CALL FOR A PUBLIC-DOMAIN SPEECHWEB

All that’s missing is innovative use of existing technology and the encouragement of public participation.

Imagine using your cell phone to navigate through a network of hyperlinked speech applications as rich and diverse as the visual applications on the conventional Web. On holiday at the beach, you could use it to navigate through a vendor’s site for information on the new camera you’re using to take snapshots of your friends. In the evening, by the fire, you could verbally browse a library of electronic stories, picking the one most appropriate for the moment. Imagine, too, being able to build speech applications and add them to a public-domain SpeechWeb as easily as you build and add pages to the conventional visual Web. You could have a speech-accessible résumé. Information (such as directions on how to get to your home) could be made available for your guests to access through hands-free speech in their vehicles. Companies (with the expertise) could create natural-language speech interfaces to their products and services, making them available to anyone, anywhere using a handheld SpeechWeb browser. Moreover, a public-domain SpeechWeb would allow blind users to participate more fully in our knowledge-based society.
A Kelsey Group study estimated that speech-enabled services could generate more than $4.6 billion in revenue for North American wireless carriers and $25 billion worldwide by 2006 [6]. However, says Dale Hartzell of Sandcherry, Inc., a software company specializing in speech products, this world of speech-enabled services will be a reality only if speech solutions are able to deliver flexibility, scalability, and economy, and speech applications can be created and deployed as easily as text- and picture-based Web services are today [6]. This has not been possible due to the fact that existing architectures for distributing speech applications are not conducive to public involvement in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs. My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs. Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.

Here, I present a new architecture—called Local Recognition Remote Processing (LRRP) I’ve developed at the University of Windsor—for distributed speech processing, showing how it can be implemented through existing software and Web protocols and that it is conducive to the development of a public-domain SpeechWeb (see the figure). My goal is to encourage participation in the construction of SpeechWebs.
can be viewed as a variant of the second architecture, supports sophisticated applications requiring powerful processing. It also allows applications to reside on conventional Web servers, reducing administrative overhead. However, using the call-center architecture in a public-domain SpeechWeb also involves three main limitations:

**Need to store voice profiles.** In order to obtain adequate speech-recognition accuracy for advanced applications, user voice profiles must be stored at each call center or at the user site and transferred to the call center each time it is accessed, neither of which is practical;

**Need for expensive software.** Application providers must employ specialized, typically expensive, software to allow their applications to be accessed from remote telephones; and

**Need to support hundreds of concurrent users.** Call centers must be able to provide speech-recognition capabilities for hundreds of concurrent users.

### SALT vs. VXML

The protocol called Speech Application Language Tags (SALT) developed by Microsoft, Intel, and other sources [11] includes tags (fewer than VXML) that are intended to be added to HTML pages so they can be browsed through speech. An advantage of this approach is that existing Web browsers are more easily extended to accommodate SALT tags; SALT relies on the execution model of HTML. On the other hand, VXML is an independent language that can be used to create and hyperlink speech applications. VXML and SALT might converge over the next five years into a single voice-application markup language.

The Microsoft Speech Server (MSS), which supports SALT, is intended to simplify the integration of an enterprise's Web services, speech processing, and telephony. MSS can help implement all three of the architectures described earlier for distributed speech applications. However, the role MSS might play in the creation of a public-domain SpeechWeb is less clear, owing to its high cost and complexity for nonexpert SpeechWeb-content developers.

### A NEW ARCHITECTURE

Statistical language models, according to [8], were popular in 1995, but by 2001 grammar-based models had claimed the preeminent position in commercial speech-recognition products. My proposed LRRP architecture takes advantage of this fact, making use of the grammar-based speech-recognition software readily available today. In the LRRP architecture, speech applications and their associated recognition grammars are stored on regular Web servers, and LRRP speech browsers reside on end-user devices. When users access remote applications, grammars are downloaded to their browsers where they are used to tailor the speech recognizer for the application. When a user utterance is subsequently recognized, it is sent to the remote application for processing. If the utterance is a request for information, the result is returned in the form of synthesized voice. Speech recognition is local, and application processing is remote. Alternatively, users might ask to be connected to other speech applications by uttering a speech hyperlink (such as “Can I talk to the chemistry expert?”). The Web address of the new application is then returned to the browser, which downloads a new recognition grammar from that address and subsequently sends all recognized utterances to the new application.

The LRRP architecture has something in common with each of the three architectures discussed earlier. Recognition is local, as it is with speech interfaces to HTML pages. The downloading of recognition grammars from remote applications is similar to SAM’s smart pages. Like VXML, LRRP is intended to support access to applications designed for speech (rather than multimodal access to HTML pages). And finally, it involves remote execution of applications, like the call-center architecture. In terms of support for a pub-
lic-domain SpeechWeb, LRRP reflects a combination of the following advantages:

Applications on conventional Web servers. Applications that reside on conventional Web servers can be written in any language, provided that input and output conform to the Web communication protocol being used;

Available to nonexperts. Nonexperts can create simple applications as scripts that return canned answers to user queries, while advanced developers can create complex applications that use natural-language processors and databases residing on powerful server-side machines;

Improved client-side speech-recognition accuracy. Recognition accuracy is improved through the downloading of application-specific grammars and locally maintained user voice profiles; and

Communication through text. Using conventional Web protocols, clients and servers communicate through text, which is efficient when users access the SpeechWeb through commercial wide-area wireless networks.

Prototype Implementation

Aiming to demonstrate the viability of the LRRP architecture, my research group and I have experimented with a number of technologies for building browsers and deploying speech applications. We built early versions of the browser (1998) in Java using IBM’s proprietary speech APIs. I demonstrated them at the 1999 Conference of the Pacific Association of Computational Linguistics [5] and at the 2002 Conference of the American Association for Artificial Intelligence [4].

Last spring, one of my students and I rebuilt the browser as a single VXML page, integrating it with two simple Java objects [12]. The single VXML page remains on the local device and accesses remote hyperlinked non-VXML speech applications. In order to run this upgraded browser, users install the Java runtime environment and a VXML interpreter, both readily available as free downloads on the Web. In order to add a speech application to the Speech-Web, developers need access to a Web server and software to create their text-in/text-out applications. We tested the browser on PCs and laptops using both wired and wireless connections to local and wide-area networks, then demonstrated it in 2004 at the second annual conference on Communications Networks and Services Research in Fredericton, N.B. [3].

My somewhat unorthodox use of VXML (to implement the LRRP browser) provides several notable advantages: the browser can be distributed freely; VXML interpreters are available for a range of devices, including desktop PCs, laptops, and pocket PCs; and the browser inherits improvements in speech-recognition technology as they’re integrated into new versions of VXML interpreters. Finally, this method of constructing the browser facilitates integration of the LRRP architecture with the conventional “networks of VXML pages” architecture; the result is a richer environment for SpeechWeb application development. I also expect that the browser can be implemented as a single HTML page containing appropriate SALT tags, with comparable advantages.

Technological Challenges

The functionality of a public-domain SpeechWeb will be limited by three main issues:

• Lack of readily available techniques and tools to create sophisticated natural-language speech applications;
• Inability of speech-recognition technology to support free-flowing speech; and
• Limited availability of speech-recognizers that execute locally on lightweight devices (such as cell phones and handhelds).

Addressing the first involves development of special tools and techniques. Tools are needed to facilitate construction of natural-language applications by nonexperts. Microsoft, IBM, and other sources have developed software for constructing natural-language interfaces to proprietary database systems. In addition, advanced developers can construct natural-language processors using Yet Another Compiler
The conventional Web has grown dramatically in terms of numbers of users and Web sites, e-commerce activity, collaborative applications, information accessibility, and social networking over the past decade. The availability of free browsers and the ease of developing HTML pages represent important factors. My aim here is to motivate development of a public-domain SpeechWeb with which speech browsers and applications can be built and deployed using readily available software and conventional Web protocols.

**Conclusion**

The conventional Web has grown dramatically in terms of numbers of users and Web sites, e-commerce activity, collaborative applications, information accessibility, and social networking over the past decade. The availability of free browsers and the ease of developing HTML pages represent important factors. My aim here is to motivate development of a public-domain SpeechWeb with which speech browsers and applications can be built and deployed using readily available software and conventional Web protocols.

**References**

11. SALTForum Speech Application Language Tags (SALT) 1.0 Specification. SALTForum, 2002; www.saltforum.org/.

**Richard A. Frost** (rfrost@cogeco.ca) is a professor in the School of Computer Science at the University of Windsor, Windsor, ON, Canada.

This research is supported by the Natural Science and Engineering Council of Canada. Sanjay Chitte, Kunal Bhatia, and (Jack) Li Su (all graduates of the University of Windsor) helped implement versions of the SpeechWeb browser.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.