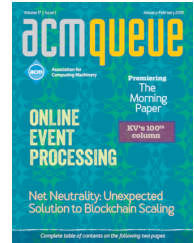# acmqueue

The January/February 2019 issue of acmqueue is out now

Subscribers and ACM Professional members login here

Databases

December 15, 2015
Volume 13, issue 9

📄 PDF

Schema.org: Evolution of Structured Data on the Web

Big data makes common schemas even more necessary.

R.V. Guha, Google
Dan Brickley, Google
Steve Macbeth, Microsoft

Separation between content and presentation has always been one of the important design aspects of the Web. Historically, however, even though most Web sites were driven off structured databases, they published their content purely in HTML. Services such as Web search, price comparison, reservation engines, etc. that operated on this content had access only to HTML. Applications requiring access to the structured data underlying these Web pages had to build custom extractors to convert plain HTML into structured data. These efforts were often laborious and the scrapers were fragile and error prone, breaking every time a site changed its layout.

Recent proliferation of devices with widely varying form factors has dramatically increased the number of different presentation formats that Web sites have to target. At the same time, a number of new personal assistant applications such as

Google App and Microsoft's Cortana have started providing sites with new channels for reaching their users. Further, mature Web applications such as Web search are increasingly seeking to use the structured content, if any, to power richer and more interactive experiences. These developments have finally made it vital for both Web and application developers to be able to exchange their structured data in an interoperable fashion.

This article traces the history of efforts to enable Web-scale exchange of structured data and reports on Schema.org, a set of vocabularies based on existing standard syntax, in widespread use today by both publishers and consumers of structured data on the Web. Examples illustrate how easy it is to publish this data and some of the ways in which applications use this data to deliver value to both users and publishers of the data.

Standards

Early on it became clear that domain-independent standards for structured data would be very useful. One approach, XML, attempted to standardize the syntax. While XML was initially thought of as the future of browser-based HTML, it has found more utility for structured data, with more traditional data-interoperability scenarios.

Another approach, MCF[18] (Meta Content Framework), introduced ideas from knowledge representation (frames and semantic nets) to the Web and proposed going further by using a common data model—namely, a directed labeled graph. Its vision was to create a single graph (or knowledge base) about a wide range of entities, different parts of which would come from different sites. An early diagram of this vision is shown in figure 1, in which information about Tori Amos is pulled together from different sites of that era into a single coherent graph.

FIGURE 1: **EXAMPLE OF A KNOWLEDGE BASE SOURCED FROM MULTIPLE SITES**
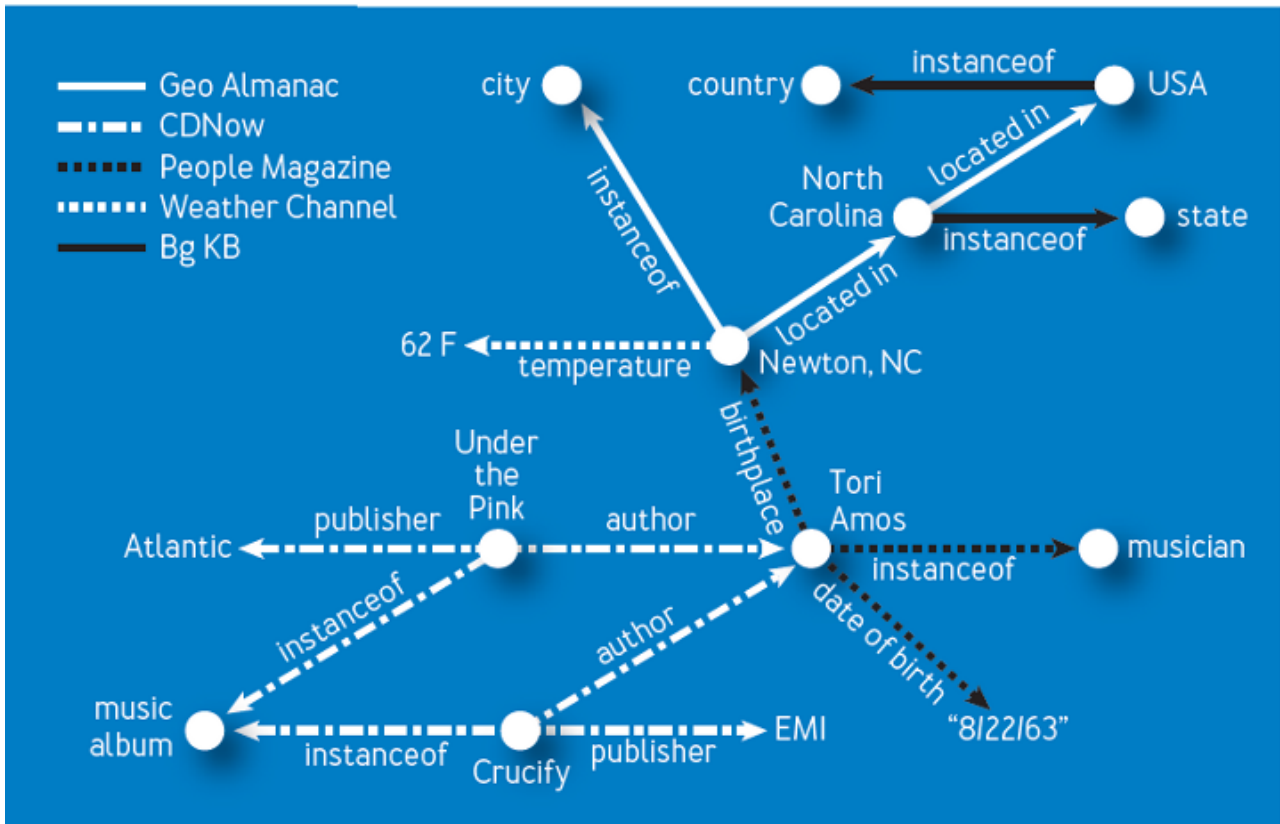
The hope at that time was to enable many different applications to work easily with data from many different sites. Over time, the vision grew to cover all kinds of intelligent processing of data on the Web. A 2001 *Scientific American* article by Tim Berners-Lee et al. on the Semantic Web was probably the most ambitious and optimistic view of this program.[5]

Between 1997 and 2004 various standards (RDF, RDFS, and OWL) were developed for the syntax and data model. A number of vocabularies were proposed for specific verticals, some of which were widely adopted. One of these was RSS (Rich Site Summary), which allowed users to customize home pages such as Netscape's Netcenter and Yahoo's My Yahoo with their favorite news sources. Another was vCard/hCard (i.e., IMC's vCard standard, expressed in HTML using microformat via the CSS class attribute), which was used to exchange contact information between contact managers, e-mail programs, etc. These were later joined by hCalendar, a format for calendar exchange, again a microformats HTML re-expression of an existing IETF (Internet Engineering Task Force) standard, iCalendar. FOAF (Friend of a Friend) predated these efforts but saw its usage for social-network data decline as that industry matured.[11] It has found a niche in the RDF (Resource Description Framework) Linked Data community as a commonly reused schema.[6]

In each of these cases where structured data was being published, one class of widely used application consumed it. Since the goal was to create a graph with wide coverage, well beyond narrow verticals, the challenge was to find a widely used application that had broad coverage. This application turned out to be text search.

The intense competition in Web search led companies to look beyond the ranking of results to improve search results. One technique used first by Yahoo and then Google was to augment the snippet associated with each search result with structured data from the results page.

They focused on a small number of verticals (eventually around ten, such as recipes, events, etc.), each with a prescribed vocabulary, reusing existing vocabularies such as hCard and FOAF when appropriate. For each, they augmented the snippet with some structured data so as to optimize the user's and webmaster's experience. This approach led to much greater adoption, and soon a few hundred thousand sites were marking up their pages with structured data markup. The program had a substantial drawback, however. The vocabularies for the different verticals were completely independent, leading to substantial duplication and confusion. It was clear that extending this to hundreds or thousands of verticals/classes was impossible. To make things worse, different search engines recommended different vocabularies.

Because of the resulting confusion, most webmasters simply did not add any markup, and the markup they did add was often incorrectly formatted. This abundance of incorrect formatting required consumers of markup to build complex parsers that were able to handle improperly formed syntax and vocabulary. These complex parsers turned out to be just as brittle as the original systems used to extract structured data from HTML and thus didn't result in the expected advances.

Schema.org

In 2011, the major search engines Bing, Google, and Yahoo (later joined by Yandex) created Schema.org to improve this situation. The goal was to provide a single schema across a wide range of topics that included people, places, events, products, offers, and so on. A single integrated schema covered these topics. The idea was to present webmasters with a single vocabulary. Different search engines might use the markup differently, but webmasters had to do the work only once and would reap the benefits across multiple consumers of the markup.

Schema.org was launched with 297 classes and 187 relations, which over the past four years have grown to 638 classes and 965 relations. The classes are organized into a hierarchy, where each class may have one or more superclasses (though most have only one). Relations are polymorphic in the sense that they have one or more domains and one or more ranges. The class hierarchy is meant more as an organizational tool to help browse the vocabulary than as a representation of common sense, à la Cyc.

FIGURE 2: **RESTAURANT RESERVATION E-MAIL MARKUP (MICRODATA SYNTAX)**

```
<p itemscope
itemtype="http://schema.org/FoodEstablishmentReservation">
     Your reservation for <span itemprop="partySize">3</span>
     at Local Edition is
     <link itemprop="reservationStatus"
     href="http://schema.org/Confirmed"/>confirmed</link> for
     <timeitemprop="startTime" datetime="2015-05-02T18:30:00Z">May
2nd,2015 at 6:30 PM</time>.
     The reservation is held under:
<span itemscope itemtype="http://schema.org/Person">
     <span itemprop="givenName">Dan Brickley</span>.
     </span>
Serve yourself when you arrive.
<span itemscope itemtype="http://schema.org/Restaurant">
     <meta itemprop="telephone" content="tel:+1-202-555-0125" />
     To get there:<br />
     <span itemprop="name"> Local Edition </span><br />
     <span itemprop="address" itemscope itemtype="http://schema.
org/PostalAddress">
          <span itemprop="streetAddress">2370 South Market
Street, San Francisco, USA.</span>
     </span>
</span>
</p>
```

us for another event at the

ns - Visit the Carnegie Mellon University h

en invited to the following event. Title   RSVP ▾

Drinks with Catelyn
via Google Calendar

🕔 May 2 6:30pm – 7:30pm

📍 Local Edition, 2370 S. Market St. San
    Francisco, CA

👥 3 people going

| Yes | Maybe | No |

about the delay on this. I was f

The first application to use this markup was Google's Rich Snippets, which switched over to Schema.org vocabulary in 2011. Over the past four years, a number of different applications across many different companies have started using Schema.org vocabulary. Some of the more prominent among these include the following:

* In addition to per-link Rich Snippets, annotations in Schema.org are used as a data source for the Knowledge Graph, providing background information about well-known entities (e.g., logo, contact, and social information).

* Schema.org-based structured data markup is now being used in places such as e-mails. For example, e-mails confirming reservations (restaurant, hotel, airline, etc.), purchase receipts, etc. have embedded Schema.org markup with details of the transaction. This approach makes it possible for e-mail assistant tools to extract the structured data and make it available through mobile notifications, maps, calendars, etc. Google's Gmail and Search products use this data to provide notifications and reminders (figure 2). For example, a dinner booking made on Opentable.com will trigger a reminder for leaving for the restaurant, based on the location of the restaurant, the user, traffic conditions, etc.

* Microsoft's Cortana (for Windows 10 and Windows phones) makes use of Schema.org from e-mail messages, as shown in figure 3.

* Yandex uses many parts of Schema.org, including recipes, autos, reviews, organizations, services, and directories. Its earlier use of FOAF (corresponding to the popularity of the LiveJournal social network in Russia) demonstrated the need for pragmatic vocabulary extensions that support consumer-facing product features.

* Pinterest uses Schema.org to provide rich pins for recipe, movie, article, product, or place items.

* Apple's iOS 9 (Searchlight/Siri) uses Schema.org for search features including aggregate ratings, offers, products, prices, interaction counts, organizations, images, phone numbers, and potential website search actions. Apple also uses Schema.org within RSS for news markup.

FIGURE 3: **FLIGHT RESERVATION E-MAIL MARKUP (JSON-LD SYNTAX) AND ITS USE IN MICROSOFT'S CORTANA**
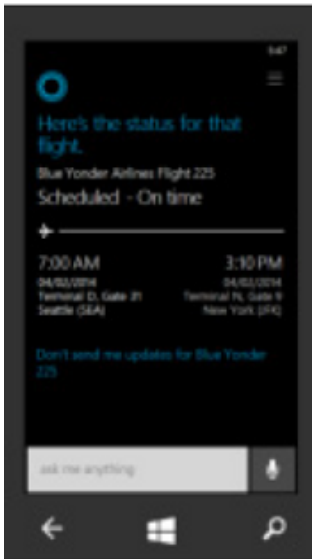
```json
{ "@context": "http://schema.org/",
  "@type": "FlightReservation",
  "reservationNumber": "QWERT0123456789",
  "reservationStatus":
"http://schema.org/Confirmed",
  "underName":{
    "@type": "Person",
    "name": "Estella Gallagher"
  },
  "reservationFor": {
    "@type": "Flight",
    "flightNumber": "123",
    "departureAirport": {
      "@type": "Airport",
      "name": " Seattle-Tacoma International
Airport",
      "iataCode": "SEA"
    },
    "arrivalAirport": {
      "@type": "Airport",
      "name": " John F Kennedy International
Airport",
      "iataCode": "JFK"
    },
    "departureTime": "2014-04-02T10:32:00Z",
    "arrivalTime": "2014-04-02T11:45:00Z",
    "airline": {
      "@type": "Airline",
      "name": "Blue Yonder Airlines",
      "iataCode": "BY"
    }
  }
}
```

Adoption Statistics

The key measure of success is, of course, the level of adoption by webmasters. A sample of 10 billion pages from a combination of the Google index and Web Data Commons provides some key metrics. In this sample 31.3 percent of pages have Schema.org markup, up from 22 percent one year ago. On average, each page containing this markup makes references to six entities, making 26 logical assertions among them. Figure 4a lists well-known sites within some of the major verticals covered by Schema.org, showing both the wide range of topics that are covered and the adoption by the most popular sites in each of these topics. Figures 4b and 4c list some of the most frequently used types and relations. Extrapolating from the numbers in this sample, we estimate that at least 12 million sites use Schema.org markup. The important point to note is that structured data markup is now of the same order of magnitude as the Web itself.

FIGURE 4A: MAJOR SITES THAT HAVE PUBLISHED SCHEMA.ORG

CATEGORY	SITES
News	nytimes.com, guardian.com, bbc.co.uk
Movies	imdb.com, rottentomatoes.com, movies.com
Jobs / Careers	careerjet.com, monster.com, indeed.com
People	linkedin.com, pinterest.com, familysearch.org, archives.com
Products	ebay.com, alibaba.com, sears.com, cafepress.com, sulit.com, fotolia.com
Video	youtube.com, dailymotion.com, frequency.com, vinebox.com
Medical	cvs.com, drugs.com
Local	yelp.com, allmenus.com, urbanspoon.com
Events	wherevent.com, meetup.com, zillow.com, eventful.com
Music	last.fm, myspace.com, soundcloud.com

FIGURE 4B: MOST FREQUENTLY USED TYPES*

WebSite, SearchAction, WebPage, Product, ImageObject, Person, Offer, BlogPosting, Organization, Article, PostalAddress, Blog, LocalBusiness, AggregateRating, WPFooter, SiteNavigationElement, WPHeader, WPSideBar, CreativeWork, Review, EntryPoint, ViewAction, Place, Rating, ItemList, Event, ListItem, VideoObject, GeoCoordinates, Thing, SocialMediaPosting, UserComments, ProfilePage, Restaurant, Brand, OpeningHoursSpecification, CollectionPage, Recipe, QuantitativeValue, RealEstateAgent, NewsArticle, ItemPage, JobPosting, MusicGroup, ImageGallery, MusicRecording, WPAdBlock, Store

FIGURE 4C: MOST FREQUENTLY USED PROPERTIES**

name, url, description, image, target, query-input, potentialAction, datePublished, author, articleBody, null, price, offers, contentURL, address, telephone, addressLocality, priceCurrency, availability, streetAddress, headline, postalCode, thumbnailUrl, addressRegion, ratingValue, mainContentOfPage, blogPost, aggregateRating, text, logo, sku, postId, blogId, image_url, bestRating, inLanguage, reviewCount, breadcrumb, email, urlTemplate, keywords, ratingCount, addressCountry, reviewRating, itemListElement, sameAs, openingHours, position, location, worstRating, startDate

*from public Web, ignores markup in e-mail, etc.

**as of July 2015

Although this article does not present a full analysis and comparison, we should emphasize that various other formats are also widespread on the Web. In

particular, OGP (Open Graph Protocol) and microformat approaches can be found on approximately as many sites as Schema.org, but given their much smaller vocabularies, they appear on fewer than half as many pages and contain fewer than a quarter as many logical assertions. At this point, Schema.org is the only broad vocabulary that is used by more than a quarter of the pages found in the major search indices.

A key driver of this level of adoption is the extensive support from third-party tools such as Drupal and Wordpress extensions. In verticals (such as events), support from vertical-specific content-management systems (such as Bandsintown and Ticketmaster) has had a substantial impact. A similar phenomenon was observed with the adoption of RSS, where the number of RSS feeds increased dramatically as soon as tools such as Blogger started outputting RSS automatically.

The success of Schema.org is attributable in large part to the search engines and tools rallying behind it. Not every standard pushed by big companies has succeeded, however. Some of the reason for Schema.org's success lies with the design decisions underlying it.

Design Decisions

The driving factor in the design of Schema.org was to make it easy for webmasters to publish their data. In general, the design decisions place more of the burden on consumers of the markup. This section addresses some of the more significant design decisions.

Syntax

From the beginning, Schema.org has tried to find a balance between pragmatically accepting several syntaxes versus making a clear and simple recommendation to webmasters. Over time it became clear that multiple syntaxes would be the best approach. Among these are RDFa (Resource Description Framework in Attributes) and JSON-LD (JavaScript Object Notation for Linked Data), and publishers have their own reasons for preferring one over another.

In fact, in order to deal with the complexity of RDFa 1.0, Schema.org promoted a newer syntax, Microdata, that was developed as part of HTML5. Design choices for Microdata were made through rigorous usability testing on webmasters. Since

then, prompted in part by Microdata, revisions to RDFa have made it less complex, particularly for publishers.

Different syntaxes are appropriate for different tools and authoring models. For example, Schema.org recently endorsed JSON-LD, where the structured data is represented as a set of JavaScript-style objects. This works well for sites that are generated using client-side JavaScript as well as in personalized e-mail where the data structures can be significantly more verbose. There are a small number of content-management systems for events (such as concerts) that provide widgets that are embedded into other sites. JSON-LD allows these embedded widgets to carry structured data in Schema.org. In contrast, Microdata and RDFa often work better for sites generated using server-side templates.

It can sometimes help to idealize this situation as a tradeoff between machine-friendly and human-friendly formats, although in practice the relationship is subtler. Formats such as RDF and XML were designed primarily with machine consumption in mind, whereas microformats have a stated bias toward humans first. Schema.org is exploring the middle ground, where some machine-consumption convenience is traded for publisher usability.

Polymorphism

Many frame-based KR (knowledge representation) systems, including RDF Schema, OWL (Web Ontology Language), etc., have a single domain and range for each relation. This, unfortunately, leads to many unintuitive classes whose only role is to be the domain or range of some relation. This also makes it much harder to reuse existing relations without significantly changing the class hierarchy. The decision to allow multiple domains and ranges seems to have significantly ameliorated the problem. For example, though there are various types (Events, Reservations, Offers) in Schema.org whose instance can take a startDate property, the polymorphism has allowed us to get away with not having a common supertype (such as Temporally Commencable Activity) in which to group these.

Entity References

Many models such as Linked Data have globally unique URIs for every entity as a core architectural principle.[4] Unfortunately, coordinating entity references with other sites for the tens of thousands of entities about which a site may have

information is much too difficult for most sites. Instead, Schema.org insists on unique URIs for only the very small number of terms provided by Schema.org. Publishers are encouraged to add as much extra description to each entity as possible so that consumers of the data can use this description to do entity reconciliation. While this puts a substantial additional burden on applications consuming data from multiple Web sites, it eases the burden on webmasters significantly. In the example shown in figure 1, instead of requiring common URIs for the entities (for example, Tori Amos; Newton, NC; and Crucify), of which there are many hundreds of millions (with any particular site using potentially hundreds of thousands), webmasters have to use standard vocabulary only for terms such as *country*, *musician*, *date of birth*, etc., of which there are only a few thousand (with any particular site using at most a few dozen). Schema.org does, however, also provide a sameAs property that can be used to associate entities with well-known pages (home pages, Wikipedia, etc.) to aid in reconciliation, but this has not found much adoption.

Incremental Complexity

Often, making the representation too simplistic would make it hard to build some of the more sophisticated applications. In such cases, we start with something simple, which is easy for webmasters to implement, but has enough data to build a motivating application. Typically, once the simple applications are built and the vocabulary gets a minimal level of adoption, the application builders and webmasters demand a more expressive vocabulary—one that might have been deemed too complex had we started off with it.

At this point, it's possible to add the complexity of a more expressive vocabulary. Often this amounts to the relatively simple matter of adding a few more descriptive properties or subtypes. For example, adding new types of actions or events is a powerful way of extending Schema.org's expressivity. In many situations, however, closer examination reveals subtle differences in conceptualization. For example, creative works have many different frameworks for analyzing seemingly simple concepts, such as *book,* into typed, interrelated entities (e.g., in the library world, FRBR [functional requirements for bibliographic records]); or with e-commerce *offers*, some systems distinguish manufacturer warranties from vendor warranties. In such situations there is rarely a right answer. The Schema.org approach is to be led by practicalities—the data fields available in the wider Web and the information requirements of applications that can motivate large-scale publication. Schema.org definitions are never changed in

pursuit of the perfect model, but rather in response to feedback from publishers and consumers.

Schema.org's incremental complexity approach can be seen in the interplay among evolving areas of the schema. The project has tried to find a balance between two extremes: uncoordinated addition of schemas with overlapping scopes versus overly heavy coordination of all topics. As an example of an area where we have stepped back from forced coordination, both creative works (books, etc.) and e-commerce (product descriptions) wrestle with the challenge of describing versions and instances of various kinds of mass-produced items. In professional bibliographies, it is important to describe items at various levels (e.g., a particular author-signed copy of a particular paperback versus the work itself, or the characteristics of that edition such as publisher details). Surprisingly similar distinctions need to be made in e-commerce when describing nonbibliographic items such as laser printers. Although it was intellectually appealing to seek schemas that capture a "grand theory of mass produced items and their common properties," Schema.org instead took the pragmatic route and adopted different modeling idioms for bibliography[12] and e-commerce.[8]

It was a pleasant surprise, by contrast, to find unexpected common ground between those same fields when it was pointed out that Schema.org's concept of an *offer* could be applied in not-for-profit fields beyond e-commerce, such as library lending. A few community-proposed pragmatic adjustments to our definitions were needed to clarify that offers are often made without expectation of payment. This is typical of our approach, which is to publish schemas early in the full knowledge that they will need improving, rather than to attempt to perfect everything prior to launch. As with many aspects of Schema.org, this is also a balancing act: given strong incentives from consumers, terms can go from nothing to being used on millions of sites within a matter of months. This provides a natural corrective force to the desire to continue tweaking definitions; it is impractical (and perhaps impolite) to change schema definitions too much once they have started to gain adoption.

Cleanup

Every once in a while, we have gotten carried away and have introduced vocabulary that never gets meaningful usage. While it is easy to let such terms lie around, it is better to clean them out. Thus far, this has happened only with large vocabularies that did not have a strong motivating application.

Extensions

Given the variety of structured data underlying the Web, Schema.org can at best hope to provide the core for the most common topics. Even for a relatively common topic such as automobiles, potentially hundreds of attributes are required to capture the details of a car's specifications as found on a manufacturer's Web site. Schema.org's strategy has been to have a small core vocabulary for each such topic and rely on extensions to cover the tail of the specification.

From the beginning there have been two broad classes of extensions: those that are created by the Schema.org community with the goal of getting absorbed into the core, and those that are simply deployed "in the wild" without any central coordination. In 2015 the extension mechanism was enhanced to support both of these ideas better. First, the notion of *hosted* extensions was introduced; these are terms that are tightly integrated into Schema.org's core but treated as additional (in some sense optional) layers. Such terms still require coordination discussion with the broader community to ensure consistent naming and to identify appropriate integration points. The layering mechanism, however, is designed to allow greater decentralization to expert and specialist communities.

Second came the notion of *external* extensions. These are independently managed vocabularies that have been designed with particular reference to Schema.org's core vocabulary with the expectation of building upon, rather than duplicating, that core. External extensions may range from tiny vocabularies that are product/service-specific (e.g., for a particular company's consumption), geographically specific (e.g., US-Healthcare), all the way to large schemas that are on a scale similar to Schema.org.

We have benefited from Schema.org's cross-domain data model. It has allowed a form of loosely coupled collaboration in which topic experts can collaborate in dedicated fora (e.g., sports, health, bibliography), while doing so within a predictable framework for integrating their work with other areas of Schema.org.

The more significant additions have come from external groups that have specific interests and expertise in an area. Initially, such collaborations were in a project-to-project style, but more recently they have been conducted through individual engagement via W3C's Community Group mechanism and the collaboration platform provided by GitHub.

The earliest collaboration was with the IPTC's rNews initiative, whose contributions led to a number of term additions (e.g., NewsArticle) and improvements to support the description of news. Other early additions include healthcare-related schemas, e-commerce via the inclusion of the GoodRelations project, as well as LRMI (Learning Resources Metadata Initiative), a collaboration with Creative Commons and the Association of Educational Publishers.

The case of TV and radio markup illustrates a typical flow, as well as the evolution of our collaborative tooling.[9] Schema.org began with some rough terminology for describing television content. Discussions at W3C identified several ways in which it could be improved, bringing it more closely in line with industry conventions and international terminology, as well as adding the ability to describe radio content. As became increasingly common, experts from the wider community (BBC, EBU, and others) took the lead in developing these refinements (at the time via W3C's wikis and shared file systems), which in turn inspired efforts to improve our collaboration framework. The subsequent migration to open-source tooling hosted on GitHub in 2014 has made it possible to iterate more rapidly, as can be seen from the project's release log, which shows how the wider community's attention to detail is being reflected in fine-grained improvements to schema details.[10]

Schema.org does not mandate exactly how members of the wider community should share and debate ideas—beyond a general preference for public fora and civil discussion. Some groups prefer wikis and IRC (Internet Relay Chat); others prefer Office-style document collaborative authoring, telephones, and face-to-face meetings. Ultimately, all such efforts need to funnel into the project's public GitHub repository. A substantial number of contributors report problems or share proposals via the issue tracker. A smaller number of contributors, who wish to get involved with more of the technical details, contribute specific changes to schemas, examples, and documentation.

Related Efforts

Since 2006 the "Linked Data" slogan has served to redirect the W3C RDF community's emphasis from Semantic Web ontology and rule languages toward open-data activism and practical data sharing. Linked data began as an informal note from Tim Berners-Lee that critiqued the (MCF-inspired) FOAF approach of using reference by description instead of "URIs everywhere":[3]

"This linking system was very successful, forming a growing social network, and dominating, in 2006, the linked data available on the Web. However, the system has the snag that it does not give URIs to people, and so basic links to them cannot be made."

Linked-data advocacy has successfully elicited significant amounts of RDF-expressed open data from a variety of public-sector and open-data sources (e.g., in libraries,[14] the life sciences,[16] and government.[15] A strong emphasis on identifier reconciliation, complex best practice rules (including advanced use of HTTP), and use of an arbitrary number of partially overlapping schemas, however, have limited the growth of linked-data practices beyond fields employing professional information managers. Linked RDF data publication practices have not been adopted in the Web at large.

Schema.org's approach shares a lot with the linked-data community: it uses the same underlying data model and schema language,[17] and syntaxes (e.g., JSON-LD and RDFa), and shares many of the same goals. Schema.org also shares the linked-data community's skepticism toward the premature formalism (rule systems, description logics, etc.) found in much of the academic work that is carried out under the Semantic Web banner. While Schema.org also avoids assuming that such rule-based processing will be commonplace, it differs from typical linked-data guidelines in its assumption that various other kinds of cleanup, reconciliation, and post-processing will usually be needed before structured data from the Web can be exploited in applications.

Linked data aims higher and has consequently brought to the Web a much smaller number of data sources whose quality is often nevertheless very high. This opens up many opportunities for combining the two approaches—for example, professionally published linked data can often authoritatively describe the entities mentioned in Schema.org descriptions from the wider mainstream Web.

Using unconstrained combinations of identifying URIs and unconstrained combinations of independent schemas, linked data can be seen as occupying one design extreme. A trend toward Google Knowledge Graphs can be viewed at the other extreme. This terminology was introduced in 2012 by Google, which presented the idea of a Knowledge Graph as a unified graph data set that can be used in search and related applications. In popular commentary, Google's (initially Freebase-based) Knowledge Graph is often conflated with the specifics of its visual presentation in Google's search results—typically as a simple factual panel. The terminology is seeing some wider adoption.

The general idea builds upon common elements shared with linked data and Schema.org: a graph data model of typed entities with named properties. The Knowledge Graph approach, at least in its Google manifestation, is distinguished in particular by a strong emphasis on up-front entity reconciliation, requiring curation discipline to ensure that new data is carefully integrated and linked to existing records. Schema.org's approach can be seen as less noisy and decentralized than linked data, but more so than Knowledge Graphs. Because of the shared underlying approach, structured data expressed as Schema.org is a natural source of information for integration into Knowledge Graphs. Google documents some ways of doing so.[7]

Lessons

Here are some of the most important lessons we have learned thus far, some of which might be applicable to other standards efforts on the Web. Most are completely obvious but, interestingly, have been ignored on many occasions.

1. Make it easy for publishers/developers to participate. More generally, when there is an asymmetry in the number of publishers and the number of consumers, put the complexity with the smaller number. They have to be able to continue using their existing tools and workflows.

2. No one reads long specifications. Most developers tend to copy and edit examples. So, the documentation is more like a set of recipes and less like a specification.

3. Complexity has to be added incrementally, over time. Today, the average Web page is rather complex, with HTML, CSS, JavaScript, etc. It started out being very simple, however, and the complexity was added mostly on an as-needed basis. Each layer of complexity in a platform/standard can be added only after adoption of more basic layers.

Conclusion

The idea of the Web infrastructure requiring structured data mechanisms to describe entities and relationships in the real world has been around for as long as the Web itself.[1,2,13] The idea of describing the world using networks of typed relationships was well known even in the 1970s, and the use of logical statements about the world has a history predating computing. What is surprising is just how

hard it was for such seemingly obvious ideas to find their way into the Web as an information platform. The history of Schema.org suggests that rather than seeking directly to create "languages for intelligent agents," addressing vastly simpler scenarios from Web search has turned out to be the best practical route toward structured data for artificial personal assistants.

Over the past four years, Schema.org has evolved in many ways, both organizationally and in terms of the actual schemas. It started with a couple of individuals who created an informal consortium of the three initial sponsor companies. In the first year, these sponsor companies made most decisions behind closed doors. It incrementally opened up, first moving most discussions to W3C public forums, and then to a model where all discussions and decision making are done in the open, with a steering committee that includes members from the sponsor companies, academia, and the W3C.

Four years after its launch, Schema.org is entering its next phase, with more of the vocabulary development taking place in a more distributed fashion. A number of extensions, for topics ranging from automobiles to product details, are already under way. In such a model, Schema.org itself is just the core, providing a unifying vocabulary and congregation forum as necessary.

The increased interest in big data makes the need for common schemas even more relevant. As data scientists are exploring the value of data-driven analysis, the need to pull together data from different sources and hence the need for shared vocabularies is increasing. We are hopeful that Schema.org will contribute to this.

Acknowledgments

References

1. Berners-Lee, T. 1989. Information management: a proposal;
http://www.w3.org/History/1989/proposal.html.

2. Berners-Lee, T. 1994. W3 future directions;
http://www.w3.org/Talks/WWW94Tim/.

3. Berners-Lee, T. 2006. Linked Data;
http://www.w3.org/DesignIssues/LinkedData.html.

4. Berners-Lee, T. 2010. Is your linked open data 5 star?
http://www.w3.org/DesignIssues/LinkedData#fivestar.

5. Berners-Lee, T., Hendler, J., Lassila, O. 2001. The semantic web. *Scientific American* (May): 29–37; http://www.scientificamerican.com/article/the-semantic-web/.

6. Friend of a Friend vocabulary (foaf);
http://lov.okfn.org/dataset/lov/vocabs/foaf.

7. Google Developers. 2015. Customizing your Knowledge Graph;
https://developers.google.com/structured-data/customize/overview.

8. Guha, R.V. 2012. Good Relations and Schema.org. Schema Blog;
http://blog.schema.org/2012/11/good-relations-and-schemaorg.html.

9. Raimond, Y. 2013. Schema.org for TV and radio markup. Schema Blog;
http://blog.schema.org/2013/12/schemaorg-for-tv-and-radio-markup.html.

10. Schema.org. Release log; http://schema.org/docs/releases.html.

11. Schofield, J. 2004. Let's be Friendsters. *The Guardian* (February 19);
http://www.theguardian.com/technology/2004/feb/19/newmedia.media.

12. Wallis, R., Scott, D. 2014. Schema.org support for bibliographic relationships and periodicals. Schema Blog; http://blog.schema.org/2014/09/schemaorg-support-for-bibliographic_2.html.

13. W3C. 1996. Describing and linking Web resources. Unpublished note;
http://www.w3.org/Architecture/NOTE-link.html.

14. W3C. 2011. Library Linked Data Incubator Group Final Report;
http://w3.org/2005/Incubator/lld/XGR-lld-20111025/.

15. W3C. 2011. Linked Data Cookbook;
http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook.

16. W3C. 2012. Health Care and Life Science Linked Data Guide; http://www.w3.org/2001/sw/hcls/notes/hcls-rdf-guide/.

17. W3C. 2014. RDF Schema 1.1; http://www.w3.org/TR/rdf-schema/

18. W3C. 1997. MCF Using XML, R.V.Guha, T.Bray, http://w3.org/TR/NOTE-MCF-XML

R.V. Guha is the creator of widely used Web standards such as RSS and Schema.org. He is also responsible for products such as Google Custom Search. He was a co-founder of Epinions.com and Alpiri. Earlier, he was co-leader of the Cyc project. He is currently a Google Fellow and a vice president in research at Google. He has a Ph.D. in computer science from Stanford University and a B.Tech in mechanical engineering from IIT Chennai.

Dan Brickley is best known for his work on Web standards in the W3C community, where he helped create the Semantic Web project and many of its defining technologies. Brickley works at Google on the Schema.org initiative and structured-data standards. Previous work included metadata projects around TV, agriculture, digital libraries, and education.

Steve Macbeth is partner architect in the application and service group at Microsoft, where he is responsible for designing and building solutions at the intersection of mobile, cloud, and intelligent systems. This work includes building platform technologies that will enable all applications, across all platforms, to understand users' behavior and preferences better, in order to behave more intelligently and learn over time. Prior to this role, Macbeth was a senior leader in the Bing Core Search, focused on overall search quality, relevance, and experimentation, and the general manager and co-founder of the Search Technology Center Asia, located in Beijing, China, where he lived and worked for three years. Prior to coming to Microsoft, he was the founder and CTO of Riptide Technologies and pcsupport.com, technology startups in Vancouver, Canada.

Originally published in Queue vol. 13, no. 9—
see this item in the ACM Digital Library

Tweet

Related:

Pat Helland – **Identity by Any Other Name**
The complex cacophony of intertwined systems

Raymond Blum, Betsy Beyer – **Achieving Digital Permanence**
The many challenges to maintaining stored information and ways to overcome
them

Graham Cormode – **Data Sketching**
The approximate approach is often faster and more efficient.

Heinrich Hartmann – **Statistics for Engineers**
Applying statistical techniques to operations data

Comments

(newest first)

Post a Comment:

name

email

Comment: (Required – 4,000 character limit – HTML syntax is **not** allowed and **will be removed**)

Post