WEB 2.0: A BUZZWORD, A SERIOUS DEVELOPMENT, JUST FUN, OR WHAT?

Gottfried Vossen
European Research Center for Information Systems (ERCIS), University of Münster
Leonardo-Campus 3, 48161 Münster, Germany
vossen@uni-muenster.de

Keywords: Web 2.0, Rich Internet Applications, Mash-ups, User-Generated Content, Software-as-a-Service.

Abstract: Since 2006, everything has its Release 2, Version 0, abbreviated “2.0,” e.g., Silicon Valley 2.0, Family 2.0, Gadgets 2.0, Pub 2.0, Jobs 2.0, Health 2.0, Entertainment 2.0, Business 2.0, Music 2.0, Enterprise 2.0, Marketing 2.0, Law 2.0, Education 2.0, etc. All of this is a consequence of the term “Web 2.0” which was created at O’Reilly Media. But what is behind this shorthand: Is it just a buzzword, does it stand for a serious development, is it just for fun, or what? This paper discusses the core developments that have led to what is currently termed “Web 2.0”: net infrastructure, advances in programming, rich interactive functionality, large data collections, and user participation as well as socialization. For all of these, there are roots in the “earlier” Web, but only their recent confluence has resulted in the hype we are currently experiencing. The paper looks at Web 2.0 from the perspective of a user, outlines the impact this is having on enterprises, and finally mentions some open issues and future directions.

1 INTRODUCTION

Since 2006, everything has its Release 2, Version 0, abbreviated “2.0,” e.g., Silicon Valley 2.0, Family 2.0, Gadgets 2.0, Pub 2.0, Jobs 2.0, Health 2.0, Entertainment 2.0, Business 2.0, Music 2.0, Enterprise 2.0, Marketing 2.0, Law 2.0, Education 2.0, etc. All of this is a consequence of the term “Web 2.0” which was created at O’Reilly Media several years ago. But what is behind this shorthand: Is it just a buzzword, does it stand for a serious development, is it just for fun, or what? This paper discusses the core developments that have led to what is currently termed “Web 2.0”: net infrastructure, advances in programming, rich interactive functionality, large data collections, and user participation as well as socialization. For all of these, there are roots in the “earlier” Web, some of which are even more than 10 years old, but only their recent confluence has resulted in the hype we are currently experiencing. The paper not only looks at Web 2.0 from the perspective of a user having new forms of fun and enjoying a host of new services, but also outlines the impact this is having on enterprises, and mentions legal questions that are often overlooked. Finally, it points out some research issues that have come along in this context.

The paper is organized as follows: Section 2 explores what we consider the core Web 2.0 dimensions, thereby answering the question of whether we are dealing with a buzzword; here we vastly follow (Vossen and Hagemann, 2007). Next, Section 3 looks at the impact Web 2.0 is having on enterprises and investigates whether it is a serious development. Then Section 4 surveys miscellaneous issues related to Web 2.0, finding out whether it is just fun. Finally, Section 5 draws a few conclusions.

2 WEB 2.0 DIMENSIONS

Figure 1 summarizes what we consider the four major dimensions that together make Web 2.0. These are the infrastructure dimension, the functionality dimension, the data dimension, and the social (or socialization) dimension, explained in more detail in the following subsections. Note that (Musser & O’Reilly, 2007) see a total of eight different driving factors for Web 2.0, which of course overlaps with those mentioned here.
2.1 Net Infrastructure

The first dimension that is fundamental for what we are experiencing as Web 2.0 is the net infrastructure and includes both hardware and software technology. Indeed, with broadband Internet and wireless technology available as a commodity especially in the developed world, we are seeing a host of new applications and services arise and delivered over the Internet and the Web, with digital radio and television only being precursors of what is there to come. Broadband communication in particular allows for an easy transfer of large files, so that, for example, it becomes possible to watch movies over the Internet on a mobile device.

According to (Friedman, 2005), the world as of today has become a flat one in which people from opposite ends of the planet can interact, play, do business with each other, and collaborate, and of that without knowing each other or having met, and where companies can pursue their business in any part of the world depending on what suits their goals and intentions best; they can also look at an entire world of customer base. There are essentially no more serious limits to what anyone can accomplish in the world these days, since the infrastructure we can rely upon and the organizational frameworks within which we can move allow for so many unconventional and innovative ways of communicating, working together, collaborating, and information exchange.

Although most of this is essentially “invisible” since it takes place in cables or in the air, present-day technology also materializes in the form of data centers that companies like Google, Amazon, and others set up around the world. Data centers are large collections of hardware (including servers, storage, and communication equipment, power supply, air conditioning) that allow for distributed processing, yet also for renting out computing power or storage space) thereby enabling cloud computing, see below).

On the software side of technology, software development has dropped in prices, since what has been an expensive service some time ago is now cheap, and the “LAMP” (a common acronym for Linux, Apache, MySQL, PHP) manifestation of the open-source world has brought along a considerable number of tools through which software development is supported today.

The Web itself has always been based on HTML as a language, and an HTML document is allowed to have scripting code embedded for making it dynamic and interactive. An important distinction refers to the question of whether scripting occurs at the client side or at the server side. Client-side scripting, very often seen in the form of JavaScript, makes use of the fact that a browser can not only render HTML pages, but also execute programs. These programs, written in a script language, are interpreted just like HTML code in general. Thus, some of the tasks arising in a Web site can be off-loaded onto the client. On the other hand, certain things cannot be done at the client side, in particular when access to a database on the Web is needed. With server-side scripting using, for example, PHP or Java Server Faces, user requests are fulfilled by running a script directly on the Web server to generate dynamic HTML pages; it can be used to provide interactive Web sites that interface to databases or other data stores as well as local or external sources, with the primary advantage being the ability to customize the response based on a user's requirements, access rights, or query results returned by a database.

Both client-side and server-side scripting is based on the classical client/server paradigm and on the fact that any such interaction is assumed to be synchronous. More recently, asynchronous processing of data allows sending data directly to the
server for processing and receiving responses from the server without the need to reload an entire page, which has led to a further separation of user interface logic from business logic now known under the acronym Ajax (Asynchronous JavaScript and XML). Ajax (Crane et al., 2006) is a Web development technique that allows developers to build rich Web applications that are similar in functionality to classical desktop applications, yet they run in a Web browser. Its main functionality stems from an exploitation of XMLHttpRequest, a JavaScript class (with specific properties and methods) supported by most browsers which allows HTTP requests to be sent from inside JavaScript code. Immediate implications of these developments will be seen next.

2.2 Novel Functionality

The second dimension is a direct consequence of what has just been said about Ajax and other scripting techniques, namely that it allows a number of applications to run directly in the browser. A common example is Google Apps, which includes Gmail, the Google mail application, Google Talk for chatting and sending instant messages, Google Calendar, a tool for scheduling events and meetings and for sharing these with others, Page Creator for designing Web pages, Start Page for creating a personalized browser start page, and Google Docs & Spreadsheet, a Web-based text processing and spreadsheet program that allows several people to work on documents or spreadsheet calculations together (see www.google.com/a for further details).

Clearly, advantages and disadvantages stem from the applications being available over the net. While the applications and their data can be accessed from anywhere on the Web, this holds currently most often only as long as a connection exists. Users do not have to worry about program installation or updates, but the security of the Web browser becomes an even more important issue.

However, not everything that we now see in terms of new functionality needs to be browser-based. Feature-rich applications independent of a browser have been around for many years. Indeed, every desktop program supports functionality implemented with a manifold of interaction patterns, including menus, movable windows, dialog boxes, drag and drop, or immediate responses. Rich applications have been available offline for a long time already; the novelty we are nowadays seeing is that many such applications have moved online, and many new rich applications are being launched directly on the Web. Rich Internet Applications (RIAs for short) are function-oriented, as opposed to mash-ups, which are more data-oriented, as they typically combine data from different sources. RIAs come in a variety of forms and range from being more playful interfaces to Web pages to being a full incarnation of the software as a service (SaaS) idea that provides entire application systems over the Web.

SaaS today comes in two major forms: for developers in the form of mash-ups or Web Procedure Calls (WPCs), our term for calls to an API over the Web, for end users (companies as well as individuals) in the form of RIAs or widget collections which either are for free or can be purchased or rented. An SaaS provider is essentially interested in offering a service which clients can use on-demand and for which he can charge a fixed or pay-as-you-go fee. An SaaS consumer, on the other hand, may be interested in specifying parameters in advance which the service should meet as well as in the customization of a service, or the consumer may simply be interested in putting a service together himself, i.e., without any specific prescription from a provider. When a software service is provided over the Web, such as in the case of the Google Maps API, the provider is essentially fixing the ways in which the service can be used. The API for the service typically consists of a number of procedures that can be called or invoked, yet the services are fixed and have to be used as is. We distinguish application services and infrastructure services below.

We mention that the SaaS model may not just be considered entirely advantageous. For example, a service lock-in may arise due to the fact that some business organizes all its processes around a certain service provider, thereby becoming heavily dependent on this provider. This can even refer to such trivial issues as email addresses following the syntax of a particular ISP; if the company switches to another ISP, all these addresses are likely to change. On a larger scale, a service provider may exploit lock-ins in various ways, such as to start charging for an initially free service after some time. In addition, there may be contractual hazards that need to be considered. Other potential problems are that the service provider may be down at unpredictable times or may even go out of business. The questions that need to be looked into here are not new, yet may easily be overlooked in this particular context. A solution already offered by various providers consists of service-level agreements (SLAs) which clarify issues such as reaction times, availability, or performance.

The business model of application services, where some software functionality is provided that a customer uses as an application, is typically the subscription model: The service might be free, but more common is a subscription or a pay-as-you-go
Table 1: A sample of modern SaaS applications.

<table>
<thead>
<tr>
<th>Application</th>
<th>Company</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizer and Calendar</td>
<td>Google Calendar</td>
<td><a href="http://www.google.com/a">www.google.com/a</a></td>
</tr>
<tr>
<td></td>
<td>37signals Backpack</td>
<td><a href="http://www.backpackit.com">www.backpackit.com</a></td>
</tr>
<tr>
<td></td>
<td>30 Boxes</td>
<td><a href="http://www.30boxes.com">www.30boxes.com</a></td>
</tr>
<tr>
<td></td>
<td>Remember The Milk</td>
<td><a href="http://www.rememberthemilk.com">www.rememberthemilk.com</a></td>
</tr>
<tr>
<td>Planning</td>
<td>PlanHQ</td>
<td><a href="http://www.planhq.com">www.planhq.com</a></td>
</tr>
<tr>
<td>Word Processing and Spreadsheets</td>
<td>EditGrid</td>
<td><a href="http://www.editgrid.com">www.editgrid.com</a></td>
</tr>
<tr>
<td></td>
<td>ThinkFree</td>
<td><a href="http://www.thinkfree.com">www.thinkfree.com</a></td>
</tr>
<tr>
<td></td>
<td>Google Docs &amp; Spreadsheets</td>
<td>docs.google.com</td>
</tr>
<tr>
<td></td>
<td>Num Sum</td>
<td><a href="http://www.numsum.com">www.numsum.com</a></td>
</tr>
<tr>
<td></td>
<td>Zoho Docs, Zoho Writer</td>
<td><a href="http://www.zoho.com">www.zoho.com</a></td>
</tr>
<tr>
<td>Database Apps (DaaS – Database-as-a-Service)</td>
<td>Smallthought Systems</td>
<td><a href="http://www.dabbledb.com">www.dabbledb.com</a></td>
</tr>
<tr>
<td></td>
<td>Dabble DB</td>
<td>aws.amazon.com/simpledb/</td>
</tr>
<tr>
<td>Project Management, Collaboration,</td>
<td>37signals Basecamp</td>
<td><a href="http://www.basecamphq.com">www.basecamphq.com</a></td>
</tr>
<tr>
<td>Shared Workspaces, Conferencing</td>
<td>CentralDesktop</td>
<td><a href="http://www.centraldesktop.com">www.centraldesktop.com</a></td>
</tr>
<tr>
<td></td>
<td>WebEx</td>
<td><a href="http://www.webex.com">www.webex.com</a></td>
</tr>
<tr>
<td></td>
<td>GoToMeeting</td>
<td><a href="http://www.gotomeeting.com">www.gotomeeting.com</a></td>
</tr>
<tr>
<td></td>
<td>activeCollab</td>
<td><a href="http://www.activecollab.com">www.activecollab.com</a></td>
</tr>
<tr>
<td></td>
<td>TeamWork Live</td>
<td><a href="http://www.teamworklive.com">www.teamworklive.com</a></td>
</tr>
<tr>
<td></td>
<td>Zimbra Collaboration Suite</td>
<td><a href="http://www.zimbra.com">www.zimbra.com</a></td>
</tr>
<tr>
<td>Accounting and Payroll</td>
<td>Xero</td>
<td><a href="http://www.xero.com">www.xero.com</a></td>
</tr>
<tr>
<td></td>
<td>Intacct</td>
<td><a href="http://www.intacct.com">www.intacct.com</a></td>
</tr>
<tr>
<td>Customer Relationship Management (CRM)</td>
<td>Etelos CRM for Google Apps</td>
<td><a href="http://www.crmforgoogle.com">www.crmforgoogle.com</a></td>
</tr>
<tr>
<td></td>
<td>salesforce.com</td>
<td><a href="http://www.salesforce.com">www.salesforce.com</a></td>
</tr>
<tr>
<td></td>
<td>RightNow</td>
<td><a href="http://www.rightnow.com">www.rightnow.com</a></td>
</tr>
<tr>
<td></td>
<td>Zoho CRM</td>
<td><a href="http://www.zoho.com">www.zoho.com</a></td>
</tr>
</tbody>
</table>

Thus, using the service is what clients have to pay for, yet there is essentially no need for an initial investment in hardware or software, except for the cases where the application provided is integrated into other applications at the client’s side. As an example, consider the Google Apps suite of tools: All components as well as the required data storage run on Google servers. The current price is US$ 50 per user account per year, while basic services with, for example, reduced storage space for the mailbox remain free of charge.

Services like Google Apps exist in a variety that is enriched almost daily, and examples range from individual service calls to entire complex applications, from individual office functions to planning, collaboration, accounting, and CRM tools (see Table 1); even databases can be set up as a service over the Web these days (“database-as-a-service, or DaaS). A practical consequence of this increasing availability of software on the Web is that many start-up companies, at least in the beginning, do not rent office space anymore, but work in virtual teams.

While there is a clearly visible trend to make applications that formerly were exclusively running on a desktop computer now available as a service on the Web, there is another trend that tries to combine desktop and Webtop software: Future browser versions are expected to comprise support for offline applications. In other words, a user will be enabled to use his or her Web applications such as Google Calendar in the browser even when he or she is offline.

There is also a category of infrastructure services that are being offered over the Web and that commonly rely on the availability of a provider’s data center. Hence, services in this category provide storage or computing power or both, but also add other types of functionality. As a prominent example in this category, we mention the Amazon Web Services (see aws.amazon.com). Amazon offers roughly a dozen different Web services, which are genuine infrastructure services in the sense that they enable product development and are not primarily intended for individual users.

2.3 The Data Dimension

The third dimension is about data and has two sides: data that is (automatically) collected or generated, and data that is intentionally created by a user. In the former case, the data is typically collected by a machine, e.g., a Web server maintaining a log of all clicks users have been exercising, or the index...
commonly built and maintained by a search engine. Data in the second category include everything that users nowadays are willing to write on the Web. Examples start with email, bookmarks, tags on photos and articles, and continue to registrations, online diaries on the form of blogs, evaluations, comments, recommendations, and online documents such as wikis. Blogs are typically expressions of personal or professional opinion or experience which other people can at most comment; wikis are pages or systems of pages describing content that other people can directly edit and hence extend, update, modify, or delete. Both forms of communication have become very popular in recent years and contribute significantly to the novel read/write nature of the Web today. In other words, users have started to use the Web as a medium in which they can easily and freely express themselves, and by doing so online they can reach a high number of other people most of which they will not even know.

There are numerous forms of usage for (ideally large) data collections, both good and bad. E-commerce sites like Amazon have developed a number of techniques for exploiting, in particular mining log data and click paths (Liu, 2007) as well as explicit user feedback to create recommendations, and search engines with a well-maintained index have proved very helpful. Moreover, context-dependent advertising such as Google AdWords has turned out to be an extremely efficient way of bringing ads to those people who might actually be interested in the respective goods or services, and has at the same time become a major source of revenue for many Web companies. On the other hand, job applicants nowadays must be afraid that their future employer will have checked them out on sites like Facebook and MySpace even before the first job interview, and fraud by, for example, stealing bank account information has become a significant and frequent crime these days. Many other examples could be mentioned.

Yet another form of exploiting data is the already mentioned mash-up. A mash-up essentially brings multiple services or content sources together in order to form a new, often valued-added service and can also be seen, as mentioned, as a form of SaaS, since it often provides a service; a host of examples (around 4,000 at the time of writing) including mapping, micro-blogging, photos, music, jobs, news or search can be found at www.programmableweb.com/mashups.

2.4 Socialization

The fourth and final dimension is socialization, which is the idea of taking software or even user-generated content and sharing or jointly using it with others. Thus, socialization covers the aspect of user-generated content as it occurs in blogs or wikis, in tagging as well as in social bookmarking. Yet it is also covers “social software,” which is software that gets better (or at least more useful) the more people use it. While most of the time the software itself, i.e., the program system, does not change based on the number of its users or the frequency with which it is used, it is the application that the software is enabling which is actually getting better. Examples include Skype, the eBay seller evaluation, the Amazon recommendation service, or Wikipedia. Especially the latter is a perfect example for what so-called mass collaboration (Tapscott and Williams, 2006) or crowdsourcing can achieve.

As has been demonstrated by Flickr and others, there is also another impact that socialization can have, namely that of improving some given software on a constant or perpetual basis. Traditionally, software has never been free of bugs, security holes, or errors, and it has been common for a software company to fix them and distribute new releases or versions of the software from time to time. The new approach is to do this at a much higher pace. Software on the Web may nowadays be in a permanent beta state of release and never finished. Thus, for outsiders maintenance occurs on a permanent basis. Such a state of perpetual beta may apply to a service that can only be accessed through an API, in which case a user is not bothered by constant release changes, at least as long as the behavior of the API is only extended, but not fundamentally modified.

Online social networks, another form of mass socialization today, bring a dimension to the Web that goes beyond simple links between pages; they add links between people and between communities. In such a network, direct links will typically point to our closest friends and colleagues, indirect links lead to the friends of a friend (“FOAF”), and so on. A social network on the Web is typically the result of employing some software that is intended to focus on building an online community for a specific purpose. Social networks connect people with common interests and may be as simple as a blog, or as complex as Facebook or MySpace for mostly private applications, as LinkedIn or Xing for professional applications, or as Twitter for both.

The primary impact that the current Web developments are having in this area are that connecting people and communities constantly becomes easier, and it is not difficult anymore to maintain a professional or personal network of buddies worldwide. Yet another impact is that a social network may open up novel sources of revenue, in particular through advertising.
We conclude this section by mentioning that two aspects should have become clear by the discussion so far: On the one hand, the most obvious change that has recently occurred on the Web is that it has changed from a pure read Web as designed by (Berners-Lee, 2000) to a read/write Web, where users not only draw information from, but also add information to it. On the other hand, the dimensions we have discussed exhibit various overlaps. Indeed, technology enables functionality, which as a “byproduct” leads to data collections, and users have a new tendency to socialize over the Web, by exploiting that functionality and the technology. Thus, “Web 2.0” is not just a buzzword, but is it a confluence of various developments that have grown over the past 10-15 years, which suddenly have the right enablers and produce joint effects, many of which follow the “long tail” observation originally made by (Anderson, 2006).

3 ENTERPRISE 2.0

In this section we look at the impact Web 2.0 is having on enterprises. What we have described in the previous section may seem to be mostly targeted at the individual user, but the point is that all dimensions and Web 2.0 aspects are on their path into the enterprise. Indeed, a careful inspection of the listings in Table 1 shows that the primary target for SaaS applications today is businesses.

Typical business applications of Web 2.0 techniques and technologies include corporate blogs and wikis, feeds, and podcasts. Blogs are a common way to stay in touch with customers, to inform about new products and to receive immediate feedback; they can also be used internally in order to discuss specific topics among the staff of an enterprise, in particular if people are geographically distributed. Blogs allow a moderated interaction between participants, be it customers or colleagues, and a simple and efficient distribution of announcements, experiences, opinions, reports, or evaluations. However, bloggers need to keep in mind that blogs are typically crawled by search engines, so that company internals written into a blog might reach the outside world and be presented as search results. Also, a blog is useless without regular updates, a reasonable number of readers, continuous moderation, and good content. It is also a good idea for a company to treat independent bloggers just like regular journalists, since they might have a considerable readership.

In order to stay up-to-date with a company blog, but also with other information an enterprise might publish, there are essentially two approaches: pull and push. The active or pull way is to read the information at my own liberty and pace; in the passive or push approach, the information will be delivered to me automatically. Indeed, blog entries and other sources can be subscribed via feeds that are based on protocols such as RSS or Atom (Johnson, 2006), and they can be read using a feed reader such as Bloglines, Google Reader, Newsgator, or NewsAlloy (or simply in the browser). A podcast is a particular form of feed consisting of audio or video material.

Wikis allow collaborative work on a common set of documents by many authors, and have been discovered as a new way of performing knowledge management in a learning organization. If staff members can be motivated to participate in the development of a wiki, this can be considered a good example of making implicit knowledge explicit, thereby attacking the core problem of knowledge management.

A social network can also act as a means of connecting employees of distinct expertise across departments and company branches and help them build profiles in an easy way, and it can do so in a much cheaper and more flexible way than traditional knowledge management systems. Once a profile has been set up and published within the network, others can search for people with particular knowledge or expertise and connect to them. If the social network is to be run outside an enterprise, providers like Ning allow an easy setup of a self-regulated and self-managed community.

In summary, a broad usage of Web 2.0 techniques and tools within an enterprise, paired with an increased exploitation of services offered over the Web and with leaving more room for the individual and its preferences, has led to the term “enterprise 2.0.” In other words, Web 2.0 turns out to be a serious development indeed, and enterprises as well as software vendors are exploiting it by integrating Web 2.0 features into their software, processes, and work environments.

4 MISCELLANEOUS

We now look at a few miscellaneous issues related to Web 2.0, including legal as well as research issues. To begin with, a number of questions need to be asked today in connection with the information kept on the Web, including, but not limited to the following:

- Is some “big brother” watching us?
is a folksonomy can be used to enhance infrastructure. Using it, the same community process Firefox extension with supporting serverMoreover, they present the how these types support programmatic uses. based on their structure and language, and discuss shortcomings. They identify several types of tags the form of mash-ups and have highlighted their browsing. (Hagemann and Vossen, 2009a) have difficult to use them for more than searching and well; however, the ambiguity of tags makes it information access. For these uses tags work pretty good portion of awareness of what you are doing tags with programmatic meaning. Users are enabled to create reliable mash-ups based on tags. Effectively, this leads to customized views of Web pages with tagged content. ActiveTags naturally increases the usability of social tagging systems and further extends the notion of user-generated content. As shown by (Hagemann and Vossen, 2009b), it turns out that these mash-up extensions can be reformulated as meta-querying, i.e., queries towards databases that may contain queries as values. This gives a generalized view on client-side mash-up provisioning which shows the robustness of traditional relational technology and can guide the engineering of Web information systems. Social network analysis, or the analysis of the “social graph” that extends around a person participating in a social network, is the subject of several recent investigations. For example, (Kleinberg, 2008) investigates the convergence of technological networks that connect computers and social networks that link humans. (Kossinets et al., 2008) study the temporal dynamics of communication in e-mail communication among faculty and staff of a large university over a two-year period. (Danescu-Niculescu-Mizil et al., 2008) develop a framework for analyzing and modeling opinion evaluation, using a large-scale collection of Amazon book reviews as underlying dataset. So to answer our third questions, we may say that Web 2.0 is definitely fun, but it is a lot more than that. There is room for numerous new business ideas (such as the payment service offered by RevolutionMoney, the mail service offered by eSnailer, the flight service offered by Virgin Charter, or the personalized TV service from Current.com, to name just a few) as well as research questions, and a good portion of awareness of what you are doing and consuming is apt.

5 CONCLUSIONS

Web 2.0 is the confluence of a variety of developments. As we have tried to point out in this short overview, it is not something entirely new, but it is generally accepted as the current “version” of the Web, characterized in particular by user participation is a variety of ways. Compared to ten years ago, we can observe that, indeed, a number of activities and applications have entirely moved to the Web and can hardly be imagined to be done otherwise. In fact, “Web 2.0” is so common nowadays in almost everything that is happening on the Web that people will soon stop mentioning “2.0” explicitly.
Clearly, this is not the end of it. From a technological point of view, a major current trend is mobility, i.e., to make all of Web 2.0 usable from mobile devices, in particular smart phones. Another direction of development is the combination of data as it is stored now on the Web with semantic information about, for example, a context or a location. Indeed, the “Semantic Web,” where machines cannot just read, but understand information and act correspondingly, is seen by many as “Web 3.0.” Cloud computing is under investigation from a number of views, with issues including inter-cloud communication and portability, metadata needed to be moved with production data for protection purposes, or intra-cloud security and privacy. Finally, the exploration of Web 2.0 from a conceptual perspective has only just begun, but is picking up momentum, as the fast increasing number of relevant papers at the annual WWW conference (and other events) clearly shows.

REFERENCES


Johnson, D., 2006. RSS and Atom in Action – Web 2.0 Building Blocks, Greenwich, CT.


BRIEF BIOGRAPHY

Gottfried Vossen is a Professor of Computer Science in the Department of Information Systems and a Director of European Research Center for Information Systems at the University of Muenster in Germany. He received his master’s and Ph.D. degrees as well as the German habilitation in 1981, 1986, and 1990, resp., all from the Technical University of Aachen in Germany. He has held visiting positions at the University of California in San Diego, at several German universities including the Hasso-Plattner-Institute for Software Systems Engineering in Potsdam near Berlin, at Karlstad University in Sweden and at The University of Waikato in Hamilton, New Zealand in 2003 and again in 2006. In 2004 he became the European Editor-in-Chief of Elsevier’s Information Systems - An International Journal, and a Director of the European Research Center for Information Systems (ERCIS) in Muenster. His research interests include conceptual as well as application-oriented problems concerning databases, information systems, electronic learning, the Semantic Web as well as Web 2.0. Dr. Vossen has been member in numerous program committees of international conferences and workshops. He is an author or co-author of more than 150 publications, and an author, co-author, or co-editor of more than 20 books on databases, business process modeling, the Web, e-commerce, and computer architecture. His most recent book Unleashing Web 2.0 – From Concepts to Creativity centers around the topic of this talk.