UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novatica** <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novatica**

UPGRADE was created in October 2000 by CEPIS and was first published by **Novatica** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for **UP**ENET (**UP**GRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- **Informatica**, journal from the Slovenian CEPIS society SDI
- **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
- **Novatica**, journal from the Spanish CEPIS society ATI
- **OCG Journal**, journal from the Austrian CEPIS society OCG
- **Pliroforiki**, journal from the Cyprus CEPIS society CCS
- **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS
- **Tölvumal**, journal from the Icelandic CEPIS society ISIP

ISSN 1684-5285

Monograph of next issue (April 2009)

"Geographic Information Systems"

(The full schedule of **UP**GRADE is available at our website)

CEPIS **UP**GRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org

Vol. X, issue No. 1, February 2009

Monograph: Universal, Ubiquitous and Intelligent Web
(published jointly with Novática*)
Guest Editors:
Klaus Birkenbihl, Encarna Quesada-Ruiz, and Pablo Priesca-Balbín

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Turning Semantics into Syntax

Bert Bos

The Web is a communication system with a combination of features that invites automatic processing, so that a reader does not simply read what an author wrote, but instead reads an enhanced version, which combines sources and displays information in a way that is adapted to the reader. For W3C, the potential of the Web lies in expressing information in formal languages that a computer can manipulate. Languages already exist or are being developed for such diverse aspects of communication as colours, sounds, geographic locations, or copyright licences. The limits have not yet been reached, but they do exist: the more precise the desired expression of meaning, the fewer people are capable of writing it. At some point, the computer has to deal with what is left implicit. In other words, the complement of the semantic Web is Artificial Intelligence. More practical issues also influence what languages can be standardized, in particular the shortage of resources.

Keywords: CSS, E-Government, EOT, HTML, Ontologies, Semantic Web, Standardization, Standards, W3C, Working Group.

Author

Bert Bos, was researching graphical user interfaces at the University of Groningen when he became co-inventor (with Håkon Wium Lie) of Cascading Style Sheets (CSS) in 1994. He joined W3C in 1995 to develop CSS and to lead W3C's Internationalization activity. He has since participated in the standardization of HTML, XML and, more recently, MathML. He has edited several (draft) standards related to CSS and MathML and has co-authored the book "*Cascading Style Sheets – designing for the Web*" (Addison-Wesley Professional, 2005).
<bert@w3.org>.

1 The Role of W3C Standards in Developing the Web

The Web is a communication system; it allows people to exchange information. Its features are found in other communication systems as well, but the combination of those features makes the Web unique. In particular, the Web shows how it is possible to use computers to enhance communication. And that is because the Web has given us ideas for how to turn semantics (i.e. meaning) into syntax. W3C has chosen the expression of meaning in formal languages as the primary means to make the Web more useful. Let me explain.

Like books, the Web allows a single author to reach many people, who do not all need to be reading at the same time. But publishing on the Web is faster and cheaper. Plus, the fact that the information is in electronic form allows easy adaptation to different environments (if the encoding of the information follows certain guidelines). Of course, there are environments where a book is easier to handle. On the other hand, it may be difficult to get the book in the first place, while access to some kind of network to connect to the Web is available increasingly more readily.

Like e-mail and e-mail mailing lists, the Web allows people to react to other people in a reasonably short time. Not as short as with face-to-face communication or the telephone, but without the time pressure: the information waits for the reader, it is not lost when the listener is unavailable at the moment the speaker speaks. Reacting is not as simple as e-mail, though, or at least not yet, because it requires write access to a Web server, either your own or somebody else's (the latter via wiki software, for example). On the other hand, it becomes easier to refer to older messages, because they have a unique address on a server. (Mailing lists can be, and often are, coupled with a Web archive for that purpose, of course).

Not that stability of information is guaranteed on the Web. Stability relies on people maintaining their data (apart from accidents, of course, such as computer failures, but in practice such accidents are rarely the reason for information disappearing). The technology on which the Web is mainly built (DNS, URLs, various protocols and formats) assumes that information stays stable for a certain time, so that it makes sense to make references, but stability is not built-in. Ted Nelson's original idea for Xanadu [1] included stability, in the sense that all earlier versions of information at a certain location remained available. The peer-to-peer system Freenet [2] also has a notion of stability: information remains bit-for-bit the same throughout its lifetime, and the lifetime does not depend on the publisher but on whether somebody somewhere is interested in the information.

On the Web, stability is still an unsolved problem. On the one hand, we do not want valuable information to disappear when the publisher is no longer able to maintain his computer (the Internet Archive [3] does its best to keep information online on its computers that no longer exists elsewhere); on the other hand we also want to be able to forget, especially privacy-sensitive information.

For the moment, the W3C position on stability is that it is a social problem (see Tim Berners-Lee's "Cool URIs don't change" [4]). As long as most information is identified by location (URL) rather than by a more indirect, abstract identifier (URN), there is not much more we can do. And we do not know yet how to make a sufficiently fast and scalable system based on URNs. Books, of course, use URNs: they refer to each other by author, title and year, sometimes ISBN.

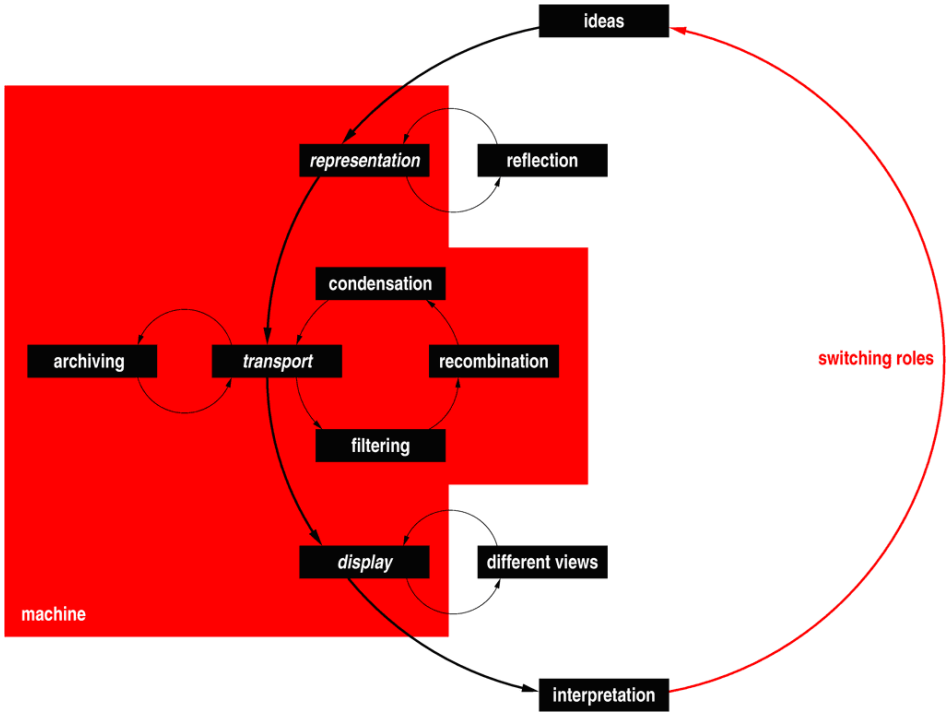

Figure 1: A Schematic Diagram of the Flow of Information on the Web[1].

To translate that identification into a location (building, room, shelf) takes time. The Freenet system mentioned earlier uses URNs; on that system information moves around to wherever it is needed and thus it does not have a fixed location. And despite research and experiments, it is still slow, unless the information happens to already be at a location near you ("near" in network terms). Quite different from the Web, where the reference is equal to the location and thus the information comes via the most direct link, in 3 seconds for a simple Web page, or a few seconds more for a complex one. Unless the information is no longer at that location, of course.

When used interactively the Web presents itself as a hypertext system. Unlike references in books, which require some effort (sometimes a considerable effort) to be followed, references in Web documents require no more than a simple click and the referent is downloaded in seconds. Only, of course, if the reference is to something else on the Web, but the popularity and size of the Web is such that there is a good chance that the linked information is indeed online. Interaction via hypertext is not unique to the Web, but the combination of hypertext with such a large and varied collection of information is.

The Web is a modular system, in the sense that it allows different modalities of information (text, sound, images, video…) to be transported in different ways (HTTP, FTP, peer-to-peer, cable, wireless…) and the various components of the system can to a large extent be enhanced or replaced individually, without the need to re-engineer the others.

Even HTTP, the most common transport protocol on the Web, is not irreplaceable. It has many useful features (content negotiation, caching, statelessness, optional authentication, etc.), but other protocols, such as some peer-to-peer protocols, have their advantages, too. As physical networks are getting faster (with fibreglass, e.g.), the size of information in bytes may become less of a problem, but peer-to-peer protocols can still help by shortening the distance between the information and the person asking for it. After all, no matter how large the bandwidth, the speed of light is finite and each request and answer sent over the ocean takes a third of a second, while sending it across town takes less than a tenth. If some information consists of many small pieces, those delays start to add up.

Interestingly enough, speed and stability appear to be related. Caching and peer-to-peer protocols can help speed up the Web only for stable information. Cookies are not a great help in that respect, and neither are session IDs, but the technology known as AJAX is probably one of the worst culprits. (Although what AJAX does to usability, both for people and intelligent agents, is probably an even bigger problem).

[1] An author (at the top) has an idea to communicate. He expresses his idea in digital form. He may use text, images, video, etc. and probably needs to read over and edit his work a few times before he is satisfied with the representation. That digital representation of the idea is transported to various places, where it can simply be stored for some time, or be integrated with other information. A reader (at the bottom) displays the digital information in some non-digital form and tries to make sense of it. He or she may try different ways of displaying the information: change the font size, print it on paper, extract just the interesting parts, etc. The reader may then want to react; become an author himself and insert his own ideas into the Web.

Text has a special place among the various modalities, because it is such an efficient way to represent human language and because it can be coded very efficiently in digital form. We also know better how to write programs for text than for other forms of information.

Nevertheless, whether text or something else, we still need to enhance information with annotations that are there for computers, not for other humans: our programs (or programmers) are not smart enough yet to recognize many implicit messages in our communications. We need to indicate keywords for search engines, mark-up text with markers for headings, lists, quotations and paragraphs. The spaces, commas and periods we are used to are not enough.

This extra information is what is usually referred to as the "Semantic Web," the digital information that allows various kinds of automatic processing to take place, that would not be possible if the text was just text and the images just images with nothing more than what a human would need to understand it.

Adding extra information requires effort. Not everybody is capable of it and it requires training. And the question is always if the effort is worth the benefit. Making our programs smarter is another solution, but that also requires effort, not to mention time-consuming research. To simplify a little, we might say that this latter solution is what Artificial Intelligence is about.

One example of where the effort of semantic coding is nearly always cost-effective is in separating a text into HTML and CSS parts. We encode the text and its structure (lists, headings, sections, special phrases, etc.) in HTML and the presentation aspects (colours, margins, fonts, or: in the case of audio: voice, volume, speed, etc.) in CSS. Not everybody is able to separate mentally presentation from structure, but people who can, often find that there is a benefit in doing so, not only for their readers, but for themselves as well. The benefit is mainly that text can be easily adapted to different devices (small screens, big screens, printers…), but also that it can be reused in a context where a different layout is required.

Seen this way, the Semantic Web is not a concrete goal that we will reach one day; it is a direction in which we want to go, just like "east". And *semantics* in this context is best defined, somewhat tongue-in-cheek, as *that information which we don't have a formal notation for yet.*

2 An Example: an Events Calendar

Everybody has his favourite examples of what the semantic Web ought to be able to do. Here is one of mine.

Imagine you live in the beautiful city of Nice. During the weekend, you want to enjoy its many offerings: listen to a concert, see a film, visit a flea market, go to a museum, etc. You visit some Web sites that show cultural events: the city's own agenda, the agenda of the opera house, the agenda of an international site specialized in dance performances, the culture section of the local newspaper…

The Web appears to work well. There are many sources of information and quite a lot of what is going on in town is announced online somewhere. Thanks to the HTML standard, all that info is at your fingertips and all you need is a Web browser.

But it takes time to read and understand everything. All those sites insist on their own layout and structure; you need to deal with different information hierarchies, different layouts, and probably even struggle through different ways of interactions, when some site's author decided to use JavaScript to make the site "easier" to use (but, unfortunately, rather different from other sites).

In this case, what you need is a common calendar format, in addition to HTML, so that the information can be easily compared by showing the various sources on a single calendar. iCalendar [5] is the Internet standard for that. hCalendar [6], the microformat [7], helps authors to manage calendar information, by letting them write just the HTML (following certain conventions) and have it automatically be made available as iCalendar when needed. (I could not immediately find an example in Nice, but I have examples for Besançon [8] and Birmingham [9]).

At the moment, in 2009, most events in Nice are available on the Web in HTML form, but only a small percentage in iCalendar (or hCalendar). This is no doubt partly because authors do not know it is possible or have not seen the benefits yet, but also because some authors do not *want* to make it easy to find information, for example because they want people to spend as much time as possible looking at the commercials on their Web site. But let us leave that discussion for another time and assume sufficiently many information providers *have* turned their semantics into syntax, i.e., into iCalender or hCalendar.

So now you have all events from all sites nicely sorted on a single calendar on your screen in front of you. Thanks to a common standard for some type of semantics, we have got rid of the layout and navigation issues and everything is now under a single program with a single interface. It is a lot easier to search through, and for a medium size city like Nice we may stop here, but maybe the computer can help us even more.

You would maybe like to filter and sort events on more criteria than just the date, such as the type of events or how easy it is to reach them. You might want to instruct your browsers with preferences such as that you are interested in subtitled films, but not in dubbed ones, or that you are interested in events less than 10 Km from your girlfriend's place.

That information is no doubt available by carefully reading the description of each event, although probably only indirectly (the address is given, but not the distance, because that depends on the reader). So we need additional standards, to make that additional information available in machine-readable form, too.

The location (address) of an event can already be made sufficiently machine-readable (using iCalendar or hCalendar) that a program can extract the location and try to compute the distance to another location, e.g., using some on-line route planner.

The type of event (concert, film, theatre, dance, exposition, or even more fine-grained categories) is not yet easily expressible. A standard, even a simple one, does not yet exist.

Of course, one can discuss endlessly what should be in such a taxonomy of events (just "film"? or "romantic film"? or "romantic Chinese period film"?), but even a simple one would be useful.

Maybe even then we are not yet done, because once all this is in place, you have a lot of information in your calendar and no doubt many events more than once: the same concert is listed on the site of the opera house and on the site of the orchestra. How do we weed out duplicates? Do we need unique identifiers for each event?

No, that is a step too far. It is easy enough for an author to mark up the type of event and the address. Most likely somebody received the information on a piece of paper and is typing it into the database that drives his Web site and which already distinguishes those fields anyway. But a global identifier for the event requires a global coordination. It is too much overhead. And then, why cannot the computer do as we do? We recognize duplicates easily enough, and if we fail to recognize a few, that is no problem either, just a few entries more in our calendar.

It seems that we have reached again the intersection between the semantic Web and Artificial Intelligence. The extra work an author has to do to make information machine-readable must be of a reasonable size compared to the expected benefit. In this case there is certainly a benefit to a globally unique number for each event, but the effort to make one is considerable and computers are already capable of recognizing duplicates reasonably well by themselves, so the ratio of cost to benefit is too high.

Let us go back for a moment to the question of taxonomies (a.k.a., vocabularies, or, sometimes, ontologies). Who makes those? W3C makes a number of standards that express some part of the meaning (semantics) of information: HTML, CSS, SVG, P3P, XML, etc. Other organizations make certain others (HTTP, Unicode, MIME, etc.) Who would make a simple (or complex!) categorization of cultural events?

It is not such a simple question to answer: making a standard requires bringing together a number of prerequisites: experts in the field, of course; a process and the supporting tools that help make sure the development goes smoothly and is fair to all parties; money, to pay for the time of the experts and the tools they use; authority and marketing, to make sure the result will become known and used; maintenance, in the sense of keeping the standard available for as long as needed and up to date if circumstances require modifications; and conformance tools, usually in the form of tests, to help implementers.

3 The Immediate Future

Cultural agendas are one application of the semantic Web and one where progress is being made towards making information re-usable, comparable, and adaptable to the user's needs. But there are others. A standards organization such as W3C is always on the look-out for concrete applications with common traits, because finding solutions for those common traits means that, firstly, there is an immediate use for those solutions and, secondly, that there is a good chance that the work will be re-usable for other, future applications.

The XML, and more recently, EXI technologies are good examples. XML captured the common aspects of many different types of SGML-based documentation systems, viz., that they contained structured, multilingual documents with both ordered and order-independent pieces; it solved part of the problem of exchanging documents between those systems. XML subsequently proved useful for many other kinds of information for which no standard for exchange existed yet, even many that had nothing to do with documents, maybe even contained no natural language text at all. EXI is doing the same for structured binary information, especially structured streaming data.

People are paying more and more attention to aspects of security, privacy and rights (such as copyrights) in electronic systems, including the Web. And once again organizations such as W3C are looking for the common traits in the various problems. Finding solutions may be a bit more difficult than for XML or calendars, because the definition of the problem itself is culturally defined. What is normal in one country is unthinkable in another (or, what is maybe even more difficult: is almost the same in another country, but not *quite*). Although whether that makes the problem more difficult to solve or easier is also a matter of point of view. In one case a solution can almost be derived mathematically, but mathematics is not everybody's cup of tea; in the other case you are dealing with people and so maybe you can change the people instead…

Various initiatives inside W3C indicate that there is a demand for progress in these areas and that it will probably happen in the near future: The Interest Group called PLING [10] was set up as a general forum for discussing all kinds of "policy languages," i.e., various ways of turning the semantics of rights into syntax. Various languages are already deployed somewhere, under study, or proposed. The WSC working group [11] is trying to establish the (initially small) set of security-related things that a browser should be able to tell a user about the information he is seeing and the service he is interacting with. A submission from Creative Commons [12] proposes a concrete language for expressing some subset of copyrights in machine-readable form. The Media Annotations working group [13] tries to define a common vocabulary for several aspects of video, including copyrights. And finally there is the proposal for a technology to embed fonts in Web documents [14], where "embed" actually refers to a particular kind of copyright licence that is very common among font vendors rather than to the act of physically putting one file inside another. (The font embedding proposal is interesting for other reasons, which have nothing to do with rights, or even with turning semantics into syntax; but more about that later).

Another development that we can expect to bear fruit soon is the harmonization of government data. W3C has an

eGovernment interest group [15], because governments realize that even government data has to cross national borders. In the European Union, for example, people move a lot between countries and so do business transactions. In many border areas, communities on both sides of the border have long discovered that it is more efficient to share services with nearby towns across the border rather than with far away ones on the same side, including hospitals, emergency services, public transport, culture, police, etc. (The oldest *Euroregion,* called Euregio [16], around Enschede on the Dutch-German border, is already 50 years old, although most are less than 15 years.) But the borders do pose problems, and some of them can be solved with standards for eGovernment.

4 The Practice of Standardization

So it seems we know where we want to go, the semantic Web, and although we are not sure how far we need to go, there is enough to do already. Why then do we not make more standards; why not create that format for embedded fonts, settle on one or two video formats, or develop a machine-readable vocabulary for medicines (another area where W3C has an interest group)?

Let us look at that standard for embedded fonts, for example. The problem to solve is that designers can specify to quite a large extent (within the limits of HTML, CSS, SVG, XSL and related standards) how a document looks, with respect to colours, layout, line heights, etc, but are very limited in the typefaces they can use. Technically, they can specify any arbitrary typefaces and even where to download the font file if it is not available on the reader's machine. But most fonts are under a licence that prohibits putting them up for download. The result is that designers can make a PDF document with almost any font they own, but not an HTML document, because there is no Web standard for embedding.

Sometimes, to solve a problem involves difficult technical research, training programmers on new techniques and a very long period of trial and error until software is good enough to handle those new techniques. CSS itself is an example of that, it took many years before programmers mastered the underlying model well enough, because for many of them it was different from what they were used to when handling documents. In particular, the notion (which HTML inherited from SGML and is now also part of XML) that a document could be represented as a tree of nested structures, rather than as a stream of state changes was quite alien to many programmers of early browsers.

But for embedded fonts technical difficulty is actually not a problem. Somebody at Microsoft (nobody seems to remember exactly who) had the idea about ten years ago what the equivalent of "embedding" is on the Web. In a traditional compound document, all pieces are contained in one large file and thus there is no question what document they are part of. Compound documents on the Web, on the other hand, consist of separate pieces downloaded individually: the text in HTML, some images in JPEG, a style sheet in CSS, etc. There is a link from the HTML to the style sheet, but not the other way round and thus if you start at the style sheet you are unaware what document it is part of. That is often an advantage and the Web exploits it with gusto. But if, as is the case with many fonts, the licence requires that a font may only be distributed as part of a document (viz., a document written by the licensee) and not on its own, then that model does not work. The clever solution is to make *two* links: a link back from the font indicates the document the font belongs to.

Microsoft implemented that in a format called EOT [17] (Embedded OpenType), derived from OpenType, an ISO standard for digital fonts. But EOT itself is a proprietary format, not a standard. CSS allows it to be used, but as nobody else than Microsoft could implement it, it never became very popular.

So we have at least two solutions to make a large number of fonts available for Web documents: change the practice of licensing so that fonts can be used without embedding, or standardize a technology that links a font to a document (or several documents at the same time, as EOT does, for more efficiency). The former is not likely to succeed in the short term, because nobody knows yet how font foundries will make money if they only sell each font once. An economic model that is not based on selling licences has yet to be invented. So we are left with the technical solution: make a font format with a link inside, just like Microsoft did with EOT, but make it a standard. (In fact, it could be EOT itself, Microsoft is willing to donate it, although with ten years of experience it is easy to improve on EOT.) Software, such as browsers, that displays or prints Web documents would simply look at the link to see if the font indeed belongs to the document and ignore it if not.

But nothing is happening yet. Font foundries have in majority said they like this solution. There is no DRM involved, no attempt to make it difficult to use a font, just this annotation (the link) that makes the licence sufficiently machine readable that software knows what it means; and foundries believe that is good enough (and fits the zeitgeist). Designers obviously want to use the fonts on the Web that they already bought for other documents (such as PDF). Software pirates have no objection either, because removing or changing the link is easy. There are of course still technical issues to research (are all forms of caching and proxying properly covered? how about documents with random URLs, such as session IDs? can you embed fonts in documents sent by e-mail?) but so far none seem particularly troublesome.

And still some browser makers have said they are not going to implement a solution like this, or at least not now. Why? There we touch on a number of practical problems of standardization.

There is the question of resources: there are only so many hours in a day, only so many programmers able to make certain things and only so many people that can work on the same project at the same time. Developing a standard for embedded fonts needs people: to discuss, write, implement and test the technology. Those people can then not work on other projects. And there are *many* other projects

going on at the same time. It is a matter of priorities: estimating the urgency of projects and the most efficient order for them.

And, as we are dealing to a great extent with projects that have commercial aspects, there is often also the politics of marketing to consider: for how long is it better to do something different from the competition instead of agreeing on a standard? what is the best time to announce support for a standard? is it possible to sell a solution twice, once as an incomplete solution and then again when it conforms to the standard? how long is it possible to delay the ratification of a standard so that competitors that have already invested in it cannot monetize their investments yet?

And thus it seems to go with embedded fonts, as well as with some other technologies that are good ideas, but nevertheless have to wait until their turn comes. We will see partial solutions from some implementers, attempts by those who have not yet mastered the technology to delay and change the standard enough that the investments of those that are ahead lose some of their value. A delay of two or more years is not rare.

A standards organization such as W3C has to deal with that, and have patience and enough vision to see the future despite the short-term politics. It also needs to have the hide of an elephant, because when things are slow to be deployed, people complain to the authors of the standard. W3C is especially subject to that, because it is well-known and open: anybody can join a W3C mailing list or send e-mail to the authors of a standard, because all documents are signed with the names and e-mail addresses of real people, often even with a photo. That is because the positive feedback and contributions outweigh the negative ones.

5 Conclusion

The success of the Web is to a large extent due to the fact that it allows users of information to adapt the information to their needs. They are not so dependent on the particular choices the author has made and can combine different sources and extract important information while ignoring the rest. It is the belief of W3C that the future lies in making information even more reusable.

Many authors could do a much better job than they do now and make it easier to use their information in ways they have not thought of, but even a little bit of HTML and CSS makes interesting applications possible. Apart from incorrect use of HTML, scripts and cookies are major hindrances to developing novel applications on the Web.

New standards are being deployed and developed that make it possible to turn even more semantics into syntax and allow the reader's computer to help him even more to make sense of the information on the Web.

The question remains, however, what the limits of syntax are: how much of his implicit meaning can a person make explicit? Those limits are different for different people and also depend on training. They depend on motivation as well: is the effort worth the benefit? It is clear, however, that much of the meaning of any message will remain implicit, hidden in representations that on the surface seem to express something else. The semantic Web will need contributions from Artificial Intelligence if computers are to help users find, catalogue, and compare certain kinds of information.

However, we are yet to reach the limits, and much can still be improved in the way we represent information digitally. More standards are needed.

But standards take time. People need to understand the problem, agree on a solution despite conflicting interests, and then other people need to learn the concepts behind the solution and get experience in applying them.

Some areas where we may see interesting new applications appear in the next two or three years are in the handling of calendars, location-based information, copyright management, security and digital typography.

References

[1] Wikipedia. Project Xanadu. <http://en.wikipedia.org/wiki/Project_Xanadu>.

[2] The Free Network Project. <http://freenetproject.org/>.

[3] The Internet Archive. <http://www.archive.org/ >.

[4] Tim Berners-Lee. "Cool URIs don't change". <http://www.w3.org/Provider/Style/URI>.

[5] F. Dawson, D. Stenerson. Internet Calendaring and Scheduling Core Object Specification (iCalendar). RFC 2445. <http://tools.ietf.org/html/rfc2445>.

[6] Tantek Çelik, Brian Suda. hCalendar. Microformats Specification. <http://microformats.org/wiki/hcalendar>.

[7] Microformats. <http://microformats.org/>.

[8] *Le portail officiele du Grand Besançon*. Besançon (France). <http://www.besancon.fr/index.php?p=44&art_id=141>.

[9] What's on – All events. Birmingham (UK). <http://www.thsh.co.uk/events>.

[10] W3C Policy Languages Interest Group (PLING). <http://www.w3.org/Policy/pling/wiki/Main_Page>.

[11] W3C. Web Security Context Working Group. <http://www.w3.org/2006/WSC/>.

[12] W3C Member Submission Request: ccREL - the Creative commons Rights expression Language. <http://www.w3.org/Submission/2008/02/>.

[13] W3C. Media Annotations Working Group. <http://www.w3.org/2008/WebVideo/Annotations/>.

[14] W3C. Fonts Activity Proposal. <http://www.w3.org/2008/09/fonts-activity-v2>.

[15] W3C. eGovernment at W3C. <http://www.w3.org/2007/eGov/>.

[16] EUREGIO - Living without borders. <http://www.euregio.de/cms/publish/content/showpage.asp?themeid=23>.

[17] W3C. Embedded OpenType (.EOT) Font Format Submission Request to W3C. <http://www.w3.org/Submission/2008/01/>.

