A survey of the World Wide Web evolution with respect to security issues

Recently, we hear more about web generations and its role in current web technologies we are using. Most of people know Web 2.0 and how the huge transformation changed from the previous version (Web 1.0). Web 2.0 is the style that became standard in the late 1990s and includes all the features that have allowed web pages to move beyond static documents. Web 2.0 marked a cultural shift in how web pages were developed, designed, and used from static era to dynamic one. It saw the meteoric rise of social media, including Facebook and Twitter, and user-generated content such as blogs, wikis, Wikipedia being perhaps the most famous and video-sharing sites such as YouTube. Its features made it very attractive for people to be familiar with it and learn to work with it. In this paper, we will go through some aspects of Web Generations from 1.0 to 3.0 and focus on some security issues for each generation.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2793v1 | CC BY 4.0 Open Access | rec: 8 Feb 2017, publ: 8 Feb 2017

A Survey of the World Wide Web Evolution with Respect to Security Issues

Cheng Lee
Catholic University
Chenglee@dr.com

Yi Chen
Catholic University
Chen.Yi@dr.com

***Abstract*— Recently, we hear more about web generations and its role in current web technologies we are using. Most of people know Web 2.0 and how the huge transformation changed from the previous version (Web 1.0). Web 2.0 is the style that became standard in the late 1990s and includes all the features that have allowed web pages to move beyond static documents. Web 2.0 marked a cultural shift in how web pages were developed, designed, and used from static era to dynamic one. It saw the meteoric rise of social media, including Facebook and Twitter, and user-generated content such as blogs, wikis, Wikipedia being perhaps the most famous and video-sharing sites such as YouTube. Its features made it very attractive for people to be familiar with it and learn to work with it. In this paper, we will go through some aspects of Web Generations from 1.0 to 3.0 and focus on some security issues for each generation.**

Index Terms—Web 2.0, Web 3.0, Web security.

I. Introduction

Web 1.0 often consisted of static HTML pages that were updated rarely, if at all [2]. The success of the dot-com era depended on a more dynamic Web (sometimes labelled Web 1.5) where content management systems served dynamic HTML pages created on the fly from a content database that could more easily be changed [3]. In both senses, so-called eyeballing was considered intrinsic to the Web experience, thus making page hits and visual aesthetics important factors.

Proponents of Web 2.0 believe that Web usage is increasingly oriented toward interaction and rudimentary social networks, which can serve content that exploits network effects with or without creating a visual, interactive Web page[5]. In one view, Web 2.0 sites act more as points of presence, or user-dependent portals, than as traditional Web sites.

The term "Web 2.0" refers to what some see as a second phase of development of the Web including its architecture and its applications [6]. As used by its proponents, the phrase refers to one or more of the following:

- The transition of Web sites from isolated information silos to sources of content and functionality, thus becoming a computing platform serving web applications to end users [3].
- A social phenomenon referring to an approach to creating and distributing Web content itself, characterised by open communication, decentralisation of authority, freedom to share and re-use, and "the market as a conversation".
- A more organised and categorised content, with a far more developed deep-linking Web architecture.
- A shift in economic value of the Web, possibly surpassing that of the dot com boom of the late 1990s.
- A marketing term to differentiate new Web businesses from those of the dot com boom, which due to the bust now seem discredited.

However, a consensus on its exact meaning has not yet been reached. Many find it easiest to define Web 2.0 by associating it with companies or products that embody its principles. Some of the more well known Web 2.0 entities are Google Maps, Flickr, del.icio.us, digg, and Technorati.

Many recently developed concepts and technologies are seen as contributing to Web 2.0 including Weblogs, Wikis, Podcasts, RSS feeds and other forms of many to many publishing; social software, Web APIs, Web standards, Ajax and others [6].

Proponents of the Web 2.0 concept say that it differs from early Web development, retroactively labelled Web 1.0, in that it is a move away from static Web sites, the use of search engines, and surfing from one Web site to the next, to a more dynamic and interactive Web [14]. Others argue that the original and fundamental concepts of the Web are not actually being superseded. Sceptics argue that the term is little more than a buzzword, or that it means whatever its proponents want it to mean in order to convince their customers, investors and the media that they are creating something fundamentally

new, rather than continuing to develop and use well-established technologies [10].

Therefore : WEB 1.0 was about connecting computers and making technology more efficient for computers and WEB 2.0 is about connecting people and making technology efficient for people.

II. WEB 2.0

Web 2.0 is definitely the next big thing in the World Wide Web. It makes use of latest technologies and concepts in order to make the user experience more interactive, useful and interconnecting. It has brought yet another way to interconnect the world by means of collecting information and allowing it to be shared affectively. It definitely has a bright future with so many Web 2.0 based websites coming up [10]. It is a revolution in the field of computers and will definitely achieve far greater success in the near future than it already has.

A. Transformation from Web1.0 to Web 2.0

Table I shows some popular examples of transformation of Web 1.0 based sites to Web 2.0 based sites:

TABLE I: WEB1.0 and WEB2.0 Transformation[3].

Web 1.0	**Web 2.0**
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	BitTorrent
mp3.com	Napster
Britannica Online	Wikipedia
personal websites	blogging
evite	upcoming.org and EVDB
domain name speculation	search engine optimization
page views	cost per click
screen scraping	web services
publishing	participation
content management systems	wikis
directories (taxonomy)	tagging ("folksonomy")
stickiness	syndication

Web communication protocols are a key element of the Web 2.0 infrastructure. Two major ones are REST and SOAP[20].

- REST (Representational State Transfer) indicates a way to access and manipulate data on a server using the HTTP verbs GET, POST, PUT, and DELETE.
- SOAP involves POSTing XML messages and requests to a server that may contain quite complex, but pre-defined, instructions for it to follow[16].

In both cases, access to the service is defined by an API. Often this API is specific to the server, but standard Web Service APIs (for example, for posting to a Blog) are also widely used. Most, but not all, communications with Web Services involve some form of XML (Extensible Markup Language).

Recently, a concept known as AJAX has evolved that can improve the user experience in some browser-based Web applications. It involves a Web page requesting an update for some part of its content, and altering that part in the browser, without refreshing the whole page at the same time. There are proprietary implementations (as in Google Maps) and open forms that can utilise Web Service APIs, syndication feeds or even screen scraping [16].

Another relevant standard is WSDL (Web Services Description Language), which is the standard way of publishing a SOAP API.

III. WHAT IS WEB 2.0?

Web 2.0 involves using the Web as a platform to deliver information which is often built via mass community contribution. Wikis and blogs are good examples of these types of applications. The main attribute of a Web 2.0 application is interactivity [3].

What this actually means in reality is that more functionality has been placed on the client-side of the equation, and less on the server, which in turn allows a request to be updated directly in the browser without needing to refresh the entire page. The perfect example of this is Google Maps. Instead of a static page, a user can drill down, or zoom in and out, on the map without having to make a request for a new page [20].

There are several key technologies (or more appropriately, groupings of different technologies implemented together to increase functionality) that can create a Web 2.0 application. Some of the most heavily implemented of these follow.

A. AJAX

Asynchronous JavaScript combined with XML, is used to increase a Web application's interactivity, responsiveness, and usability by exchanging small bits of data with the server so the entire page does not need to be refreshed each time the user makes a new request.

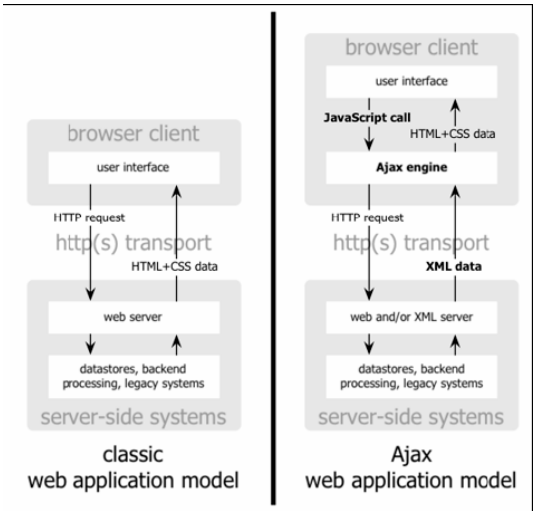

Figure 1: AJAX vs Classic Web Application Architecture [16].

B. *RSS*

most often called Really Simple Syndication or Rich Site Summary, is a collection of "feed" formats used to publish frequently updated content such as news or blogs. The initials "RSS" are variously used to refer to the following standards:

- Really Simple Syndication (RSS 2.0)
- Rich Site Summary (RSS 0.91, RSS 1.0)
- RDF Site Summary (RSS 0.9 and 1.0)

RSS formats are specified in XML (a generic specification for data formats). RSS delivers its information as an XML file called an "RSS feed," "webfeed," "RSS stream," or "RSS channel"[10].

C. *JSON*

JavaScript Object Notation, is used in conjunction with Javascript much in the same way XML is used in AJAX.

D. *Flash*

Flash is a very popular method for adding video and interactivity to Web sites. Most browsers offer support for Flash, and contain a client-side application to run Flash files [18].

E. *SOAP*

Simple Object Access Protocol, is used by most Web services to send XML data between the Web service and the client Web application making the information request[10].

F. *REST*

Representational State Transfer is used to increase a web application's response time and server loading characteristics via support for caching. For example, most blog sites are REST based (as opposed to RPC,or Remote Procedure Call), since they download an XML RSS feed file that contains links to other resources [12].

IV. Rich Internet Applications

Rich Internet applications (RIA) are web applications that have the features and functionality of traditional desktop applications. RIAs typically transfer the processing necessary for the user interface to the web client but keep the bulk of the data back on the application server[12].

RIA's typically:

- run in a web browser, or do not require software installation
- run locally in a secure environment called a sandbox
- can be "occasionally connected" wandering in and out of hotspots or from office to office.

A. *CSS*

In computing, Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can be applied to any kind of XML document, including SVG and XUL [16].

B. *XHTML*

The *Extensible HyperText Markup Language*, or XHTML, is a markup language that has the same depth of expression as HTML, but a stricter syntax.

C. *TAG*

A tag is a (relevant) keyword or term associated with or assigned to a piece of information (like picture, article, or video clip), thus describing the item and enabling keywordbased classification of information it is applied to.

Tags are usually chosen informally and personally by the author/creator or the consumer of the item —i.e. not usually as part of some formally defined classification scheme. Typically, an item will have one or more tags associated with it.

D. *WIKI*

A wiki is a website that allows visitors to add, remove, edit and change content, typically without the need for registration. It also allows for linking among any number of pages. This ease of interaction and operation makes a wiki an effective tool for mass collaborative authoring. The term wiki also can refer to the collaborative software itself (wiki engine) that facilitates the operation of such a site, or to certain specific wiki sites, including the computer science site (the original wiki) *WikiWikiWeb* and online encyclopedias such as *Wikipedia[13]*.

E. *WEBLOG*

A blog is a usergenerated website where entries are made in journal style and displayed in a reverse chronological order.

Blogs provide commentary or news on a particular subject, such as food, politics, or local news. A typical blog combines text, images, and links to other blogs, web pages, and other media related to its topic. The ability for readers to leave comments in an interactive format is an important part of most early blogs [20].

Most blogs are primarily textual although some focus on photographs (photoblog), sketchblog, videos (vlog), or audio (podcasting), and are part of a wider network of social media.

F. Podcasts

Another fashionable tool associate with Web 2.0 is Podcasting, which is simply making audio files (most commonly in MP3 format) available online so that users can then download them to their desktop media player like itunes and Windows Media Player etc) then listen to them whenever they want. To do this users need a podcatcher, a piece of software that allows you to download podcast episodes via a RSS feed [10].

G. Social Networking

Currently the fourth most popular website in the English speaking world, MySpace allows users to set up interactive and personalised web profiles detailing personal information like; education, age, interests, and hobbies. After users sign up for a free MySpace account they are able to edit and customise their profile page[12]. They can also chose to display friends, upload photographs, videos, music, create a blog, post comments on other user profile pages, and send messages to other users.

Implemented together or separately, these technologies have greatly increased the flexibility of Web applications.

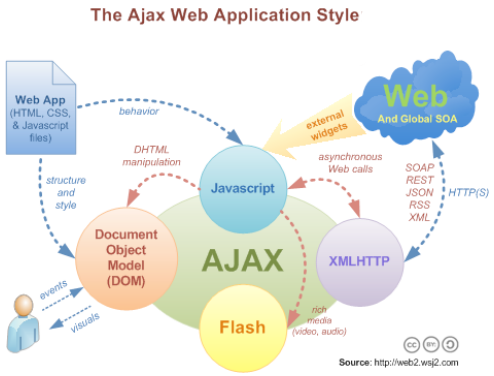

Figure 2: Ajax Web Application Style

A Web site could be said to be built using Web 2.0 technologies if it features a number of the following techniques [4]:

Technical:

- CSS, semantically valid XHTML markup, and Microformats
- Unobtrusive Rich Application techniques (such as Ajax)
- Technologies such as XUL and SVG
- Syndication of data in RSS/Atom
- Weblog publishing
- JCC and REST or XML Web Service APIs
- Some social networking aspects

General:

- The site should not act as a "walled garden" - it should be easy to get data in and out of the system.
- Users usually own their data on the site and can modify at their convenience[6].
- Data returns should be dynamic, not static, changing depending on variables associated with the user's query[12].

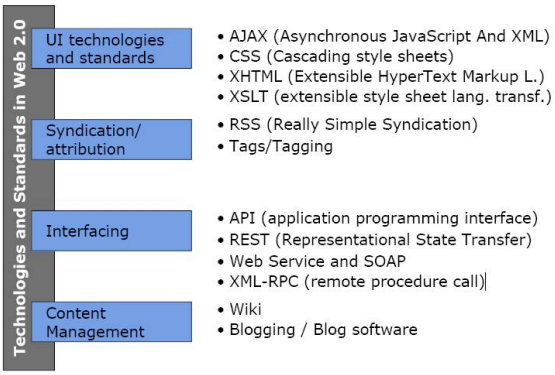

Figure 3: Technologies and standards in Web 2.0 [5].

V. Web 2.0 Examples

1. Flickr – A photo sharing website which allows users to upload their photographs and share it with anyone and everyone.

2. OrkutSocial :networking site which allows the users to send messages and communicate with other members.

3. YouTube : It allows the users to upload their videos and share it with everyone.

4. Blogs : Maintained by individuals or groups, they can be used to convey anything.

5. Google AD sense : Allows users to earn money through posting Google ads on their websites.

6. Wikipedia : Online encyclopedia wherein the users contribute by writing the articles, definitions,etc. It is completely edited and maintained by the users.

7. Scribd : Users can upload any documents on the website where other users can either download or view those documents online[25].

VI. Web 2.0 and Security

Web 2.0 is bringing in new security concerns and attack vectors into web.Here is the list of 10 attack vectors along with a brief overview of each:

A. Cross-site scripting in AJAX

In Cross-site scripting malicious JavaScript code from a particular Web site gets executed on the victim's browser thereby compromising information. AJAX gets executed on the client-side by allowing an incorrectly written script to be exploited by an attacker. The attacker is only required to craft a malicious link to coax unsuspecting users to visit a certain page from their Web browsers. This vulnerability existed in traditional applications as well but AJAX has added a new dimension to it[25].

B. XML poisoning

XML traffic goes back and forth between server and browser in many of the WEB 2.0 applications. Web applications consume XML blocks coming from AJAX clients. It is possible to poison this XML block. Not uncommon is the technique to apply recursive payloads to similar-producing XML nodes multiple times. If the engine's handling is poor this may result in a denial of services on the server. Many attackers also produce malformed XML documents that can disrupt logic depending on parsing mechanisms in use on the server. There are two types of parsing mechanisms available on the server side – SAX and DOM. This same attack vector is also used with Web services since they consume SOAP messages and SOAP messages are nothing but XML messages. Large scale adaptation of XMLs at the application layer opens up new opportunities to use this new attack vector[15].

XML external entity reference is an XML property which can be manipulated by an attacker. This can lead to arbitrary file or TCP connection openings that can be leveraged by an attacker[13]. XML schema poisoning is another XML poisoning attack vector which can change execution flow. This vulnerability can help an attacker to compromise confidential information.

C. RSS / Atom injection

RSS feeds are common means of sharing information on portals and Web applications. These feeds are consumed by Web applications and sent to the browser on the client-side. One can inject literal JavaScripts into the RSS feeds to generate attacks on the client browser. An end user visits this particular Web site loads the page with the RSS feed and the malicious script – a script that can install software or steal cookies – gets executed. This is a lethal client-side attack. Worse, it can be mutated[8].

With RSS and ATOM feeds becoming integral part of Web applications, it is important to filter out certain characters on the server-side before pushing the data out to the end user.

D. Malicious AJAX code execution

AJAX calls are very silent and end-users would not be able to determine whether or not the browser is making silent calls using the XMLHTTPRequest object. When the browser makes an AJAX call to any Web site it replays cookies for each request. This can lead to potential opportunities for compromise. For example, John has logged in to his bank and authenticated on the server. After completing the authentication process he gets a session cookie. His bank's page has a lot of critical information. Now he browses other pages while still logged in to his bank's account Web page and lands at an attacker's Web page.

On this page the attacker has written silent AJAX code which makes backend calls to his bank without John's consent, fetches critical information from the pages and sends this information to the attacker's Web site. This leads to a security breach and leakage of confidential information[9].

E. Client side validation in AJAX routines

WEB 2.0 based applications use AJAX routines to do a lot of work on the client-side, such as client-side validations for data type, content-checking, date fields, etc. Normally, these client-side checks must be backed up by server-side checks as well. Most developers fail to do so; their reasoning being the assumption that validation is taken care of in AJAX routines. It is possible to bypass AJAX-based validations and to make POST or GET requests directly to the application – a major source for input validation based attacks such as SQL injection, LDAP injection, etc. that can compromise a Web application's key resources. This expands the list of potential attack vectors that attackers can add to their existing arsenal.

F. WSDL scanning and enumeration

WSDL is an interface to Web services. This file provides key information about technologies, exposed methods, invocation patterns, etc. This is very sensitive information and can help in defining exploitation methods. Unnecessary functions or methods kept open can cause potential disaster for Web services. It is important to protect WSDL file or provide limited access to it. In real case scenarios, it is possible to discover several vulnerabilities using WSDL scanning [29].

G. RIA thick client binary manipulation

Rich Internet Applications (RIA) use very rich UI features such as Flash, ActiveX Controls or Applets as their primary interfaces to Web applications. There are a few security issues with this framework. One of the major issues is with session management since it is running in browser and sharing same session. At the same time since the entire binary component is downloaded to the client location, an attacker can reverse engineer the binary file and decompile the code. It is possible to patch these binaries and bypass some of the authentication

logic contained in the code. This is another interesting attack vector for WEB 2.0 frameworks [25].

H. Parameter manipulation with SOAP

Web services consume information and variables from SOAP messages. It is possible to manipulate these variables. For example, "<id>10</id>" is one of the nodes in SOAP messages[10]. An attacker can start manipulating this node and try different injections – SQL, LDAP, XPATH, command shell – and explore possible attack vectors to get a hold of internal machines. Incorrect or insufficient input validation in Web services code leaves the Web services application open to compromise. This is a new available attack vector to target Web applications running with Web services [5].

I. XPATH injection in SOAP message

Web applications consume large XML documents and many times these applications take inputs from the end user and form XPATH statements. These sections of code are vulnerable to XPATH injection[17]. If XPATH injection gets executed successfully, an attacker can bypass authentication mechanisms or cause the loss of confidential information. There are few known flaws in XPATH that can be leverage by an attacker. The only way to block this attack vector is by providing proper input validation before passing values to an XPATH statement.

VII. WEB 3.0

Web 3.0 isn't just about shopping, entertainment and search. It's also going to deliver a new generation of business applications that will see business computing converge on the same fundamental on-demand architecture as consumer applications. So this is not something that's of merely passing interest to those who work in enterprise IT. It will radically change the organizations where they work and their own career paths [27].

At the WWW2006 conference in Edinburgh,Tim Berners-Lee stated that he believes that the next steps are likely to involve the integration of high-powered graphics (Scalable Vector Graphics, or SVG) and that underlying these graphics will be semantic data, obtained from the RDF Web,that 'huge data space'[27].

Web 3.0 thus promises to be much more useful than 2.0 and to render today's search engines more or less obsolete. But there's also a creepy side to 3.0, which Markoff only hints at. While it will be easy for you to mine meaning about vacations and other stuff, it will also be easy for others to mine meaning about you [27]. In fact, Web 3.0 promises to give marketers, among others, an uncanny ability to identify, understand and manipulate us - without our knowledge or awareness.

REFERENCES

[1] Salisbury, W. David, Rodney A. Pearson, Allison W. Pearson, and David W. Miller. "Perceived security and World Wide Web purchase intention." Industrial Management & Data Systems 101, no. 4 (2001): 165-177.

[2] Aghaei, Sareh, Mohammad Ali Nematbakhsh, and Hadi Khosravi Farsani. "Evolution of the world wide web: From WEB 1.0 TO WEB 4.0." *International Journal of Web & Semantic Technology* 3, no. 1 (2012): 1.

[3] Garfinkel, Simson, and Gene Spafford. *Web security, privacy & commerce*. " O'Reilly Media, Inc.", 2002.

[4] Lawton, George. "Web 2.0 creates security challenges." *Computer* 40, no. 10 (2007).

[5] Rosenberg, Jothy, and David Remy. Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption. Pearson Higher Education, 2004.

[6] Andersen, Per. *What is Web 2.0?: ideas, technologies and implications for education*. Vol. 1, no. 1. Bristol: JISC, 2007.

[7] Carrie, E. Gates. "Access control requirements for web 2.0 security and privacy." In *Proc. of Workshop on Web 2.0 Security & Privacy (W2SP 2007*. 2007.

[8] Noureddine, Adam A., and Meledath Damodaran. "Security in web 2.0 application development." In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, pp. 681-685. ACM, 2008.

[9] Doroodchi, Mahmood, Azadeh Iranmehr, and Seyed Amin Pouriyeh. "An investigation on integrating XML-based security into Web services." In GCC Conference & Exhibition, 2009 5th IEEE, pp. 1-5. IEEE, 2009.

[10] Bhatti, Rafae, Elisa Bertino, Arif Ghafoor, and James BD Joshi. "XML-based specification for Web services document security." *Computer* 37, no. 4 (2004): 41-49.

[11] Chadwick, Andrew. "Web 2.0: New challenges for the study of e-democracy in an era of informational exuberance." *Isjlp* 5 (2008): 9.

[12] Rubin, Aviel D., Daniel Geer, and Marcus J. Ranum. *Web security sourcebook*. John Wiley & Sons, Inc., 1997.

[13] Pouriyeh, Seyed Amin, and Mahmood Doroodchi. "Secure SMS Banking Based On Web Services." In SWWS, pp. 79-83. 2009.

[14] Garfinkel, Simson, and Gene Spafford. Web security, privacy & commerce. " O'Reilly Media, Inc.", 2002.

[15] Akhawe, Devdatta, Adam Barth, Peifung E. Lam, John Mitchell, and Dawn Song. "Towards a formal foundation of web security." In Computer Security Foundations Symposium (CSF), 2010 23rd IEEE, pp. 290-304. IEEE, 2010.

[16] Ritchie, Paul. "The security risks of AJAX/web 2.0 applications." *Network Security* 2007, no. 3 (2007): 4-8.

[17] Pouriyeh, Seyed Amin, Mahmood Doroodchi, and M. R. Rezaeinejad. "Secure Mobile Approaches Using Web Services." In SWWS, pp. 75-78. 2010.

[18] Benzel, Terry. "The IEEE Security and Privacy Symposium Workshops." IEEE Security & Privacy 14, no. 2 (2016): 12-14.

[19] Dalvand, Babak, Saeed Safaei, and Mojtaba Nazari. "Fast Parallel Molecular Solution to the Maximum Triangle Packing Problem on Massively Parallel Bio-Computing." In FCS, pp. 169-173. 2009.

[20] Cormode, Graham, and Balachander Krishnamurthy. "Key differences between Web 1.0 and Web 2.0." *First Monday* 13, no. 6 (2008).

[21] Maness, Jack M. "Library 2.0 theory: Web 2.0 and its implications for libraries." *Webology* 3, no. 2 (2006): 2006.

[22] Allahyari, Mehdi, Krys J. Kochut, and Maciej Janik. "Ontology-based text classification into dynamically defined topics." In Semantic Computing (ICSC), 2014 IEEE International Conference on, pp. 273-278. IEEE, 2014.
[23] Safaei, Nozar, Babak Dalvand, Saeed Safaei, and Vahid Safaei. "Molecular solutions for the maximum K-Facility dispersion problem on DNA-based supercomputing." FCS, 2011.
[24] Assefi, Mehdi, Guangchi Liu, Mike P. Wittit, and Clemente Izurieta. "Measuring the Impact of Network Performance on Cloud-Based Speech Recognition Applications." *International Journal of Computer Applications-IJCA* 23 (2016): 19-28.
[25] Naedele, Martin. "Standards for XML and Web services security." *Computer* 36, no. 4 (2003): 96-98.
[26] Adams, Carlisle, and Sharon Boeyen. "UDDI and WSDL extensions for Web service: a security framework." In *Proceedings of the 2002 ACM workshop on XML security*, pp. 30-35. ACM, 2002.
[27] Hendler, Jim. "Web 3.0 Emerging." *Computer* 42, no. 1 (2009).
[28] Barassi, Veronica, and Emiliano Treré. "Does Web 3.0 come after Web 2.0? Deconstructing theoretical assumptions through practice." *New media & society* 14, no. 8 (2012): 1269-1285.
[29] Bratt, Steve. "Semantic web and other W3C technologies to watch." *Talks at W3C, January* (2007).
[30] Hendler, Jim, and Tim Berners-Lee. "From the Semantic Web to social machines: A research challenge for AI on the World Wide Web." *Artificial Intelligence* 174, no. 2 (2010): 156-161.