

Pardon Me, Your Computer's Showing:
Using speech to speed and streamline desktop computing

Kimberly Patch

Redstart Systems
22 Conway St., Boston, MA 02131
kim@redstartsystems.com
(617) 418-3090

Copyright is held by the author.

SpeechTEK West 2007, February 21-23, San Francisco, California, USA.

ABSTRACT

The way we use computers has been heavily shaped by the dominant input methods of keyboard and mouse. Controlling a computer using speech has, not surprisingly, closely mimicked keyboard and mouse input.

While it is good to tap existing knowledge, it is important not to let experience confine new methods of communication.

The potential differences between speech input and keyboard and mouse input are akin to the differences between road and air travel. Following a road via airplane is faster than driving, but the real power of air travel is the ability to travel any route, including areas inaccessible by car, like large bodies of water, mountain ranges and polar regions.

The real power of speech is the ability to command the computer in ways not possible using the keyboard and mouse alone.

The keys to unleashing this potential are minimizing steps, making commands easy to remember, and enabling combinations.

This paper includes speech command sequence demonstrations that show these principles in action. Demonstrations include word processing, tables and graphs, windows handling, cutting and pasting among documents, Internet navigation, email, speech links, a list utility and a virtual calculator.

This presentation is the second in a series outlining the untapped potential of speech on the desktop. The first presentation was given at *SpeechTEK 2006* August 7-10, New York and is posted at www.redstartsystems.com/papers.html.

1. INTRODUCTION

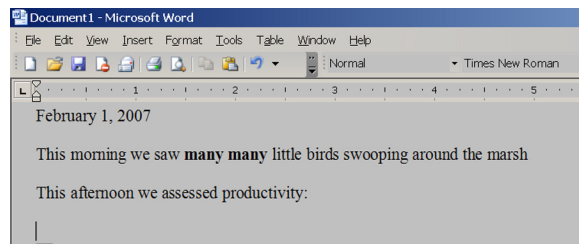
I'm here to talk about how to use speech to control a computer in a way that's natural and efficient. First I'm going to show you how I use speech to control a computer. I'll do some word processing, make a table and graph, move windows around, open dialog boxes, and access the Internet.

The speech interface software I'm using is Redstart Systems' Utter Command. Some of you might have seen some features of Utter Command at a SpeechTek talk in New York last summer. We're now just finishing up beta testing. This version uses the NaturallySpeaking speech engine. We replace all the NatSpeak commands.

You might notice that our syntax is different from existing speech interfaces and we tend to get more done with each command.

Another thing that makes us different is our commands are not program specific – these commands work across all programs. As you saw, these generic commands allow you to do what you want even in a complicated program like Excel.

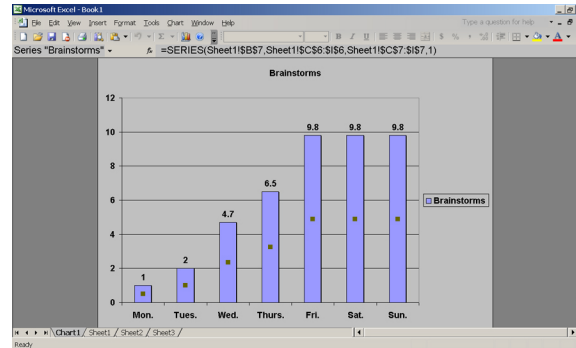
Speech On Word Open
 Today Date Long
 Another Graph
 we saw many many little birds
 skimming over the water
 4 Before
 swooping over the marsh
 2 Before 1 Before Delete
 around
 3rd Word 2 Words Bold
 Go Home
 This morning
 Right Delete w
 Another Graph
 Under t w Close



Excel Open Max
 Cell Charlie 6
 Days Short Tab
 Cell Bravo 7
 Cap Brainstorms
 Tab 1 Tab 2 Tab Short
 4 Point 7 Tab Short
 6 Point 5 Tab Short
 9 Point 8 Tab Short Repeat 3
 Alternate Equals Enter
 8 by 91 No Touch
 Cell Echo 7 Touch
 Cell Golf 7 Touch
 5 2 Letters
 1 Ups

	Mon.	Tues.	Wed.	Thurs.	Fri.	Sat.	Sun.
Brainstorms	1	2	4.7	6.5	9.8	9.8	9.8

Function 11
 Control Left
 Shift Control Papa 1 5 Enter
 This Bold
 Control Right
 Control 1 Under Delta
 3 Control Tab
 Under Victor Enter



Control Page Down
 Copy to Word

	Mon.	Tues.	Wed.	Thurs.	Fri.	Sat.	Sun.
Brainstorms	1	2	4.7	6.5	9.8	9.8	9.8

Size 90 by 40
 Window 0 by 50
 Window 0 by 0
 Screen Clear
 Word Max
 Excel Close No
 Word Close No
 Control Panel Delta Down Enter
 New York Times Site
 Screen 3
 Task Processes
 Window Close Times 4 Speech Off

2. KEYS TO USING SPEECH TO SPEED COMPUTER CONTROL

There are three keys to using speech commands to speed and streamline computer control:

- Minimizing steps
- Making commands easy to remember
- Enabling combinations

Utter Command is strong in all three areas because it is anchored by an efficient grammar – Human-Machine Grammar — that follows the way the brain works and is designed for the purpose of controlling a computer. Let’s take a detailed look at each of the three keys to speeding and streamlining computer control.

2.1 Minimizing steps: retrieving a picture

Controlling a computer using speech has, not surprisingly, closely mimicked the input methods we are familiar with – the keyboard and mouse. This has made for a disappointing decade and a half of speech control of the desktop. Although it's good to tap existing knowledge, it's also important not to let experience confine new methods of communication. It takes 11 keyboard and mouse steps to paste a picture that's buried four levels deep within the file system into a Word document.

The steps are necessary because the keyboard and mouse use restricted resources – a limited number of key combinations and a limited amount of screen space to click on. It doesn't seem slow going through a series of menus because it's familiar, and things are busily moving along. But when you think outside the silicon box it becomes apparent that there really shouldn't be as many steps involved.

Instructing a person using as many steps as it takes to retrieve a picture using the mouse and keyboard might go something like this: stand up, walk to the file cabinet, open the third drawer, go to the "B" section, find the bird folder, open the bird folder, find the picture subfolder, find the redstart singing picture, take it out of the folder, put in my hand, let go. Instructing a person is usually much more efficient: Can you get the Redstart Singing picture?

It's obvious that it's inefficient to instruct a person step-by step. It's inefficient to instruct the computer step-by-step as well. But it's not as obvious because inside the box these steps, required by the GUI, are comfortably familiar.

Most of today's desktop speech software follows in the footsteps of the keyboard and mouse, pausing between steps. But it doesn't have to be that way. It takes two or three steps, depending on the method you use, to paste a picture into a document using speech. Eleven steps versus two steps is a 550% difference in efficiency.

Fig. 1: Pasting a picture, keys/mouse

Click Insert
 Click Picture
 Click From File
 Click Look In
 Click Program Files
 Click Redstart Systems
 Click Utter Command
 Click Demo
 Click Bird Pictures
 Right Arrow
 Enter

Speech Demo 2: Pasting a picture using speech

Speech On
 Word Open
 Under i p f
 Bird Pictures Folder
 1 Right Enter
 Window Close No 3 Seconds Break
 Redstart Singing File
 All Copy to Word New
 Window Close No Window Close Speech Off

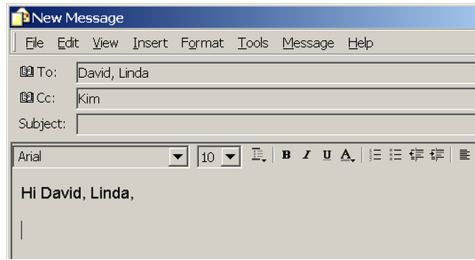
2.2 Minimizing steps: sending e-mail

Let's look at another task – sending email. Addressing an email to two people, CCing a third and writing a subject and first line takes 12 keyboard and mouse steps. It can be dramatically fewer using speech.

2.3 Road versus Air

The potential differences between speech input and keyboard and mouse input are akin to the differences between road and air travel. Following a road via airplane is faster than driving, but the real power of air travel is the ability to fly point-to-point and travel any route. Using speech to follow in the footsteps of the mouse is akin to using an airplane on the ground. It's accommodating the GUI rather than moving beyond it.

Fig. 2: Addressing an email to two people, CCing a third and readying a subject and first line

Key/mouse: 12 steps	Efficient Speech: 3 steps
Click Outlook Express icon Click Message Click New Message Type "David, Linda," Tab Key Type "Eric" Tab Key Type "Paris trip" Tab Key Type "David, Linda," Enter Enter	Express David Linda CC Eric Paris trip 1 Tab 

Speech Demo 3: Addressing an email

Speech On
 Express David Linda CC Kim
 Paris trip
 1 Tab
 Window Close No Window Close Speech Off

2.4 Remembering Commands

As we've seen, steps are very important – the speech interface is not easy to use unless it's fairly efficient. But there's more involved in making the speech interface easy to use than keeping steps to a minimum. Those steps have to be easy to remember.

I'm going to talk about talking for a minute, then we'll come back around to remembering commands.

When discussing philosophy you need a lot of words on tap. You're exploring a complicated subject and changing gears on-the-fly based on agreement, argument, and different types of understanding. Vocabulary matters. Subtle timing matters. Dramatic gestures matter. And they all take cognitive energy. This is okay when you're arguing philosophy because arguing philosophy is what you're focused on.

Talking to a computer is a means to get something else done. Here cognitive load matters. You don't want to use half your brain power remembering commands when you're trying to concentrate on writing a talk or doing research or putting numbers in a spreadsheet or editing a picture or even organizing your email.

2.5 History

Remembering speech commands is a long-standing problem. In the late '90's NatSpeak architect Joel Gould attended a Boston Voice Users Group meeting and posed the question "How do you remember commands?"

A year later, Joel illustrated the problem during a demo of a new version of NatSpeak. He changed font sizes and colors. The audience, which included regular and long-time NatSpeak users, oo'ed and ah'ed. Joel asked if these were commands we'd use, and many of us emphatically said yes, these are great, we're going to use them all the time. Then he said "you do have these – these are in the old version. Nobody uses them". A few years later, I talked to someone who had done NatSpeak trade show demos and she talked about how hard it was to memorize the long Excel commands that looked impressive.

When speech trainers get together a similar topic comes up again and again: why do people so often ask trainers to be supplemental memory, to simply feed commands as needed?

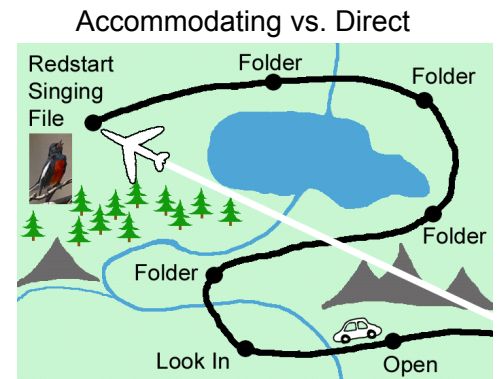
All of these anecdotes point to the same problem—the type of speech command language we're using imposes too heavy a cognitive load.

2.6 What speech command language needs to accomplish

The language that's required to control a computer isn't nearly as complicated as the vocabulary needed to talk about philosophy. It's not even as complicated as what you would use for idle chat with a friend.

Using speech on a computer is more like giving orders in a fast food kitchen — "Two Fry", calling a play on a football field — "Counter Trey Right", a police dispatch — "Unit 26, Code 11-31, 13th and Vine", or an air traffic control communication — "Delta 265, clear to land, runway three zero". Probably the most comparable language to the ideal for controlling the objects on a computer screen are the commands fictional magical characters like witches and wizards give to objects around them: "Chair Dance", "Teeth Grow". Here's a similar exercise: picture me as your stereo controller – that's all I do – what would we work out as a speech interface? Would you say "Can you please change the channel to my second favorite blues station"? Or "Blues 2"?

Fig. 3: Opening a file four layers deep



The language humans work out in command-and-control situations is a natural for issuing the same commands over and over to a computer.

This type of language – efficiently worded instruction — is what the Human-Machine Grammar underlying Utter Command is all about.

Fig. 4: Efficiently-worded instruction comparison

Human-Human Two Fry Counter Trey Right Delta 265, Clear to Land, Runway 3-0 Chair Dance Blues 2	Human-Computer Word Open Down Home Delete Bird Pictures Folder Window 0 by 20 Window Close No
--	--

2.7 Human-Machine Grammar

Human-Machine Grammar and Utter Command were informed by cutting edge research in cognition, linguistics, networks, and interfaces. Important factors include

- How the brain processes words
- The memory phenomenon of chunking
- The network phenomenon known as six degrees of separation

Human-Machine Grammar

- Has no synonyms
- Uses logical rules to minimize wording possibilities
- Follows the way the brain uses language

2.8 In Practice

Utter Command has 257 command words. Ninety-seven of these identify keyboard keys, leaving 160 words to learn to master all of Utter Command. These words are by design easy to remember, and commands are consistently constructed according to 16 grammar rules.

Learn about a third of the words, many of which are common and obvious, and a handful of general rules, and you'll find yourself humming along nicely saying things like 5 Down, Line Cut, and Window Close.

Figure 5 shows the 60 most commonly used words of the Utter Command vocabulary. Most are very familiar — copy, window, undo — and many are paired — up, down, left, right, before, after, top, bottom, open, close.

Fig. 5: 60 most common Utter Command words (+ numbers and screen labels)

All Caps	Menu	Speech	Left/Right
Another	Message	Spell	Lefts/Rights
By	Microphone	This	Before/After
Cap	Mouse	Touch	Before/Afters
Check	New	Touch 2	Up/Down
Clear	Nope	(Touch) Right	Ups/Downs
Compound	Paste	Tray 1-20	Line/Line Up
Copy	Redo	Under	Lines/Line Ups
Cut	Screen	Undo	On/Off
Go	Seconds	Win(dow)	Open/Close
Graph	Short	Word	Top/Bottom
Insert	Site	Words	Max(imize)/ Min(imize)
		<0-200>	
		<1st-20th>	
		<screen labels>	

And here are the the most common Human-Machine Grammar rules:

- Match words used for a command as closely as possible with what the command does
- Be consistent
- Eliminate synonyms
- Follow the way people naturally adjust language to fit a situation
- Follow the order of events (generally identify or select, then carry out action)

Match words used for a command as closely as possible with what the command does. For example, **“Window Close”**. Be consistent – always use the same words for the same actions – **“Line Bold”**, **“Line Copy”**. Eliminate synonyms – the smaller the command vocabulary the easier it is to remember and use. Follow the way people naturally adjust language to fit a situation – commands that follow the way the brain works are easier to remember and use. And follow the order of events as they’ll be carried out – generally you identify or select an object like a line or paragraph, then carry out an action – **“Line Delete”**. This dramatically cuts wording possibilities and is easier because it follows the way you think about doing the task.

Put the words together with the rules and you get commands like **“Speech On”**, **“3 Before”**, **“Line Copy”**, **“Screen Clear”**, and **“Window Close”**.

2.9 Combining commands

This brings us to the third key to using speech to speed and streamline computer control: enabling combined commands. Utter Command’s terse grammar makes it possible to say several commands in a single phrase, which speeds computing considerably. You’ve heard me say combined commands like **“3 Befores Delete”** and **“Word Close No Speech Off”**.

3. METRICS

I’d like to stress one other thing. Like the keyboard and mouse – Utter Command works across all programs. It allows you to do everything you do with the keyboard and mouse, often faster. The grammar generally cuts the number of steps by 100 to 200 percent and in some cases many times that. Here are some typical commands – you’ve seen me do all three of these.

Take a look at the first example – it takes a single UC command to bold words before and after the cursor, but it’s three steps using the keyboard and mouse and four steps using speech that follows in the footsteps of the keyboard and mouse. And as you’ve seen, the difference in steps is more dramatic with tasks like preparing an email message and starting a program and opening a file that lives four folders deep in the file system.

We also know that 74% of users prefer a structured rather than natural language approach to speech recognition — that’s from a Carnegie Mellon study [1].

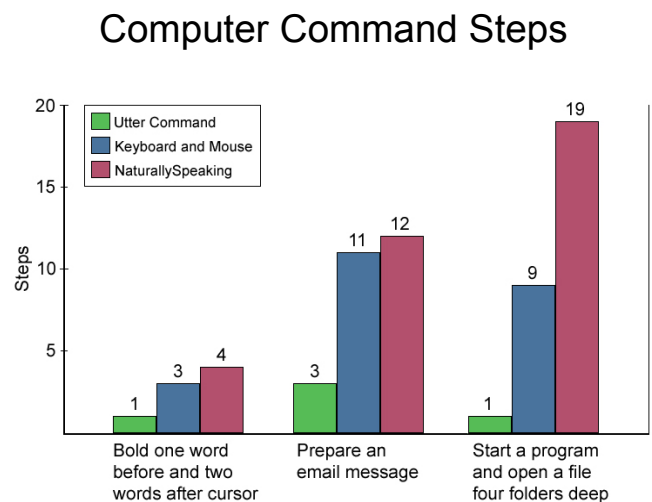
4. BEYOND THE KEYBOARD AND MOUSE

And here’s where things get interesting. The real power of speech is the ability to command the computer in ways not possible using the keyboard and mouse alone. Getting away from the constraints of the keyboard and mouse – once you recognize you can be free from them – gives you new abilities.

I’ll demonstrate four types of commands that go beyond the keyboard and mouse in some way:

- Speech links
- Document access and navigation
- Favorites-like lists
- Virtual calculator

Fig. 6: Command-step comparison, UC, key/mouse, NatSpeak



4.1 Speech Links

Augmented reality is inherent in speech recognition. Here's a simple example.

4.2 Documentation

The augmented reality inherent in speech recognition can also speed getting around documentation. Finding what you want in documentation is a classic problem.

We think one reason is the number of steps involved in finding what you want. Finding instructions to add a special character to your Word document using the standard electronic documentation, for instance, takes 8 steps if you go through the table of contents and 7 steps if you use the search function.

The Utter Command manual table of contents — printed or electronic — provides all the information you need to use a single speech command to go directly to any section of the documentation.

Speech Demo 5: One-step documentation access.

<p>Speech On UC Full 10 Point 9</p> <p>7 Seconds Break <i>This is the Reference section — say you want a more detailed explanation of the same commands from the Lesson section. It's one step to get there.</i></p> <p>UC Lesson 10 Point 9</p>	<p>2 Seconds Break <i>And here's the same section in a quick reference cheat sheet.</i></p> <p>UC Quick 10 Point 9 Window Close Speech Off</p>
--	---

4.3 List Commands

Utter Command contains a utility that allows you to keep favorites-like lists of Web sites, files and folders you can access in a single command.

4.4 Virtual Calculator

All the functionality you've seen until now is in the first version of Utter Command. This next demonstration is a proof-of-concept prototype.

It's a classic example of the limits of the GUI that, even though Windows comes with a calculator, people still buy separate calculators. It takes fewer steps to reach for a physical calculator than to click more than once to get to the calculator applet.

Using speech you can change the equation. You don't have to reach for the virtual calculator at all, and it presents the answer exactly where you want it in your choice of ways. The work and the answer, or just the answer.

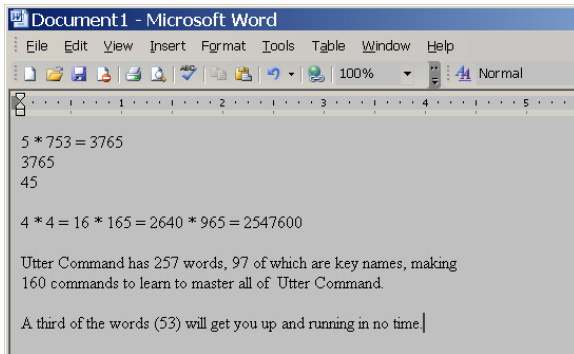
Speech Demo 4: Accessing static URLs and searching a Web site

<p>Speech On Demo 9 File</p> <p>8 Seconds Break <i>Here's a Web address that's not linked, so you can't click on it. But if you're using speech you don't need the link — it's inherent in the command.</i></p> <p>Line Copy to Firefox</p> <p>9 Seconds Break <i>This makes any Web address a speech link whether or not it is linked.</i></p> <p><i>Here's another type of speech link — searching for any word on dictionary.com</i></p> <p>Word Open 2 Down 1 Afters This Dictionary Search</p> <p>Window Close Times 2 Speech Off</p>

Speech Demo 6: Web sites, files, folders and keywords

<p>Speech On Slash Dot Site Demo 1 File Windows UC Demo Folder Window Close Times 3 9 Seconds Break <i>UC List also contains a list of keywords that works with the Find function of any program. You can arrange this list as you like, deleting and adding anything you want.</i></p> <p>Demo 8 File Find Scene 2 Find Scene 3 Find Courtney 7 Seconds Break <i>These commands work because "Scene 2", "Scene 3" and "Courtney" are all on my keyword list.</i></p> <p>Window Close Speech Off</p>

Fig. 7: The end result of the virtual calculator Speech demo



5. SUMMARY

To sum things up, there's a lot of good news here. Using a command language that's based on efficient human process rather than the constraints of the keyboard and mouse allows you to take only as many steps as necessary. If you don't have to think between steps, there's no need for separate steps. It makes commands easy to remember, which frees your brain to do tasks on a computer. And it enables combinations – which further speeds things up. Few steps, easy-to-remember commands, and combinations allow you to fly rather than take the ground route. You've heard me use 100 speech commands during this talk. If I'd used the mouse and keyboard instead, I'd have had to use 338 commands (Demo 1: 45/116; Demo 2: 8/28; Demo 3: 5/15; Demo 4: 9/25; Demo 5: 7/18; Demo 6: 11/41; Demo 7: 14/95).

The subject of this paper is the tip of an iceberg. Human-Machine Grammar and Utter Command are baseline technologies that will enable much more.

The Human-Machine Grammar rules and dictionary are posted at the Redstart Systems site at www.redstartsystems.com. Human-Machine Grammar (HMG) is a mature structured grammar that was developed over the past decade in a real-world environment. HMG includes the 257 words that appear in Utter Command as well as more than 100 more that go beyond what is currently in Utter Command. There are also talks posted on the site that go into more detail about the thinking and science behind Human-Machine Grammar [2-5]. We're encouraging everyone who writes speech commands to use Human-Machine Grammar.

The Human-Machine Grammar rules and dictionary are posted at the Redstart Systems site at www.redstartsystems.com. Human-Machine Grammar (HMG) is a mature structured grammar that was developed over the past decade in a real-world environment. HMG includes the 257 words that appear in Utter Command as well as more than 100 more that go beyond what is currently in Utter Command. There are also talks posted on the site that go into more detail about the thinking and science behind Human-Machine Grammar [2-5]. We're encouraging everyone who writes speech commands to use Human-Machine Grammar.

6. STEP-COUNTING METHODOLOGY

Our method for comparing speech versus mouse-and-keyboard command steps is as follows:

- We ignore pure text.
- When a string of characters occurs as part of a speech command, we count the characters, regardless of how many there are, as a single command for the keyboard and mouse. For instance, "Tab 7.8", a single UC command, counts as two mouse and keyboard commands: "Tab Key" and the text string "7.8".
- We use the most efficient keyboard/mouse command sequence, disregarding any awkwardness involved in switching between keyboard and mouse.
- We assume that Web addresses are in the first layer of a favorites list.
- We assume that files have not been recently accessed.

Speech Demo 7: The virtual calculator

Speech On Word Open

Five Times Seven Five Three Equals
Another Line

Five times Seven Five Three Answer
Another Line 2 Seconds Break

Or you can have it both ways.

Five Four Divided by One Point Two Wait Answer
Another Graph 2 Seconds Break

You can also keep going.

4 Times 4 Equals

Times 1 6 5 Equals

Times 9 6 5 Equals

Another Graph 2 Seconds Break

This should sum things up.

Utter Command has 257 words

Comma 97 of which are key names

Comma making New Line

2 5 6 Minus 9 7 Wait Answer

Commands to learn to master all of

Utter Command Period New Line

A third of the words Comma

1 6 0 Divided By 3 Wait Answer

Comma will get you up and running in no time Period

Window Close No Speech Off

7. REFERENCES

- [1] Stefanie Tomko and Roni Rosenfeld. Speech Graffiti vs. Natural Language: Assessing the User Experience. Proc. HLT/NAACL, Boston, MA, 2004. (www.cs.cmu.edu/~usi/pubs.htm)
- [2] Kimberly Patch. When Natural Language Isn't: The Need for a Dedicated Speech Interface. SpeechTek 2006. August 8, 2006. (<http://www.redstartsystems.com/papers.html>)
- [3] Kimberly Patch. It's All about the Interface: Speech Recognition That Works for Both Human and Computer. MIT IAP Seminar. January 17, 2006. (<http://www.redstartsystems.com/papers.html>)
- [4] Kimberly Patch. Utter Command: Human-Machine Linguistics, Human-Machine Grammar, and a New Interface. Boston Voice Users Group. July 12, 2005. (<http://www.redstartsystems.com/papers.html>)
- [5] Kimberly Patch. An Intentionally Modest Proposal for a Speech Recognition Command Grammar. Boston Voice Users Group. February 10, 2004. (<http://www.redstartsystems.com/papers.html>)

8. FURTHER REFERENCES

The following are key papers from the Carnegie Mellon University Universal Speech Interfaces project (www.cs.cmu.edu/~usi/):

- Stefanie Tomko and Roni Rosenfeld. Speech Graffiti habitability: What do users really say? Proc. SIGDIAL, Boston, MA, 2004.
- Roni Rosenfeld, Dan Olsen and Alexander Rudnicky. Universal Speech Interfaces. *Interactions*, VIII(6), 2001, pp. 34-44.
- Stefanie Shriver, Roni Rosenfeld, Xiaojin Zhu, Arthur Toth, Alex Rudnicky, Markus Flueckiger. Universalizing Speech: Notes from the USI Project. In Proc. Eurospeech 2001.
- Stefanie Shriver, Arthur Toth, Xiaojin Zhu, Alex Rudnicky, Roni Rosenfeld. A Unified Design for Human-Machine Voice Interaction. In Proc. CHI 2001.
- Ronald Rosenfeld, Xiaojin Zhu, Stefanie Shriver, Arthur Toth, Kevin Lenzo, Alan W Black. Towards a Universal Speech Interface. In Proc. ICSLP 2000.

The following paper describes a speech interface based on a structured grammar that includes many made-up words:

- Nils Klarlund. Editing by Voice and the Role of Sequential Symbol Systems for Improved Human-to-Computer Information Rates. IEEE International Conference on Multimedia & Expo (ICME). Baltimore, Maryland. July 6-9, 2003 (www.clairgrove.com/papers/fallacy.pdf).

The following paper describes the cognitive load imposed by speech recognition and explains some of the differences between human-human communications and human-computer communications:

- Ben Shneiderman. The Limits of Speech Recognition. *Communications of the ACM*. September 2000. Vol. 43, No. 9 63. (www.cs.umd.edu/~ben/p63-shneidermanSept2000CACMf.pdf)

The following books have informed Human-Machine Grammar:

- Words and Rules*, Stephen Pinker, Basic Books, New York, NY, September 1999
- Linked: The New Science of Networks*, Albert-Laszlo Barabasi, Perseus Books Group, New York, NY, May 2002
- The Psychology of Everyday Things*, Donald Norman, Basic Books, New York, NY, April 1988
- The Humane Interface: New Directions for Designing Interactive Systems*, Jef Raskin, Addison-Wesley Professional, Boston, MA, March, 2000