Monograph of next issue (April 2009)

"Geographic Information Systems"

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

CEPIS **UPGRADE**
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 (**UP**GRADE **European NETwork**)

CEPIS NEWS

The Web: What a Building!

Klaus Birkenbihl

The article gives an overview on Web standards and technologies with emphasis on those developed at W3C. Selections of these are subjects of research and development described in this magazine. This introduction outlines the context of use and the purpose of the different technologies and illustrates relationships and dependencies among them.

Keywords: Accessibility, Mobile Web, Privacy, Security, Semantic Web, Voice Communication, W3C, Web Applications, Web Services.

1 Introduction

It's 17 years since the World Wide Web first appeared in the wild. And it's changed a lot since then. It still is changing in its technology, the services it provides and the way that they are used. Meanwhile it holds so many options that a unique view on what it offers might be very difficult. So the editors decided to pick some developments that they consider mainstream to give an impression of what is about to come.

- Web everywhere - the Mobile Web.
- Web for everyone - the Accessible Web.
- Web of structured information - the Semantic Web.
- Web of participation and communication - Collections, Blogs and Forums.
- Web as a service - Rich Web Applications.
- Web for business - Web Services.
- Web for media - Video and Voice on the Web.

All these should rely on a common set of technologies and standards. Functionalities of the different aspects should interoperate.

Some other topics are relevant for most of these aspects. They are horizontal by nature and some are subject to different cultural views.

- Privacy on the Web.
- Security and trust on the Web.
- Treatment of intellectual property.
- Content classification and rating.
- Accessibility of the Web.

We will not cover here the differing views on subjects like privacy, intellectual property and content classification. These are still subject of political and social discussions. Web technologies should approach whatever is seen beneficial in a given social and political context.

Most of the work on standards and technologies described in this paper is or was conducted at the W3C (The World Wide Web Consortium) which was launched 1994 by Web inventor Tim Berners-Lee.

2 The Basement

Three technologies used in combination made up the success of the Web:

- A common syntax for documents that may contain

Author

Klaus Birkenbihl is CEO of his company ICT Media GmbH and coordinator of W3C's international Offices program. Klaus holds a degree in Mathematics from the University of Bonn and started his career at the GMD National Research Lab for Information Technology. Later he was with the Institute for Media Communication of Fraunhofer-Gesellschaft. Klaus was a teacher for Computer Science at the University of Cologne and the University of Applied Sciences Bonn-Rhein-Sieg. Klaus started his activities in computer networking in 1983 when he was manager of GMD's Computing Center in Bonn and joined the European Academic Research Network (EARN, 1983) as a founding member. He acted as Deputy Director of EARN Germany (1987-1990), and was a member of the Member of the EASInet steering committee. Klaus was Treasurer and Chairman of the German Chapter of the Internet Society where he is still on the supervisory board. He launched the W3C Germany Office in 1997 and was on W3C's advisory board 2002-2005. After Klaus left Fraunhofer in 2005 he launched his own company and started to work for the W3C team.
<Klaus.Birkenbihl@ict-Media.de>.

hyperlinks (HTML).

- A syntax for locating documents (or fragments of documents) and the option to use this syntax within HTML to build hyperlinks (URL).
- A protocol to send request and responses between clients (usually browsers) and servers (HTTP).

These technologies allowed the spanning of a global Web of documents that, from a user perspective, provided a graphical user interface and a very easy way to jump from one document to a related one using a link provided in the first one.

This Web of documents quickly became a victim of its success. There was a huge amount of documents and the desire to integrate information from databases. Shop offers, flight schedules, etc. required extensions of the concept. In the short term it was scripting (server side and client side) and CGI that allowed for dynamic Web pages. But there was also a need for more structured and systematic solutions. So a generalization and extension of the three original technologies had to be made. In terms of structuring documents and data, two concepts were introduced: The XML Infoset and RDF Graphs.

The XML Infoset is a generalization of the HTML con-

cept (while being stricter on the syntax). As HTML, XML structures a document or other entities in a tree with named (tagged) nodes. HTML uses a predefined set of node names and rules to restrict their use within a document. The XML Infoset holds no such restrictions. Introducing specific restrictions can be used to form new data structures or markup languages. In fact thousands of those XML languages exist today for a huge variety of purposes. A few assistive technologies came with XML as there are:

■ XML Namespaces as a means to separate one space of tags from another.

■ XML Schemas to allow the definition of restrictions to specify a new XML language.

■ XQuery/Xpath as a means to navigate through or query XML documents.

■ XML Base to cope with relative URLs.

■ XPointer to allow reference to XML documents and fragments.

■ DOM an abstract Interface to process XML.

XML relies on the concept of defining complex entities by composing them with simpler ones. RDF Graphs try to define entities by describing their relationship to others. An RDF graph is composed of *triples* of the form *subject predicate object*. Where *subject* (a resource) denotes something you want to define, *predicate* defines the relationship (property), and *object* (a resource) is something for which the relationship holds. *Objects* (and even *predicates*) in one triple can appear as *subjects* in others. So we may see huge networks of relations. Properties that have their origin in mathematics or logic (like "is a subset of", "implies"…) lay the ground for complex queries and reasoning on such networks. As XML RDF comes with a set of associated technologies:

■ RDFS is a Schema language that defines a vocabulary that gives meaning to some resources (like *Class*, *Literal*, *Datatype*…) and properties (like *subClassOf*, *subPropertyOf*, *domain*, *range*…) This opens a standardized way to express set or logic related properties. OWL supplements and extends these definitions.

■ SPARQL is a query language for RDF

Though XML and RDF follow different concepts (hierarchical and relational) in structuring information, they are not completely separated from each other. RDF/XML is a special XML language to represent (serialize) RDF Graphs, admittedly not the most human-readable one. Basic data types used in RDF are from XML Schema.

Beside the technologies introduced here there is a collection of so called Web Architecture Principles that constitute rules on how to use these technologies to build new ones.

3 Cornerstones

We want the Web to be universal. This is a noble aim with quite a few implications. In the end it means that everybody should be able to use the Web with a wide range of devices in any language he knows. It is quite obvious that this has its limitations (e.g. the author of these lines will probably never be able to understand an article on microbi-

ology in Chinese). This is no surprise. The surprising thing is how much universality can be obtained by following a few guidelines. The areas for which these guidelines are developed are: accessibility, device independence, internationalization.

W3C is the developer of some of these guidelines. In the end there are a few general rules that are refined and complemented:

■ Use standards wherever they exist.

■ Add redundancy in the way you present your information.

■ Present your structure in a way that it can be recognized.

■ Separate content from presentation.

There is an overlap in the technical requirements to achieve this. A page that separates content from presentation can be more easily adapted to various devices but also to accessibility needs. A picture of a building in a document accompanied by a text might help a visually impaired person to understand it and it also may help somebody who had never seen the building before to identify it or enable a search engine to find it. Texts reach more people by adding redundancy (by providing them in multiple languages for instance). Using standardized character sets helps automatic machine translation by providing a unique representation for symbols and characters.

But the implications go even beyond the subjects mentioned here. E.g. a Web site that is accessible is usually more efficiently searchable. (Or as Karsten M. Self stated in Arachnophobia: "*Google is, for all intents, a blind user. A billionaire blind user with tens of millions of friends, all of whom hang on his every word. I suspect Google will have a stronger impact than the ADA in building accessible Web sites*"). In general: designing for accessibility, device independence, and internationalization (as a very welcome side effect) significantly contributes to the usability and the value of a site.

4 Supporting Pillars

Having all these technologies and knowledge what should we expect from the Web? Every technology is there to make our life easier. It should reduce the effort we have to spend, to reach our goals. As information technology in general the Web offers a broad spectrum of application areas. Other than information technology the Web offers these applications networked on a global scale. Broadband connections and nearly unlimited storage and computing power, together with interoperable technologies, open the field for multimedia and huge databases. Business models rise and fall, killer applications pop-up. Social networking, privacy, sharing knowledge and intelligence, managing intellectual property, participation in decision processes, marketing of ideas and products, the importance of the *long tail*... all had to be rethought or became only relevant because of the Web.

We see a convergence of Web technology with mobile phone technology. Digital mobile phones appeared in public during the first half of the 90s, at the same time as the

Web. Since then the growth of this market even outperforms the growth of the Internet. Mobile phones play a crucial role for the economy of many developing countries. Today every mobile phone can access the Web. So both technologies converge.

On the other hand there are some things that remain a mess. Searching can easily become a nightmare. Phishing and the abuse of browser vulnerabilities threaten users. Automatic translation is primitive when trying to access information in languages the user does not speak. So there is still a lot to be fixed and researched and a lot of technological, social and political decisions to be made.

A classification of the technologies that are available or under development might be by the following topics:

- Web applications.
- Semantic Web.
- Web Services.
- Mobile Web.
- Voice Communication.
- Privacy and security.

4.1 Web Applications

Plain HTML is all you need to build a Web application. At the other end of the spectrum are fully fledged applications e.g. for office work or picture processing. Video, audio, sophisticated graphics, presentational Software for mathematics or chemistry can also be included. Plugins and Interpreters for programming languages allow almost all content or processing that works on a desktop to be used within a Web page. Quite a few technologies are available to support this. Some are standardized, some are proprietary. Some are "open" (e.g. by providing Open Source) some are "locked".

Some very important technologies and standards in this context are provided by the W3C, free of charge and open for everyone. Among these standards HTML is the one most widely used. It is also the standard that provides the facility to embed objects that follow other standards e.g. to do graphics and video, images. For good or ill there are two HTMLs that are under standardization by the W3C today: HTML 5 and XHTML 2. This might not be the place to go into details. At the risk of oversimplifying one might characterize these two directions as follows:

- The main focus of XHTML 2 is to better use the advantages of XML within HTML. At the same time one wants to get rid of constructs (elements) whose functionality might better be implemented in cleaner and more general way. Allowing integration of other technologies or requirements (compound documents, semantic Web, accessibility) is also a design goal.

- HTML 5 is more concerned with advancing the existing HTML (HTML 4 or XHTML 1). It is not necessarily reliant on XML, though it supports its use, but upward compatibility from earlier versions of HTML is very important.

Of course there is quite some overlap in goals and principles of both HTMLs but the differences in their design goals are significant.

In order to allow separation of the presentation from the content of HTML or XML documents, a technology named CSS was developed. It provides flexible means to define for every element how it is to be presented within the document. CSS is a very powerful technology that allows Web designers to adopt their pages to a wide range of devices or window sizes. A more general approach is by XSL. XSL allows not only the formatting of existing documents, but also transformation of their content. So the elements of a document can be rearranged, compacted or extended.

In order to allow the reader of a Web page to reply (e.g. to place an order in a job or to register for an event) HTML documents may contain forms. Forms are available as part of the HTML standard. Much more powerful (and properly integrated in the XML world) is XFORMS that can be used with any XML language.

A very powerful XML language for graphics is SVG. It allows vector based two dimensional graphics which, other than pixel based images, can be scaled up and down without losing quality. SVG allows also vector based animations. Many modern browsers support SVG or a subset of it.

As was said before (X)HTML allows to include non (X)HTML objects. These are either referred by a link or directly included, often as XML documents with another namespace (e.g. SVG or the mathematical mark-up language MathML). This should normally work well unless there is supposed to be some user interaction or synchronization of animations. There is series of compound document standards in development (WICD, CDF...) that will allow better integration of parts of a document using different technologies.

The Web, especially when combined with powerful programming languages and interfaces, can act as a platform for sophisticated applications. So we see a lot of applets, widgets and applications that rely on Web technologies. Games, maps, image processing or complete office suites can be based on Browsers using technologies like Compound Document Formats, XML, the DOM (and DOM events), XFORMS, XHTML, SVG and others.

In order to build presentations with media from various sources in a synchronized manner SMIL might be used. SMIL was published in 1998 and is an XML language that allows arrangement of media in time and plane.

4.2 Semantic Web

Finding things that are on the Web is still one of the big challenges. There are a lot of search engines around: text based, based on subject catalogues and specialized engines like lowest price finder, etc. Nevertheless finding e.g. all music events in Berlin that take place in April next year might be a challenge if you can't find a specialized database. The reason for this is that Web documents usually do not tell "this is about an event that takes place in April" in a way that is understood by search engines. A human reader can usually easily deduce this when he reads the page. Not so a search engine. It finds the word "April" but has no idea

that this might be the month in which an event takes place, it does not even know the page is about an event, nor what an "event" might be. The same applies to the word "music".

In order to solve this we have to "teach" computers. This is done by defining relationships between the things we are talking about. We have (to find ways) to tell computers that music is a performing art. That a performing art is an art. That everything of type music has a duration. That music can be a live performance. That a live performance is an event. That every event has a start time etc. Secondly we need a way to let computers know that a certain piece of data is e.g. a start time. And we have to be able to specify the event this start time belongs to.

We mentioned above that RDF is a language that allows us to talk about relations. But in order for it to be usable for computers we also need a unique way to name *properties* and *resources*. Here the concept of the URL comes handy. The URL constitutes a way to uniquely identify Web documents. But there is no technical reason to constrain the mechanism of URLs to Web pages. By removing this restriction we obtain the URI which is used by RDF to uniquely identify *resources* and *properties*.

Properties are *resources* too. So we can make statements about *properties*. E.g. we could define that a *property* "isSubClassOf" allows us to conclude that if "car isSubClassOf vehicles" and "myCar isMemberOf car" then "myCar isMemberOf vehicles". This allows the deduction of new statements from existing ones by applying logic. There are different RDF vocabularies (and rules) defining the meaning of such properties. Starting with the simplest RDFS there are (today) three levels under OWL standardized: OWL Lite, OWL DL and OWL Full. Each level offers different options for specifying and concluding relations. There are algorithms that help to find such conclusions (and, irritatingly, it is not sure that algorithms processing some OWL Full will ever finish).

Having these technologies, what does it mean for the Web? We could imagine an "overlay" Web where e.g. every start date of an event identifies itself as start date, refers to the event it belongs to, which identifies itself as being a performance, and identifies that it is music... There is a standard named RDFa which allows including RDF statements in an XHTML document.

Besides searching, another use of linked data has emerged in the Web: the mash-up. Though today often based on proprietary APIs the idea behind it is to use data from one source and merge it with some information from another source. Using Semantic Web technologies to link the data would ease the construction and maintenance of mash-ups a lot.

Another example of using linked Data is the FOAF project. It provides a RDF vocabulary to describe a person. Part of the description can be Data about who this person knows, which can link to or contain another FOAF description. Some open social Networks are based on FOAF.

Beside RDF another technology is used to link data on the Web: Microformats. These use the "class" attribute of (X)HTML to denote that an HTML element contains some

data. In order to be interoperable, Microformats have to be administered centrally. Microformat specifications are available from microformats.org.

4.3 Web Services

Web Services are less about browsing and more about the interoperation of software applications that run on computers interconnected by a network. A set of standards help to achieve this interoperability and to couple applications in loose way to achieve complex operations. Web Services play a central role especially in e-business. The opportunity to pass data from one business step to the next using an XML data format appears very attractive and has the potential to automate parts of the business process. So we saw a lot of concepts and buzzwords (EAI, ESB, MOM, SOA ...) in this area that in the end are based on Web Services. Several requirements have to be met in order to get Web Services working:

- You need a protocol to exchange messages between the applications.
- You need to be able to address services and messages.
- You need to describe services in a machine readable way.
- You may want to apply policies to control the use of services.

There are a few standards that lay the ground for Web Services based on these requirements:

- SOAP specifies the format of messages that can be used to communicate with a service.
- WS-Addressing describes how to address Web Services and messages.
- WSDL is a language to describe a service by the kind of messages it will receive or send and its functionality. There is also a standard for Semantic Annotations to WSDL (SAWSDL) which allows the use of ontologies to deceive the semantics of Web Services.
- A means to specify policies for the use of a service can be specified and attached to a service by using the Web Services Policy specifications.

4.4 Mobile Web

Mobile use of the Web is characterized by devices with limited capabilities (e.g. mobile phones) in terms of connectivity, bandwidth and processing power of the client. So Mobile Web standards are about making browsing possible and a pleasant experience when using mobile devices. Mobile Web standards focus on two aspects:

- Downsizing of existing standards to match the constraints of mobile devices.
- The development of guidelines especially for authors (but for authoring-tool manufacturers as well) on how to make sure that a Web site is mobile friendly.
- A way for servers to learn about the capabilities of an attached client.

XHTML Basic, SVG Tiny, SVG Basic, SMIL mobile, XFORMS Basic and CSS mobile are standards that include

those modules of the respective technologies that mobile devices should provide. In order to tailor the content delivery of a server to the capabilities of a mobile device (or the browser used on the device) the server should be able to learn about the capabilities of the device. The DDR simple API deceives an interface that servers can use to query device description repositories that hold this information.

A second aspect of mobility is that location awareness might be an advantage for the user. Many mobile devices have some build-in capability to locate their position on the globe. In order to use this information for services to the user, work for a Geolocation API specification has been started.

4.5 Voice Communication

One might wonder about the minimum requirements to access the Web. Would a plain old telephone work? Sometimes? Quite some information is disseminated or communicated by phone. So it is quite reasonable to ask how this could be supported by Web technologies. Everyone who used a computer based phone service before knows that, with respect, there is space for improving the user experience. In order to pave the road for better (semi-) automated voice communications, a set of standards was developed that covers the following areas:

■ VoiceXML is about defining frameworks for dialogues. It is a mark-up language to collect information from a user. The flow of the dialogue is controlled by the VoiceXML mark-up depending on the answers of the user. VoiceXML is an XML language with scripting capabilities.

■ Two speech oriented standards support the use of VoiceXML. The SSML is an XML based language to assist the generation of synthetic speech. SRGS allows the specification of grammars for recognition of speech.

CCXML provides support for telephony call control and can be used in conjunction with dialogue systems such as VoiceXML. EMMA is an XML language to describe input received on several channels (voice, handwriting, keyboard ...) simultaneously.

4.6 Privacy and Security

The Web has changed our lives in many ways. One of the most ambivalent changes is in our privacy. While using the Web we disclose a lot of our personal data, interests and preferences. Every question we ask, every Web site we visit can be registered. It is hard for the user to guess how this information will be used. In order to protect their citizens governments have introduced privacy laws. But the scope of legislation usually ends at the national border which limits its effect in the Internet. *Sine die* this problem cannot be resolved by technical means only. People have to handle their private data with care and socially acceptable and technically implementable new models for privacy have to be found.

P3P is a standard for specifying a machine readable description of a privacy policy. It e.g. allows companies to publish their privacy policies in a way that a user agent (e.g. as part of a Web browser) can understand. Since there is no way to enforce the policy, P3P is useful in cases where the user can trust the site that issues a policy. This is usually the assumption when doing business. Some browsers (e.g. Internet Explorer 6+) use P3P information provided by servers to allow users to control the use of cookies.

There are instances we trust. But can we be sure that the Web site that pretends to be the home page of our bank is really our bank? Can we securely transfer confidential data over the Web? Can your bank make sure that a money transfer from your account was really issued by you? That it was not modified on the way?

These problems are not specific to the Web. Mathematics (especially number theory) has provided us with the instrument of encryption that solves many problems in this area. The internet protocols provide with TLS/SSL the means to apply these instruments to Internet protocols. There are some things that obviously still have to be resolved to apply this technology on the document or data level. XML Encryption, XML Signature and XML Key Management are three related technologies that allow application of well known cryptographic methods on fragments of XML documents e.g. to digitally sign a part of the body of a SOAP message.

There is ongoing work on how to evaluate or inform users about the security context of an environment they are using. In the end this work is about making better use of certificates, encryption, Web site dynamics and other information provided by the server. This should enable users to better understand the security context of a Web site when making trust decisions.

Though there is a lot of progress in this area there are still quite some unresolved issues with respect to privacy and security.

5 The Roof: Web Sites and Killer Applications

All these technologies and standards are rather a means to an end than ends in themselves and the list is far from being complete. Together with steadily increasing bandwidth and computing power they make the web a place where people use their creativity for doing business, conduct discussions, disseminate ideas or build relationships. During the last 17 years the Web has transformed the Internet from a geek's toy to a powerful infrastructure that is formed by millions of individuals, business people, journalists, artists, scientists... Google, Yahoo, Flickr, Ebay, Youtube, Myspace, Wikipedia, Mozilla name only a few popular projects that live in this space.

But there are still a lot of open questions that wait for ideas, technical solutions, standards or a social agreement to be answered. In parallel to the work on millions of Web sites, Blogs, Wikis, shops and fancy applications there is ongoing work on better standards and technologies. The W3C provides a framework where vendors, authors and users cooperate to improve the supporting frame of the Web.

Abbreviations

ADA: Americans with Disabilities Act.
API: Application Programming Interface.
CCXML: (Voice Browser) Call Control XML.
CDF: Compound Document Formats.
CGI: Common Gateway Interface.
CSS: Cascading Style Sheets.
DDR: Device Description Repositories.
DOM: Document Object Model.
EAI: Enterprise Application Integration.
EMMA: Extensible MultiModal Annotation markup language.
ESB: Enterprise Service Bus.
FOAF: Friend of a Friend.
HTML: Hypertext Markup Language.
HTTP: Hypertext Transfer Protocol.
MathML: Mathematical Markup Language.
MOM: Message Oriented Middleware.
OWL: Web Ontology Language.
P3P: Platform for Privacy Preferences.
RDF: Resource Description Framework.
RDF/XML: RDF in XML Syntax.
RDFS: RDF Schema.
RDFa: RDF in attributes.
SAWSDL: Semantic Annotations for WSDL.
SMIL: Synchronized Multimedia Integration Language.
SOA: Service-Oriented Architecture.
SOAP: (originally for) Simple Object Access Protocol
(long version is misleading and not used anymore).
SPARQL: Protocol and RDF Query Language.
SRGS: Speech Recognition Grammar Specification.
SSML: Speech Synthesis Markup Language.
SVG: Scalable Vector Graphics.
TLS/SSL: Transport Layer Security / Secure Sockets Layer.
URI: Uniform Resource Identifier.
URL: Uniform Resource Locator.
VoiceXML: Voice Extensible Markup Language.
W3C: World Wide Web Consortium.
WICD: Web Integration Compound Document.
WSDL: Web Service Description Language.
XFORMS: XML application that represents the next
generation of forms.
XHTML: Extensible Hypertext Markup Language.
XML: Extensible Markup Language.

References

The W3C Technology Stack.
<http://www.w3.org/2004/10/RecsFigure.png>.
About W3C Technology.
<http://www.w3.org/Consortium/technology>.
W3C Technical Reports and Publications esp. all Standards.
<http://www.w3.org/TR/>.
About W3C Activities.
<http://www.w3.org/Consortium/activities>.