Convergence of Web 2.0 and SOA: Taking Advantage of Web Services to Implement a Multimodal Social Networking System

Stan Kurkovsky, David Strimple, Eric Nuzzi, Kerry Verdecchia
Central Connecticut State University
kurkovskysta@ccsu.edu

Abstract

The term Web 2.0 describes web-based applications such as social networking sites, wikis, and blogs that facilitate collaboration, creativity, and sharing among users. Web 2.0 applications are often enabled by reusing content from other web-based applications or web services. This trend is largely parallel to the principles of service-oriented architecture (SOA), which encourages integration and reuse of online services. In this paper, we discuss the possibility of convergence of Web 2.0 and SOA, which is illustrated by a multimodal social networking system built using web services. We also present the results of benchmarking experiments aimed to study the latency caused by web service composition and integration and its effects on the end-user experience.

Keywords: web services, service-oriented architectures, Web 2.0, social networking, VoiceXML.

1. Introduction

In 2004, Tim O'Reilly introduced the term Web 2.0 describing a rapidly growing type of web-based applications such as wikis, media tagging and sharing, blogs, and social networking that thrive on collective user intelligence propelled by the user-generated content and collaboration. In this philosophy, the web is viewed as merely a platform for creating systems that are "tied together by a set of protocols, open standards, and agreements for cooperation" among different application platforms [14]. From the technological perspective, many Web 2.0 applications are supported by a number of loosely coupled web-based information services and a combination of various rich media elements. As a practical implication of such a technological approach, dynamic creation and sharing of information in Web 2.0 applications maximizes the massively collective intelligence of the user community while providing benefits to each individual user [9].

An explosive growth in popularity of Web 2.0 paradigm and practical applications of service-oriented architecture (SOA) resulted in an increased interest in similarities, convergence, and a potential symbiosis of these two concepts [12,19,20]. SOA postulates the encapsulation of application logic within networked services that have formally defined and uniform interfaces and are publicly available through well-defined discovery mechanisms [1]. These services can be consumed and combined with other services as needed by each particular application. The idea of using loosely coupled services accessible through standardized interfaces is central to SOA; it provides for a great degree of service reuse, encapsulation of complexity, interoperability, and rapid application development. These features are similar to those of Web 2.0, where services and content provided by one application may be reused, recomposed and enriched by a wide range of other applications, potentially offering value-added services of a higher level. For example, Google Maps is probably the most popular service used in a countless number of mash-ups, which is well documented at www.gmapsdirectory.com and googlemapsmania.blogspot.com.

A great degree of similarity between SOA and Web 2.0 may be leading to a convergence or a symbiosis between the two paradigms. On the one hand, a Web 2.0 application can be created as a composition of loosely coupled and reusable services – the main building block in SOA. On the other hand, collective intelligence of self-serving users of Web 2.0 applications could be formalized into a value-added web service that could be used by other applications.

In this paper, we describe our experience of using web services and SOA principles to build a multimodal Web 2.0 application that facilitates social bookmarking [11]. Users of SeeCentralConnecticut (SeeCCT) can create voice or text bookmarks describing notable places, buildings, or other landmarks. All such bookmarks can be accessed via web or voice-based telephone interface. Additionally, users can form communities using different dimensions of their affinity, such as common interests or geographical proximity. We used a number of web services in our application; in particular, we used the web services provided by BeVocal.com to implement VoiceXML-based speech interface.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work in the areas of social networking, speech and multimodal interfaces, web services and SOA. Section 3 describes SeeCCT and outlines its architecture, while Section 4 presents the results of our experiments aimed at benchmarking the system's performance. Section 5 concludes the paper with a summary.

2. Related work

The web has become an increasingly important part of our everyday life making a profound impact on the way how people communicate. It is a part of the human nature to seek staying in touch with friends and being able to share the latest news and experiences. Today, there are new communication channels that allow us to send messages (email, instant messaging or texting, etc), blog about our daily activities, find new acquaintances based on common interests via social bookmarking [10], and share mementos via photo and video sharing services [7]. Furthermore, anonymity and the capability to communicate with others without a personal contact have made it more appealing to many users. As a result, by empowering the users to create shareable content, Web 2.0 applications helped advance the bottom-to-top information creation paradigm. The Internet itself is being transformed from a vast information repository to a facility that helps people connect using a range of social networking systems (SNS) [10,18,22]. Current SNS typically focus on a particular dimension of shared interests that may bring people together: maintaining and establishing new professional contacts (LinkedIn), communication among college classmates (FaceBook), or connecting people based on their musical preferences (Last.fm). There is an emerging class of mobile SNS deployed mainly in urban areas and designed to support serendipitous social connections [4,21]. SeeCCT and three other SNS described below (Murmur, Yellow Arrow, and Dodgeball) belong to this class of applications.

Murmur (murmurtoronto.ca) is an example of a social networking system that helps connect people based on their presence at or an interest in a particular landmark. Murmur allows its users to record and listen to virtual tours of various cities. Specially designed signs are installed in places of interest; each sign includes a telephone number that anyone can call and listen to a recording. Similarly to Murmur, Yellow Arrow (yellowarrow.net) distributes small yellow arrow-shaped stickers with a unique code. The code can be sent as a text message to Yellow Arrow, which replies describing something of significance to the original creator of the message who placed the yellow arrow sticker at that location. Dodgeball is currently available in 22 cities across the US; it provides a range of social networking tools including text messaging and mapping services [23]. For example, a preferred list of the user's friends currently located within a 10-mile radius could be set to receive text notifications about the current activities of that user.

Modern software is often designed with multiple venues for input/output in order to create an interactive interface that is closer to the natural interpersonal communication, or to make these applications more adaptable to the needs of different users in varying contexts [17]. Multimodal applications have a stronger acceptance potential because they can generally be accessed in more than one manner (e.g. using speech and web interface) and by a broader range of users in a varying set of circumstances. Speech remains the most basic and the most efficient form of interpersonal communication, and facilitating voice communication remains the main function of even the simplest mobile phone. As mobile devices grow in popularity and become ever more powerful and feature-rich, they still remain constrained in terms of the screen and keyboard size, battery capacity, and processor speed. There is a wide variety of models, manufacturers, and operating systems for mobile devices, each of which may have unique input/output capabilities. This creates new challenges to the developers of mobile applications, especially if they are to embrace different interaction modalities. Current efforts to establish a standard for the multimodal interface specification still remain far from being mature and are not widely accepted by the industry [3]. However, multimodal interface design is a rapidly evolving research area, especially in the area of mobile information services [2,8,13,16].

Web services offer a standards-based solution to the problem of connecting producers and consumers of information over the Internet [15]. Web services provide an extensible framework for inter-application communication and interaction, which is based on well-established web communication protocols (HTTP/HTTPS, SMTP) and open standards (XML). The framework of web services consists of three areas: communication protocols (the Simple Object Access Protocol, SOAP), service descriptions (the Web Services Description Language, WSDL), and service discovery (the Universal Description, Discovery, and Integration, UDDI) [6]. SOA is an architectural style for building software systems using a collection of services distributed over the network that can be combined together through a standardized data exchange. Web services are commonly used to implement SOA-based applications. One of the main advantages of SOA is an ability to rapidly develop applications using loosely coupled and reusable services that are accessible via standardized interfaces used to hide the complexity of their implementation. In many ways, these principles are mirrored by Web 2.0 applications, which too focus on reusing of existing resources through integration and composition. Similarly to SOA, many existing Web 2.0 applications also thrive because of their agility, both in application development and content generation. Despite these similarities, there are some features in which SOA and Web 2.0 differ significantly. The phenomenon of Web 2.0 is fueled by the user-generated content and social interactions, whereas the main objective of SOA is to interconnect business functionality and to facilitate computer-to-computer interaction. Subsequently, Web 2.0 applications focus on the presentation and broad usability of human-readable content, while typical SOA applications are geared to a specific and narrow business-oriented audience [20].

A significant potential of current SNS whose features include many different technologies assisting people to connect with each other are often blocked by the inability to cross the boundaries of each individual system. In each new SNS, the users usually need to create a new profile, although much of this information could be reused or imported from an existing profile on another system. Furthermore, integration and composition of multiple social connections from multiple SNS could create new social relationships and rich technological services of a much higher order. One of the possible solutions addressing this problem is to leverage the capabilities of SOA by creating a specialized social network-oriented framework of web services that would help interconnect people and content in more meaningful ways [5,19].

3. Multimodal mobile social bookmarking

In this paper, we present the See Central Connecticut (SeeCCT) system that incorporates many SNS features aimed at mobile users and is implemented using a composition of web services. Unlike Yellow Arrow, instead of using text messages, SeeCCT provides audio responses when accessed by phone. The user-generated content of the system's initial implementation is limited to attractions and notable places in Connecticut. SeeCCT is multimodal and has a web-based and a voice-based interface. The users can create, read, or listen to bookmarks using a telephone or the Internet at www.SeeCCT.com. They can also use this website to access bookmarks, search by the town name or by type, such as park, restaurant, school, etc, or by the user ID who created the bookmark. Users can create new bookmarks or leave feedback to existing bookmarks using both the website and voice interfaces [11].

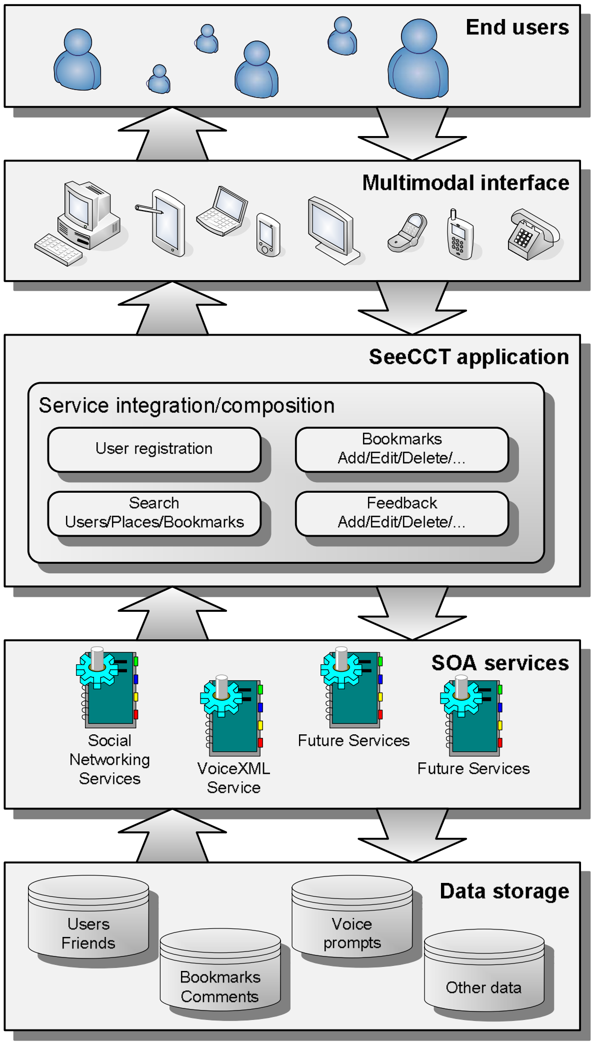
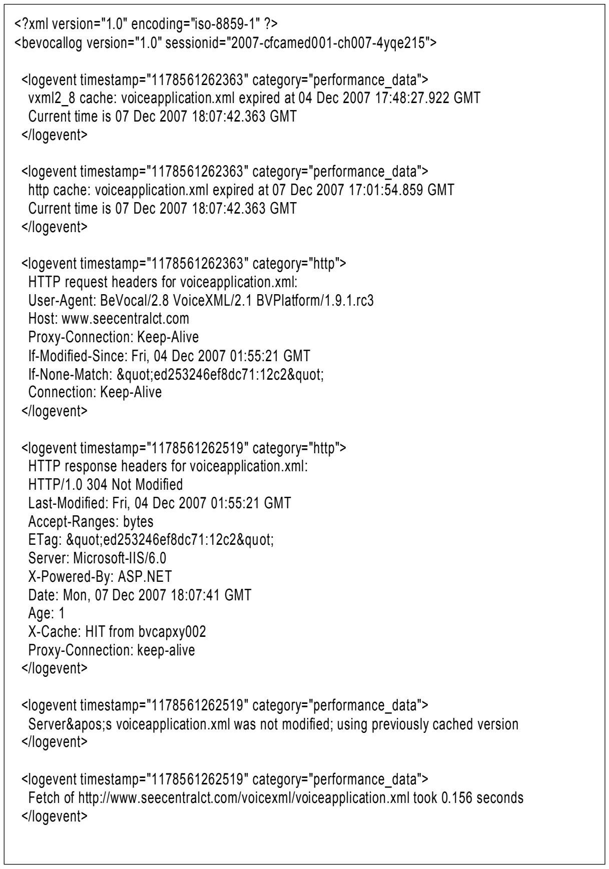


Figure 1. Service-Oriented Architecture of SeeCCT

Multimodal interface of the system includes the voice and web interfaces, as shown in Figure 1. Voice users interact with the system through the interface provided by BeVocal VoiceXML portal. SeeCCT Service Integration layer uses ASP.NET with C# to access the Social Networking Services and VoiceXML access services provided by BeVocal. This architecture is open to the addition and integration of other web services that may be needed in the future.

SeeCCT could be accessed using a combination of its voice or web interfaces. Generally, depending on the type of search request, the returned results may contain a combination of text and audio. Depending on which interaction mode is used, the system will generate the output suitable for that interface. If the user connects to the system using the voice interface, all text output will be played via VoiceXML text-to-speech converter. When the user accesses the system via the web, audio output will be linked to the resulting HTML pages and ready for playback [11].



```
<?xml version="1.0" encoding="iso-8859-1" ?>
<bevocallog version="1.0" sessionid="2007-cfcamed001-ch007-4yqe215">

 <logevent timestamp="1178561262363" category="performance_data">
  vxml2_8 cache: voiceapplication.xml expired at 04 Dec 2007 17:48:27.922 GMT
  Current time is 07 Dec 2007 18:07:42.363 GMT
 </logevent>

 <logevent timestamp="1178561262363" category="performance_data">
  http cache: voiceapplication.xml expired at 07 Dec 2007 17:01:54.859 GMT
  Current time is 07 Dec 2007 18:07:42.363 GMT
 </logevent>

 <logevent timestamp="1178561262363" category="http">
  HTTP request headers for voiceapplication.xml:
  User-Agent: BeVocal/2.8 VoiceXML/2.1 BVPlatform/1.9.1.rc3
  Host: www.seecentralct.com
  Proxy-Connection: Keep-Alive
  If-Modified-Since: Fri, 04 Dec 2007 01:55:21 GMT
  If-None-Match: &quot;ed253246ef8dc71:12c2&quot;
  Connection: Keep-Alive
 </logevent>

 <logevent timestamp="1178561262519" category="http">
  HTTP response headers for voiceapplication.xml:
  HTTP/1.0 304 Not Modified
  Last-Modified: Fri, 04 Dec 2007 01:55:21 GMT
  Accept-Ranges: bytes
  ETag: &quot;ed253246ef8dc71:12c2&quot;
  Server: Microsoft-IIS/6.0
  X-Powered-By: ASP.NET
  Date: Mon, 07 Dec 2007 18:07:41 GMT
  Age: 1
  X-Cache: HIT from bvcapxy002
  Proxy-Connection: keep-alive
 </logevent>

 <logevent timestamp="1178561262519" category="performance_data">
  Server&apos;s voiceapplication.xml was not modified; using previously cached version
 </logevent>

 <logevent timestamp="1178561262519" category="performance_data">
  Fetch of http://www.seecentralct.com/voicexml/voiceapplication.xml took 0.156 seconds
 </logevent>
```

Figure 2. Fragment of a sample VoiceXML call log

Because SeeCCT is a proof-of-concept application, the list of locations about which bookmarks could be made is limited to towns within a fifteen-mile radius of Hartford, the capitol of Connecticut. The system currently must be bootstrapped with this data, but it can be easily extended to allow creating bookmarks for new locations. The bookmarks can contain text, audio, or both. Each bookmark record also contains the information about its creator, bookmark type, and address. Using the SeeCCT web interface, bookmarks are searchable by each of these categories. Searching for bookmarks by user provides the capability to find other users who left an interesting bookmark or other records made by the same user.

As an SNS, SeeCCT facilitates interaction among the community of its users. The value of the system grows if a user can do more than simply read or listen to the records made by others. Feedback mechanism provides the medium for such an interaction within the system. For example, if a user leaves an outstandingly positive review about a particular restaurant in Hartford, another user may post a reply with a follow-up question or just saying thanks. Yet another user may choose to share a negative experience with the same restaurant. Therefore, a dialog may form through bookmarks and feedback. As a result, such an exchange will draw a better picture of the subject matter, and, therefore, will reveal the true value of the system. We conducted a substantial study to gauge the end-user acceptance of SeeCCT that indicated a high level of the users' satisfaction with the system [11].

4. Benchmarking the system

While the user interface for the voice platform is provided by a simple telephone, the services behind the scenes perform multiple complex tasks. The distributed nature of SeeCCT architecture can lead to a problem where the invocation and integration of various web services can generate latency in the system that may be unacceptable to a user. This latency can manifest itself in silence while loading a prompt or a bookmark, or the inability to quickly recognize an input value in order to perform the required action.

As shown in Figure 2, BeVocal creates XML logs for each call detailing the inner workings of the platform, along with error messages and time taken to process various events. A series of benchmarking experiments were performed on the SeeCCT system to determine its response time during a simulated call. A script was repeatedly run using BeVocal's Vocal Scripter utility. The call logs were then parsed and analyzed. The benchmark process exercised multiple execution paths within the SeeCCT system, which included logging into the system and creating, listening to and retrieving a bookmark. During these experiments, the system performed the retrieval of 20 existing and recording of 35 new bookmarks. Figure 3 shows the average latency for the most time-consuming network operations tracked during the benchmarking along with the number of times each of these operations was invoked. For identification purposes, each recorded prompt was assigned a 5-digit ID.

One of the most interesting findings was that the "Get VoiceService Web WSDL" operation was executed over 430 times. While the average latency was under 200 ms, some of the calls took almost an entire second, which could result in latency that would be noticeable by a user and, therefore, undesirable.

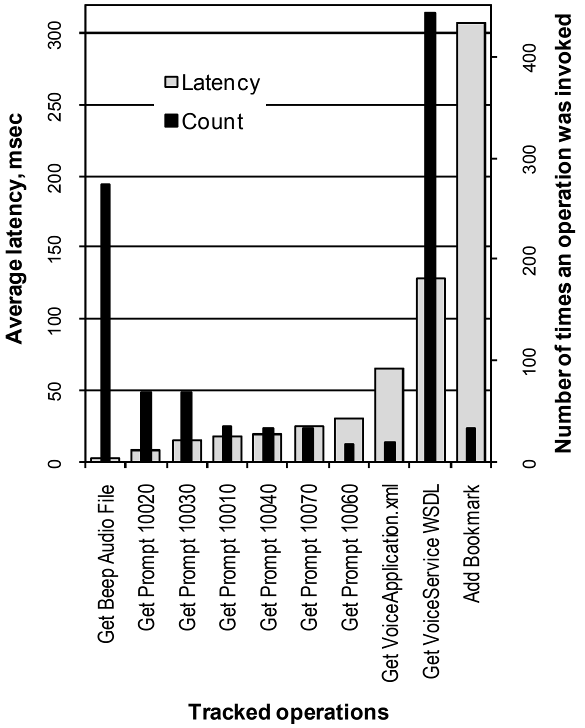


Figure 3. Aggregate latency and the number of times the tracked operations were invoked

The SeeCCT VoiceXML script communicates with web services via SOAP to send and retrieve data to and from the SeeCCT application servers. The WSDL file contains a description of all the methods supported by the application, which would reasonably need to be downloaded once at the beginning of each telephone call. The reason that the WSDL is downloaded so often can be attributed to the specifics of the VoiceXML specification. The flow of control in VoiceXML can generally be directed by two directives: <goto> and <subdialog>. The <goto> tag simply transfers control over to a new dialog within the current execution context preserving all global variables, and the subdialog may read and write to any global memory area. The <subdialog> tag executes a dialog within a new execution context, preserving nothing except variables explicitly passed to the subdialog, and passing back to the caller no other variables except those explicitly requested. This new execution context does not have a copy of the downloaded and parsed WSDL. The result is that every time a new dialog is called, the WSDL must be downloaded and parsed.

This problem described above could be solved by restructuring some parts of the code using the <goto> tag. However, widespread use of <goto>'s can lead to decreased code readability and "spaghetti code". It would be worthwhile to investigate more aggressive caching for the WSDL, allowing the file to be stored on the VoiceXML portal servers, eliminating the need to download the WSDL over the Internet for each <subdialog> execution.

The efficiency of storing files locally on the VoiceXML portal servers is illustrated by the "Get Beep Audio File" operation. The "beep" audio file is permanently stored on the server in the same datacenter as the servers processing the VoiceXML. The beep file is fetched multiple times, but because it is stored locally to the VoiceXML processor, the overhead is much lower. Low average latency of several "Get Prompt" operations shows the benefits of a more aggressive caching strategy. The bookmark and feedback audio are set as publicly cacheable. Even though the first fetch of each file took well over one second, resulting in a noticeable pause for the user, this pause only occurred once during the benchmark executions. This caching could also be extended to the audio prompts, which are read frequently, but do not change very often. Doing so would increase responsiveness of the system, while also greatly decreasing the bandwidth load on the network.

Another avenue for performance improvement is the pre-fetching of audio content. The bookmark and feedback audio files can be hundreds of kilobytes in size, resulting in long download times before the first playback. However, there is a gap between the time when the system knows that a bookmark or feedback will likely be played and the time that the system must actually play it. If the system fetches the audio file based on the probability that it will later need to play it then this first-play latency can be eliminated. Audio pre-fetching can be implemented with the pre-fetch attribute of the VoiceXML audio tag, or by downloading the file into the VoiceXML portal's cache so that the subsequent accesses can then be directed to the cache.

5. Summary

Many researchers believe that the current proliferation of Web 2.0 applications has a very strong potential to change the nature of how information is published on the web. The old top-to-bottom paradigm, in which a relatively small number of media corporations created content for the end users to access, may have to compete with the new bottom-to-top paradigm, when end users create massive amounts of content each contributing a small piece of information. An effective interaction among the users of many Web 2.0 applications, including SNS, is often enabled by the integration and composition of media and services provided by other online applications. This approach is similar to the principles of SOA. The relationship between Web 2.0 and SOA can be taken to the next level and the two paradigms may converge. In this paper, we demonstrated the viability of such an approach. SeeCCT is an SNS that is built using the SOA approach facilitated by a number of web services, including those provided by a VoiceXML portal. As a proof-of-concept application, it is limited in scope, but its web service-based architecture is scalable and will allow the system to embrace a larger scope of localities. We conducted a series of experiments to gauge the latency of the system's interaction with the users using its speech interface, which reveal some possible ways for improving the system by taking advantage of a more aggressive caching or pre-fetching of audio data.

6. References

[1] G. Alonzo, *Web Services: Concepts, Architectures and Applications*, Springer, Berlin, 2004.

[2] A. Augello, A. Santangelo, S. Sorce, G. Pilato, A. Gentile, A. Genco, S. Gaglio, "A Multimodal Interaction Guide for Pervasive Services Access," *IEEE International Conference on Pervasive Services*, 2007, pp. 250-256.

[3] J. Barnett, M. Bodell, D. Raggett, A. Wahbe (Eds.), "*Multimodal Architecture and Interfaces*," W3C Working Draft (2006), Retrieved on Mar 1, 2008 from http://www.w3.org/TR/2006/WD-mmi-arch-20061211.

[4] D. Bottazzi, R. Montanari, A. Toninelli, "Context-Aware Middleware for Anytime, Anywhere Social Networks," *IEEE Intelligent Systems*, Sep 2007, pp. 23-32.

[5] J. Breslin, S.Decker, "The Future of Social Networks on the Internet," *Internet Computing*, IEEE, Nov-Dec 2007, pp. 86-90.

[6] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana, "Unraveling the Web Services Web" *Internet Computing*, Mar-Apr 2002, pp. 86–93.

[7] S. Föll, J. Pontow, D. Linner, I. Radusch, "Classifying Multimedia Resources Using Social Relationships," *8th IEEE International Symposium on Multimedia*, 2006, pp. 690-695.

[8] G. Frattini, F. Gaudino, V. di Carlo, "Mobile Multimodal Applications on Mass-market Devices: Experiences," *18th International Conference on Database and Expert Systems Applications*, 2007, pp. 89-93.

[9] R. Högg, M. Meckel, K. Stanoevska-Slabeva, R. Martignoni, "Overview of Business Models for Web 2.0 Communities," *Proceedings of GeNeMe*, Dresden, 2006, pp. 23-37.

[10] T. Ishikawa, P. Klaisubun, M. Honma, "Navigation Efficiency of Social Bookmarking Service," *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2007, p. 280-283.

[11] S. Kurkovsky, D. Strimple, E. Nuzzi, K. Verdecchia, "Mobile Voice Access in Social Networking Systems," *5th International Conference on Information Technology: New Generations*, 2008.

[12] A. Mikroyannidis, Toward a Social Semantic Web, *IEEE Computer*, Nov 2007, pp. 113-115.

[13] W. Mueller, R. Schaefer, S. Bleul, "Interactive Multimodal User Interfaces for Mobile Devices," *37th Annual Hawaii International Conference on System Sciences*, 2004.

[14] T. O'Reilly, "*What Is Web 2.0,*" Retrieved from http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html on Feb 28, 2008.

[15] C. Peltz, "Web Services Orchestration and Choreography," *IEEE Computer*, Oct 2003, pp. 46-52.

[16] F. Pittarello, A. Celentano, "Deployment of Multimodal Services: an Ontology Driven Architecture," *IEEE International Conference on Pervasive Services*, 2007, pp. 267-274.

[17] L.M. Reeves, J. Lai, J.A. Larson, S. Oviatt, T.S. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin, M. McTear, T.V. Raman, K.M. Stanney, H. Su, Q.Y. Wang, "Guidelines for multimodal user interface design," *Communications of the ACM*, Jan 2004, pp. 57-59.

[18] D. Rosenblum, "What Anyone Can Know," *IEEE Security & Privacy*, May-Jun 2007, pp. 40-49.

[19] C. Schroth, "Web 2.0 versus SOA: Converging Concepts Enabling Seamless Cross-Organizational Collaboration," *9th IEEE International Conference on E-Commerce Technology*, 2007, pp. 47-54.

[20] C. Schroth, T. Janner, "Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services," *IT Professional*, May 2007, pp. 36-41.

[21] J. Thom-Santelli, "Mobile Social Software: Facilitating Serendipity or Encouraging Homogeneity?" *IEEE Pervasive Computing*, Jul-Sep 2007, pp. 46-51.

[22] A.C. Weaver, B.B. Morrison, "Social Networking," *IEEE Computer*, Feb 2008, pp. 97-100.

[23] N.D. Ziv, B. Mulloth, "An Exploration on Mobile Social Networking: Dodgeball as a Case in Point," IEEE International Conference on Mobile Business, 2006, pp. 26-32.