Ubiquity, an ACM publication
November, 2010

Edging Toward the Semantic Web:

Protocols, Curation, and Seeds

by Espen Andersen,
The Norwegian School of Management

Editor's Introduction

The evolution from an interactive Internet (often called Web 2.0) toward a more intelligent, semantic web will not happen as a result of dramatic new inventions or jointly agreed standards, but through a gradual evolution and recombination of existing technologies. To get to a Web 3.0, we will need to first create (and maybe be satisfied with) a Web 2.5, and that will happen through the gradual evolution of effective, user-based interaction protocols (based on user dialogues) and the use of queries as information passing mechanisms.

Espen Andersen, a Ubiquity editor, explores this question and proposes a path. He is Associate Professor of Strategy at the Norwegian School of Management and a frequent writer and speaker on technology evolution and management topics.

Edging Toward the Semantic Web:

Protocols, Curation, and Seeds

by Espen Andersen,
The Norwegian School of Management

I have been told that in 1986, a conference about public computer networks was held at Harvard. A large concern was the lack of publicly available computer networks, then known as videotext, in the U.S. France had launched Minitel, a proprietary information service accessible through special terminals via the public telephone system, in 1982. Millions of Frenchmen were looking up phone numbers and ordering services electronically, with addressing, display and payment services taken care of. All the U.S. had to show for itself was a tangle of undisciplined academic incompatibilities known mainly as Arpanet and BITNET, as well as some private services such as CompuServe, with less than 200,000 subscribers. Clearly, if the U.S. ever was going to get anywhere, a centrally designed, publicly funded, focused and disciplined approach such as the French one was called for.

The rest, of course, is history. Arpanet and BITNET eventually became the Internet, the integrated functionality stack and proprietary nature of Minitel effectively hindered its technical and functional evolution, and now the world runs on an architecture created to share processing power and information protocols designed for browsing academic papers. Creative, layered evolution triumphed over carefully designed completeness.

Such is actually the fate of most technology innovations: A proprietary, closed system initially provides an idea of what is possible. But over time, the winner will be the one that deploys the less complete but vastly more flexible weapon of modularity.

Getting Forward From Here

As computers have become more powerful and communications faster, more and more information—be it data values or web pages—are dynamically computed rather than statically retrieved. A processing-poor system will need intermediate, pre-computed data, such as an estimated stock level or a pre-rendered graphic. A more powerful system will count the number of items that are or will be available for sale, and call that the stock level, possibly rendering the result in a graphical format directly. Add increases in storage capability, and the system can remember your display preferences and dynamically adapt to them, too.

Most of the performance increase, however, has been spent on making the machine easier to use by having it take over more and more of what the user previously had to do. Interfaces have evolving from toggle switches to command lines to a number of metaphors [1] such as desktop and offices, and finally to devices effectively hiding the fact that they are computers, such as the Apple iPad. This gradual insertion of new layers has not been limited to the human-computer interface. However, computers themselves increasingly interact in languages tailored less for efficiency than for human understanding. ASCII, object orientation, HTML and XML are all standards that sacrifice performance for ability to handle complexity. The semantic levels of interaction with and between computers have been raised.

For all their user-friendliness and vast information-processing capabilities, however, computers are still far from being perceived as “intelligent,” still the holy grail of computing. From Turing [2] to Kurzweil [3, 4], the endpoint of computer evolution has always been seen as something anthropomorphic, an entity that not only can communicate with humans on their own terms, but also relieve humans from the burden of having to deal with complexity. But if the experience of 50 years of user interface and information access technology is anything to go by, the most successful interaction between humans and computers take place when both adapt their behavior to each other—and the most successful interaction platforms, be they between humans or computers, those that allow for the evolution of their own protocols. While we may design a Web 3.0 and create the technologies for it, what eventually wins out is, normally, something halfway there in terms of technology specifications. The rest of it evolved as a matter of use rather than design.

So far, the route to creating a semantic web—a web that to a larger extent understands meaning rather than syntactically precise commands—has been one of definitions and ontologies, such as RDF/XML [10] or Topic Maps [5]. While worthy initiatives and useful tools, these standards are not the answer, for the same reason that the Minitel was not the answer: They allow well defined completeness by enforcing discipline.

I think we will get to the next generation Internet—semisemantic, if you will—through evolution and user interaction. To get there, we will use bits and pieces of existing technologies in new ways. Some of these pieces are exemplified by the responsive interaction protocol of the Palm Pilot, the curated datasets of Wolfram Alpha, and the emerging user behavior of treating search queries as information transfer mechanisms.

Step 1. Efficient Protocols: Learning From the Palm Pilot

Back in the early-to-mid-90s, Apple launched one of the first pen-based PDAs, the Apple Newton. The Newton was, for its time, an amazing technology, but for once Apple screwed it up, largely because they tried to make the device do too much. One critical error was the handwriting recognition software, which would let you write in your own handwriting, and then try to interpret it based on whole words or sentences. I am a physician's son, and have taken after my father in the handwriting department. Newton could not make sense of my scribbles, even if I tried to write clearly. Since handwriting recognition is computationally intensive, it also took a long time doing in concluding on a result. I bought one, and then sent it back in disgust.

Then came the Palm Pilot.

Small, light, with excellent battery capacity, the Palm Pilot could do one thing well: managing a calendar, a contact list, and a stack of notes. The input was the same as the Newton—handwriting—but rather than letting the user write in his or her own style, the Pilot demanded Graffiti, a alphabet-like set of signs which was recognized by the individual character rather than words and sentences. Most of Graffiti resembled regular characters enough that you could guess them, for the rest you could consult a small plastic card. The feedback was rapid, so experimenting usually worked well, and pretty soon you had learned—or, rather, your hand had learned—to enter the Graffiti characters rapidly and accurately [6].

In other words, by demanding that you, as a user, conform your input to that demanded by the computer, interaction speed improved. Less was more. And this is not limited to handwriting.

Semantics is really about symbols and shorthand. A word is created as shorthand for a more complicated concept by a process of internalization. When learning a language, rapid feedback helps (which is why I think it is easier to learn a language with a strict and terse grammar), simplicity helps, as does a culture of permissiveness in creating new words by relying on shared context and intuitive combinations. English has thrived at least partially because of its willingness to deal with imprecision and importing of new words to describe new concepts. Some social media applications, such as Twitter, evolve similar new ways of adding meaning to words, referring to other users with an @ in front (such as @stephenfry for a well known example) and searchable concepts with a # (for example, #semanticweb).

Ontologies, like language, evolves through information exchange between people. And, perhaps, machines.

Step 2. Trustworthy Search Results: Learning From Wolfram Alpha
Wolfram Alpha, officially launched in May 2009, is an exciting and rapidly evolving "computational search engine." Rather than looking up pre-existing documents, it actually computes the answer—every time. If you ask for the distance is to the moon, for example,

Google and other search engines will find you documents that tells you the average distance, whereas Wolfram Alpha will calculate the actual distance at that point in time. Wolfram Alpha does not store answers, but creates them every time. It primarily answers numerical, computable questions and tries to do so in natural language.

On the surface Google (and other search engines) and Wolfram Alpha share much functionality: Google can calculate "17mpg in liters per 100km" and Wolfram Alpha will handle non-computational queries such as "Norway" by giving a pre-edited informational answer. The difference lies more in what kind of data the two services work against, and how they determine what to show you: Google crawls the web, tracking links and monitoring user responses, in a sense asking every page and every user of their services what they think about all web pages (mostly, of course, we don't think anything about most of them, but in principle we do.) Wolfram Alpha works against a database of “curated” facts with a set of defined computational algorithms. It stores less and derives more. (That being said, they will both answer the question "what is the answer to life, the universe and everything" the same way....)

While the technical differences are important and interesting, the real difference between Wolfram Alpha and Google lies in what kind of questions they can answer—to use Clayton Christensen's concept, what kind of jobs you would hire them to do [7]. You would hire Google to figure out information, introduction, background and concepts—or to find that email you didn't bother filing away in the correct folder. You would hire Wolfram Alpha to answer precise questions and get the facts, rather than what the web collectively has decided is the facts.

As for user interaction, there are signs that Wolfram Alpha works the same way as Graffiti did: As Steven Wolfram says in his talk at the Berkman Center, people start out writing natural language queries but pretty quickly trim it down to just the key concepts (a process known in search technology circles as "anti-phrasing"). In other words, by dint of patience and experimentation, we (or, at least, some of us) will learn to write queries in a notation that Wolfram Alpha understands, much like our hands learned Graffiti on the Palm Pilot.

Step 3. Learning From User Behavior: Seeds, Not Links

Search engines have made us change our relationship to information, substituting search for categorization [8]: We no longer bother filing things away to remember them, for we can find them with a few keywords, relying on sophisticated query front-end processing and the fact that most of our not that great minds think depressingly alike. Some, predictably Nicholas Carr [9], worry that having all this information will reduce our ability to process more substantial thoughts.

As search engines become a more dominant interface to information, another change in user behavior is emerging: The use of search terms (or, if you will, seeds) as references. You frequently here people refer to search queries as navigation: "Google *x* and you will find it." Searching is fast. If you want to apply for a visa to the U.K., you can either go to the Home Office web page and click your way through various levels of departmentese, or you can enter "U.K. Visa" in Google and have the pertinent form as the top result. The risk in doing that, of course, is that somebody can externally manipulate the result, through search engine optimization or by creating online reference campaigns. General Motors, a few years ago, had an ad for a new Pontiac model, at the end of which they exhorted the audience to "Google 'Pontiac'" to find out more. Mazda quickly set up a web page with Pontiac in it, bought some keywords on Google, and drove their intended result high up on Google's list [10]. (Another issue with the use of search queries for information passing is the increasing personalization of search results, but that could be taken care of within the protocol, specifying that the receiver should see the same result as the sender.)

A curated data set of the type used by Wolfram Alpha will reliably return the same answer, albeit freshly calculated, every time. Should the answer change, it would be because the underlying data was changed (or, extremely rarely, because somebody figured out a new way and better way of doing the calculation.) It would not be because someone external to the company has figured out a way to game the system. This means that we can use references to Wolfram Alpha as shorthand. Enter "budget surplus" in Wolfram Alpha, and the results will stare you in the face (trustworthy, if depressing). In the sense that math is a language for expressing certain concepts in a very terse and precise language, Wolfram Alpha seeds will, I think, emerge as a notation for referring to factual information—the way Google searches is now. Over time, we will see web pages use Wolfram Alpha queries first as references, then as modules, then as dynamic elements.

Putting It All Together

The implications of these three evolutions—simple, evolving protocols based on human interfaces; curated data sets giving reliable, human-readable answers; and the use of search queries as information passing mechanisms—becomes quite important when put together. Together, these three notions can form alternative building blocks towards a meaning-driven, semantic web.

When computers—or, if you will, programs—needed to exchange information in the early days, they did it in a machine-efficient manner. Information was passed using shared memory addresses, hexadecimal codes, assembler instructions and other terse and efficient, but humanly unreadable encoding schemes. Sometime in the early 80s, computers were getting powerful enough that the exchanges gradually could be done in human-readable format. The SMTP protocol, for instance, a standard for exchanging email, could be read and even hand-built by humans (as I remember doing in 1985, to send email outside the company network.)

The World Wide Web, conceived in the early 90s and live to a wider audience in 1994, had at its core an addressing notation (the URL) which could be used as a general way of conversing between computers, no matter what their operating system or languages. (To the technology purists out there, yes, WWW relies on a whole slew of other standards as well, but I am trying to make a point here.) It was rather inefficient from a machine communication perspective, but very flexible and easy to understand for developers and users alike. Over time, it has been refined from pure exchange of information to the sophisticated exchanges needed to make sure it really is you when you log into your online bank—essentially by increasing the sophistication of the HTML markup language towards standards such as XML, where you can send over not just instructions and data but also extensions such as labels, definitions and to a certain extent relationships between information items.

The semantic web, as defined by Berners-Lee et al [11] is built on these technologies and is seen as the next logical step from a web based on linking via syntax to a web of meaning—a continuation of the move to programming further and further away from the metal, if you will. Through explicit organization of content, expressed in XML and defined in RDF and OWL, web pages have become nested layers of information containers, palimpsests on which content is continually updated. In the process, however, it has lost much of the simplicity that enabled the first World Wide Web—it is no longer possible to understand a web page by looking at its source, or to create it from scratch using a simple text editor. Apparent simplicity is facilitated by complex metadata addition.

Human requests for information from each other are imprecise but rely on shared understanding of what is going on, ability to interpret results in context, and a willingness to use many clues and requests for clarification to arrive at a desired result. Observe two humans interacting over the telephone—they can have deep and rich discussions, but as soon as the conversation involves computers, they default to slow and simple communication protocols: Spelling words out (sometimes using the international phonetic alphabet), going back and forth about where to apply mouse clicks and keystrokes, double-checking to avoid mistakes. We just aren't that good at communicating as computers—but can the computers eventually get good enough to communicate with us?

I think the solution lies in mutual adaptation, and the exchange of references to data and information in other terms than direct document addresses may just be the key to achieving that. Increases in performance and functionality of computers have always progressed in a punctuated equilibrium fashion, alternating between integrated and modular architectures. The first mainframes were integrated with simple terminal interfaces, which gave way to client-server architectures (exchanging SQL requests), which gave way to highly modular TCP/IP-based architectures (exchanging URLs), which may give way to mainframe-like semi-integrated data centers [12]. I think those data centers will exchange information at a higher semantic level than any of the others—and a terse, evolutionary interaction protocol, the notion of predictable responsiveness pioneered by Wolfram Alpha, and the use of queries as information passing mechanisms just might get us, if not to the Web 3.0, at least some way there.

References

1. Stephenson, N., *In the Beginning... Was the Command Line*. 1999, New York: Avon Books.
2. Turing, A.M., *Computer Machinery and Intelligence.* Mind, 1949. **59**(236): p. 433-460.
3. Kurzweil, R., *The Age of Spiritual Machines*. 1999, New York: Penguin.
4. Kurzweil, R., *The Singularity is Near: When Humans Transcend Biology*. 2005: Viking Adult.
5. Pepper, S. *The TAO of Topic Maps: Finding the Way in the Age of Infoglut*. in *XML Europe*. 2000.
6. MacKenzie, I.S. and S.X. Zhang. *The Immediate Usability of Graffiti*. in *Graphics Interface '97*. 1997.
7. Christensen, C.M., *The Rules of Innovation.* Technology Review, 2002 (June).
8. Andersen, E., *The Waning Importance of Categorization.* ACM Ubiquity, 2006. **7**(19).
9. Carr, N.G., *Is Google Making Us Stupid?* The Atlantic Monthly, 2008 (July/August).
10. Rangaswamy, A., C.L. Giles, and S. Seres, *A Strategic Perspective on Search Engines: Thought Candies for Practitioners and Researchers.* Journal of Interactive Marketing, 2009. **23**: p. 49-60.
11. Berners-Lee, T., J. Hendler, and O. Lassila, *The Sematic Web.* Scientific American, 2001(May 17).
12. Barroso, L.A. and U. Hölzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, in *Synthesis Lectures on Computer Architecture*, M.D. Hill, Editor. 2009, Morgan & Claypool.

About the Author

Espen Andersen is an Associate Editor at *Ubiquity* and is Associate Professor of Strategy with The Norwegian School of Management BI and European Research Director with The Concours Group.

DOI: 10.1145/1880066.1891342

