDL as given above is that it forces every user to explicitly mark-up their documents with a GRDDL transformation, so authors of documents must both understand GRDDL and, if adding GRDDL manually, must be able to edit the head of their documents. Since many authors may not have the ability or time to add GRDDL transformations explicitly, it makes far more sense to have a GRDDL transformation be implicit for the vocabulary (dialect). Instead of putting the burden on end-users, GRDDL permits XML vocabulary authors create a GRDDL transformation for each vocabulary and then put it in their namespace document. Due to the “self-describing” nature of XML documents, GRDDL-aware clients can “follow their nose” from the root element of a document to a namespace document in order to discover a GRDDL transformation. Furthermore, this process may be repeated recursively, although GRDDL forbids circular loops. The following rule describes the principle (3):

(1) If there is an namespace document \( NS \) which has a GRDDL result that includes a triple whose subject is the namespace document, whose predicate is the GRDDL namespace transformation predicate,\(^3\) and whose object is the URI of the transformation \( TX \), then an information resource \( IR \) has an XML representation with root node \( N \) and with a root

\(^2\) http://www.w3.org/2003/g/dataview#'

\(^3\) http://www.w3.org/2003/g/data-view#namespaceTransformation
element with a namespace name $NS$ has the GRDDL transformation $TX$ of $N$.

To ease making XML vocabularies GRDDL-enabled, with the above rule GRDDL can be deployed with a host of different namespace documents, ranging from RDF Schema to XML Schema, as long as the required RDF triple demarcating a GRDDL transformation can be embedded in the document, either explicitly using schema annotations or RDF Schema, or implicitly by GRDDL-enabling a XHTML namespace document. Since GRDDL can be deployed in XHTML (which always has the namespace of html), document authors can instead use “profiles,” a URI-based mechanism to extend the semantics of XHTML, as given in Figure 2 to link XHTML to GRDDL documents.

For each popular microformat vocabulary, the Semantic Web community is in the midst of manufacturing an isomorphic RDF model. XFN (“XHTML Friends Network,” a social networking microformat) can currently map to a mixture of VCard RDF and FOAF via GRDDL. Likewise, hCard can map to VCard/RDF, rel-license can map to the RDF vocabulary for Creative Commons, and hCalendar can map to RDF Calendar.

5 RDFa: RDF in Hypertext

Although work in RDFa historically precedes the development of microformats, RDFa directly confronts microformats by creating a microformat-like syntax for directly embedding RDF in HTML(1). The issue of how to add RDF to HTML has been a subject of debate since the invention of RDF, and the original methodology of simply cutting and pasting raw RDF/XML into HTML leads to both validity problems and is unreadable by humans. RDFa creates a simple human-readable syntax for embedding RDF in XHTML while maintaining validity, and may be part of upcoming versions of HTML.

RDFa elegantly solves the embedding of decentralized vocabularies in hyper-
text by allowing document author to place full URIs in semantic markup using already existing hypertext tags. RDFa enables multiple full-fledged namespaces unlike the single ‘flat’ name space of microformats. RDFa should be relatively easy for many documents creators to deploy, as it uses a number of analogies and extensions to XHTML to place RDFa content in links, as well as span and div classes, in a similar fashion to microformats. The use of class, about and rel attributes on arbitrary HTML tags make RDFa more flexible than microformats (although also arguably more confusing for users), since the same RDF graph can be embedded in HTML in multiple ways. An equivalent example to our example of microformats and GRDDL (3) is shown using RDFa in Figure (1).

RDFa obeys the informal maxim that allows an agent to simply “view source” to discover RDFa without any additional processing. Unlike GRDDL, RDFa lets particular RDF triples to transparently attach to nodes within HTML, and so giving RDFa the advantage that it can be “cut and pasted” with semantics preserved. Furthermore, RDFa is a purely declarative, and so unlike GRDDL it does not rely on a Turing-complete ’black box’ to convert HTML to RDF (8).

Should one use GRDDL or RDFa? For XML documents outside of XHTML currently the only option is GRDDL, as RDFa only applies as a profile of XHTML (and perhaps future versions of HTML), not as a general purpose notation for inserting RDF into XML. Regards hypertext, if one wishes for a XML vocabulary to automatically deploy as RDF with no work for on the authors of individual documents, then one should employ GRDDL. This makes GRDDL an easy upgrade tool for existing XML vocabularies and uses of microformats. If one wants to add RDF to individual XHTML documents, then one should use RDFa. Since RDFa is a XHTML dialect, a GRDDL transformation from RDFa to RDF/XML also exists. Lastly, there is ongoing work to enable cross-platform integration of RDFa and microformats by allowing a GRDDL transformation to convert existing microformat data into RDFa, keeping the HTML in line. However, the debate between RDFa and GRDDL cuts much deeper.
The two approaches of embedding RDF into hypertext mirrors the seminal debate between “procedural and declarative semantics” in artificial intelligence. The debate focused on what was the most appropriate manner for AI to model human intelligence, either as a formalism based on logic use inference to simulate intelligence, or through creating computer programs which model the actions of human intelligence given some “input” and “output.” This is analogous to RDFa, which declaratively inserts RDF in hypertext, versus GRDDL, which uses a Turing-complete procedural “black box” to convert hypertext to RDF. Procedural semanticist Terry Winograd stated that “the operations on symbol structures in a procedural semantics need not correspond to valid logical inferences about the entities they represent” since “the symbol manipulation processes themselves are primary, and the rules of logic and mathematics are seen as an abstraction from a limited set of them” (15). AI researchers began exploring other options they termed more “procedural” using primarily Lisp, which they thought could better model the algorithmic and “temporal dimension” of meaning by transforming one “symbolic structure to another” (14). One such popular methodology was “semantic networks” that represent knowledge via “nodes and links” and as such is the direct ancestor of RDF.

While the procedural view of semantics first delivered impressive results through programs like SHRDLU (14), since the “semantics” were ad-hoc and task-dependent, so that they could not be used outside the limited domain in which they were created. When researchers attempted to communicate or combine their knowledge representation schemes, no-one really knew what the symbols in the language “meant” except the author. Even the “link” in semantic networks was interpreted in at least three different ways (16). In response to this crisis, declarative semantics over victorious over procedural semantics, and many of the ad-hoc “procedural” knowledge systems could be re-expressed in first order-logic (6). Furthermore, the focus on semantics allowed progress to be made on the issue of undecidability that was a severe problem for both procedural and declarative semantics, and researchers began reformulating less expressive forms of logic that maintained decidability, leading to “description logics” that evolved into OWL DL (9). The original RDF model that was published without formal semantics almost repeated a classical failure of AI, and so the refactoring of RDF based upon formal semantics gave RDF the ability to combine knowledge representations and communicate about them in an unambiguous manner, and so power “mash-ups” on a Web-scale. Declarative semantics has moved from a far-fetched model of intelligence in AI to the proper model for Web-scale engineering.

If history repeats itself, RDFa would naturally be the declarative victor over the procedural GRDDL. While upon first inspection this analogy is tempt-
ing, it is a bit facetious. First of all, GRDDL itself is a declarative method of describing a function that transforms XPath data models to RDF graphs, and there is no a priori reason why the transformation itself has to be “ad-hoc” and cannot have well-defined semantics. Furthermore, is RDFa truly so innocent of computation so that “what you see is what you get?” Inference is possible from RDFa, so statements in RDFa may entail further logical statements. Indeed, the relationship between logical procedures and computational procedures runs more deeply than one thinks.

7 The Curry-Howard Isomorphism

The debate between declarative and procedural semantics is, under the right formal conditions, a red herring. While the declarative logicists were refining their knowledge representations for decidability, the Lisp programmers that motivated procedural semantics began disciplining their programs with their own formal semantics that tightly coupled with typed lambda calculi. The Curry-Howard Isomorphism states that there is a tight coupling between logics and functional programming, often given by the slogan “Proofs are Programs” (12). The simplification of proofs can be equivalent to steps of computation. First formulated as a correspondence between the typed lambda calculus and intuitionistic logic by Curry and Howard, it was called “Propositions are Types” since the types of programs can be seen as formulae. So ‘p is a proof of proposition P’ is equivalent to ‘p is of type P’ and both can be written as p : P. This result has been generalized for many logical formalisms, such as the Girard-Reynolds isomorphism for linear logic (13). So the debate between procedural and declarative semantics can for these formalisms be resolved, as they are simply two different manners of describing the same system.

While the functional programming language theory community has exploited this result with much success, the result has in general not received any attention from the knowledge representation community. This is unfortunate, as there is no reason besides deployment why the procedural Javascript has to be the dominant language of the Web 2.0. Functional programming works well in widely distributed environments and the use of declarative logic has been crucial in Semantic Web prototypes of trust and secure computing, and transformations that automatically generated trusted proofs would be of vast utility, solving Javascript’s problems with cross-domain trust.

As regards GRDDL, while currently GRDDL transformations are almost entirely in XSLT, GRDDL can use any programming language for its transformation language. XSLT has been compared to “programming in COBOL” with regards to many tasks, and a GRDDL transformation that allowed a declarative proof to be generated step-in-step with its computations over an
XPath data model and that guaranteed the formal properties of the program would be far preferable to the Turing-complete black box of current GRDDL transformations.

Our example proof in Figure 3 uses intuitionistic logic, explicitly forbidding the use of the Law of the Excluded Middle \((A \lor \neg A)\), since to judge any statement to be true we must be able to construct a proof of it. So the computational result \(p\) can prove proposition \(P\) only if there exists a \(p : P\), and since one can not prove \(\neg A\) if one can construct a proof of \(A\), \(\neg A\) can not be given a truth value. This matches “open-world assumption” of the Web where statements are always subject to change given the possibility of new information. We use natural deduction-style proofs to make proofs about transformations. One common transformation wanted is one that converts microformats in non-XML HTML with microformats and converts it to XHTML and RDFa. We can arrange this as a function composition, since if we have a GRDDL transform that takes a possibly invalid HTML document \(H\) and applies the HTML 5 algorithm to convert it to a valid XPath Data Model \(X\) \((H \Rightarrow X)\) and then we apply a further transformation that replaces the microformats with their URI equivalents \((X \Rightarrow R)\). We can then prove that if we have these two functions (and all the needed information decryption requires) we can take a hypertext document and produce a version in RDFa \((H \Rightarrow R)\), i.e. transitivity of implication.

In natural deduction assumptions are given numeric subscript, and the application of rules to eliminate them share the subscript, while an alphabetic subscript is used as a shorthand for a type statement. Everything that is assumed is given in square brackets, and rules are both introduced \((\Rightarrow I)\) and eliminated \((\Rightarrow E)\). Due to the Curry-Howard Isomorphism, inference steps have standard computational equivalents, as shown in our example proof in Figure 3.

\[
\begin{align*}
  x : (X \Rightarrow R) & \quad [h : H]_1 \quad a : (H \Rightarrow X) \quad (\Rightarrow E) \\
  (ah) : X & \quad (\Rightarrow E) \\
  (x(ah)) : R & \quad (\Rightarrow E) \\
  \chi h. (x(ah)) & : (H \Rightarrow R) \quad (\Rightarrow I)_1 \\
\end{align*}
\]

\text{Fig. 3. Example Proof using Curry Howard Isomorphism}

8 The Future of Hypertext and Semantics

Is GRDDL enough to align the Hypertext Web with the Semantic Web? There are both practical and theoretical issues that deserve more inquiry. Practically, many microformats do not have profile documents that a GRDDL-aware agent can retrieve, and many microformats with profiles do not specify them in
their source documents. The creation of a unified default profile for currently existing microformats for GRDDL is one solution. Microformats which are in invalid HTML can be converted over to an XPath Data Model by the earlier mentioned using the HTML to DOM algorithm being developed for HTML 5, as used in our example in Figure 3. Currently no default XML processing model leads from stream of bytes specified as an XML document to a XPath data model, and so issues like having a GRDDL-aware namespace specified only in the DTD are unaddressed. Since XML documents themselves cannot be guaranteed to be unambiguous, the resulting triples could be ambiguous unless a processing language that specifies the processing model can be deployed (5).

RDFa does not share in these problems, yet relies on content authors, not vocabulary and profile namespace maintainers, to explicitly create RDFa, which would be of generally higher-cost than creating microformats, since RDFa requires the use of full URIs rather than the shorthand used by microformats. Furthermore, since RDFa cannot currently be used in XML vocabularies besides XHTML or in XML Schema-level validation, XML is currently left with GRDDL as the only standardized and self-describing methodology to move information from XML to RDF. Since GRDDL is not capable of converting RDF back to XML and thus round-tripping between formats.

The crux of the future of the Semantic Web lies in emphasizing to users the actual benefits of converting from the Hypertext Web to the Semantic Web, primarily the use of RDF as a “mash-up” data format. Many services replicate the same kind of data. If one has many different accounts on a number of ranging social networking services, such as a user with a Myspace and a Twitter account, the user must continually re-enter their data and keep their data segregated in the “closed gardens” of each service, where they cannot be easily accessed and used by other services. By using GRDDL and RDFa, users can convert data in hypertext to RDF in order to power Web-scale semantic “mash-ups” while maintaining their own control over the privacy and use of their data. For this to be the case, current languages like Javascript have to at least use RDF natively. Beyond Javascript, new languages have to be designed that base themselves on theory such as the Curry-Howard isomorphism. By resolving the false dichotomy between procedural and declarative semantics, the Curry-Howard isomorphism shows that under certain conditions computations simultaneously are the proofs needed to provide trusted computing over heterogeneous combinations of data. This hybridization of theory and principles of Web architecture4 provides the foundations needed for a new generation of Web-based applications that move the Web beyond the “closed garden” of the Web 2.0 to the open world of the Semantic Web.

4 The use of URIs, the “open world” assumption, and self-describing documents.
References


