Redstart Systems President Kimberly Patch gave this talk at the SpeechTek 2006 Advanced Technologies Speech Symposium: Advanced Techniques for Improving User Interaction. You can download this talk and the accompanying slides at www.redstartsystems.com. We welcome your comments on this talk or anything related at www.redstartsystems.com/contact.html.

When Natural Language isn't: the need for a dedicated speech interface

Before I get started I'd like to ask you some questions.

I need a show of hands, how many people here use speech recognition software on the desktop? How many who use speech on the desktop use it mostly for dictation? How many also use speech to control the computer?

How many who use speech to control the computer use it 100 percent of the time -- hands-free? At least 80 percent of the time? At least half the time? Twenty percent of the time?

When I asked these questions at an MIT seminar attended in part by people who use speech because of hand injuries, I got a similar answer. Today, very few people, even motivated people, really use speech to control the computer.

Speech on the desktop showed great promise a dozen years ago.

SLIDE 2: Optimistic headlines from the '90's

But it hasn't yet lived up to that promise. The question is why. The answer can not only improve speech on the desktop, it can improve human-machine communication in general.

Speech control of the desktop has not lived up to expectations because

SLIDE 3: The wrong tool for the job

we're using the wrong tool for the job.

We learn to speak before we're two, and we're good at it. Most of our experience, however, is in human-human communication. Communicating with a computer is different.

SLIDES 4-6: Communication Skills Comparison

Humans recognize words, and computers recognize words. But this is where the similarities end. Humans understand word meanings, but today's desktop computers do not. Humans are adept at adapting language on-the-fly, but computers are not.

Because humans can adapt, we naturally work out efficient language when we're doing the same thing over and over again -- when a restaurant worker tells the cook that someone's asked for two orders of french fries, an efficient communication like "2 fry" is easily understood.

SLIDE 7: 2 Fry

There's also a curveball in the mix -- expectations.

Slide 8: Communication Skills Comparison

Fictional computers like Hal appear to understand word meanings and adapt language on-the-fly... unlike today's desktop computers.

Given this situation our usual human-human communication style doesn't work well between human and computer. What to do? There are two possibilities.

SLIDE 9: Fake It

The first approach is to fake it -- use long lists of synonymous commands to make it look like the computer understands and adapts.

This pseudo-natural-language approach has been a mainstay of desktop speech software.

And, as you know, it hasn't worked very well. The approach has several drawbacks.

SLIDE 10: NaturallySpeaking's commands to move the cursor to the beginning of the line

First, even long lists of synonymous commands -- "go to the beginning of the line", "go to beginning of line", "go to start of line", "move to top of line" etc. -- are not enough. A user will invariably try something you haven't thought of -- "Go Home", perhaps, or "jump to front".

SLIDES 11-13: Drawbacks to the pseudo natural-language approach

Second, with many ways to say a command users tend not to get into habits. Habits are important, because they allow action without much cognitive effort.

The third drawback is the largest one, but it's hidden: synonymous commands don't lend themselves to being combined.

I'm going to take a little more time to explain this last drawback in detail.

We know language can be efficient -- Two Fry. How can we bring this kind of efficiency to the human-computer relationship? Not just by speaking the keys or clickable screen elements we're accustomed to.

Next time you're in a room with someone who's flying along on the keyboard and mouse take a look at what that person's actually doing -- often many clicks -- to carry out a task like navigating to a particular file. Most of this clicking is accommodating the computer -- keys and screen space for clickable items are limited.

SLIDE 14-15: If the human doesn't have to think

But if the human doesn't have to think between steps, there's no need for separate steps other than to accommodate the computer.

Let me repeat that.

If the human doesn't have to think between steps, there's no need for separate steps. In theory, speech has no real estate problem -- there are more than enough unique phrases for commands. When you try to tap into this promise, however, you run up against a pair of challenges.

SLIDE 16-17: Speech interface challenges

How do you remember commands? We'll come back around to this challenge a little later. And how do you enable combined commands without running into an exponential problem?

SLIDE 18: The Exponential Explosion

Working with only 20 commands, if you enable 10 wordings for each and 4-command combinations you've got an untenable 1.5 billion possibilities. Without synonymous wordings it's four orders of magnitude lower -- 116,000.

You need to combine many more than 20 commands to control a computer. The pseudo-natural language approach is incompatible with the need to combine commands.

This brings us to the second possibility -- structured grammar.

SLIDE 19: The solution: a Structured Grammar

Instead of using the computer to fake natural language, tap the human's natural ability to adapt. Work out a grammar ahead of time. Use the right tool for the job.

Carnegie Mellon researchers did a study comparing speed and ease-of-use of structured grammar vs. natural language. The results are telling.

SLIDE 20: Carnegie Mellon study

They found that 74% of users prefer a structured rather than natural language approach to speech recognition

There is also interface precedent along these lines -- a computer keyboard comes in a standard configuration rather than a box of keys you can arrange any way you wish. Here's another -- Palm's Graffiti writing system uses a standard way of writing each letter rather than accommodating multiple letter shapes.

My company, Redstart Systems, has developed a structured grammar -- Human-Machine Grammar, and we have implemented it in speech interface software called Utter Command.

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

SLIDE 21-23: Important Factors

How the brain processes words, the memory phenomenon of chunking, and the network phenomenon known as six degrees of separation.

SLIDE 24-26: Human-Machine Grammar

Human-Machine Grammar has no synonyms, uses logical rules to minimize wording possibilities, and follows the way the human brain uses language.

One grammar rule that's key to cutting command possibilities is to say computer steps in the order they'll be carried out. In general, you tend to select an object, then carry out an action on the object, for instance "Line Bold", or "Window Close".

SLIDE 27: Asking Computer/Commanding Objects

This dovetails with the key to cognitive ease in talking to a computer: don't ask the computer to do something for you, instead, talk directly to the objects on the computer screen, for instance **"Window Close"**, or **"3 Lines"**

This concise grammar eliminates the drawbacks we talked about earlier

SLIDE 28: Structured Grammar's advantage

It is unambiguous, builds habits and enables combined commands. These attributes also meet the first key challenge we talked about earlier.

SLIDE 31: How do you remember commands?

Structured grammar makes commands more memorable. Commands that are easily remembered and can be combined enable the real power of speech -- efficiency beyond the keyboard and mouse.

SLIDE 32: Editing

These next four slides show the number of steps needed to carry out several tasks using the keyboard and/or mouse, the NaturallySpeaking pseudo-natural language interface, and the Utter Command structured grammar interface.

Take a look at the third example -- bolding one word before and two words after the cursor. This takes
5 keyboard steps,
3 mouse and keyboard steps,
4 NatSpeak steps,
or 1 step using Utter Command

This pattern is the same across a wide variety of categories -- editing,

SLIDE 33: Moving the mouse

moving the mouse arrow

(2 Count)

opening folders and files

SLIDE 34: Opening folders and files

In some categories the difference widens dramatically. Look at the second example -- opening a file -- 15 key clicks, 9 mouse clicks, 19 NatSpeak commands, but just one step using Utter Command.

The same goes for preparing email, and accessing the Web.

SLIDE 36: Classic Computer Interface Challenges

Now I'll show you how Utter Command addresses four classic computer interface challenges.

The first challenge is easy folder and file access. The version of Utter Command I'm using runs on top of the NaturallySpeaking speech engine and is now in Beta testing.

I'll first call up a folder named "UC Demo" in Windows Explorer, then I'll call up the same folder in Excel.

Windows UC Demo Folder
Excel UC Demo Folder
Window Close Times 3
3 Seconds Break

Utter Command also allows you to call up files directly.

Demo One File
Window Close Off

The second challenge is keeping track of how to execute the same command in different programs. Who knows what you have to click to enter the date in Excel? How about Notepad? How about Word? They're all different.

Utter Command uses commands that are the same across all programs. We'll now use the command Date Short to paste the date into Excel, Notepad, and Word.

Speech On Excel Open
Date Short
Notepad Open
Date Short
Word Open
Date Short
Window 2 Off

The third challenge is keeping command steps to a minimum. To close these windows using a mouse or keyboard I have to put the focus on the window I want, close the window, and sometimes take a third step to tell the computer whether or not I want to save the file. I can combine these instructions using speech.

First we'll switch to a view where you can see the task bar -- watch closely, these happen quickly.

Speech On Notepad Open
Excel Close No
Word Close No
6 Seconds Break

Speech combinations speed making changes to text and speed repetitive tasks like formatting.

Window Max
Line Delete
5 Enter
If the human doesn't have to think between steps there's no need
3 Befores Delete
5th Word 2 Words Delete
1 Undo

Line duplicate Repeat 10
Down Enter Repeat 10
Up Backspace Repeat 10
Home Hyphen Space
Down Home Hyphen Space Repeat 10
Speech Off

Now we'll use a single speech command to size two windows, specifying where the windows meet.

Speech On Wordpad Open
Window 3 4 X 70
Window Close
4 Seconds Break

Now we'll change the size of the remaining window, then move it a couple of times.

Size 50 By 50
Window 20 By 20
Window 0 By 50
Window Close No Off

The fourth challenge is fluid switching among programs. Using speech you often need just a single command to carry out an action no matter where the program focus is.

I'll demonstrate a few of these now -- we'll adjust the volume, set a reminder, and access the Internet.

Speech On
Volume 50
Volume 90
3 Seconds Meeting
Google Site
Slash Dot Site
Weather Site
10 Seconds Break

You can also combine some of these advantages. Here's a command that addresses an email message to several recipients, adds an opening, and you can call it up from within any application.

Express Bill Lisa CC Eric
Window Close No
Window Close Times 2 Microphone Off

With the right interface you can do anything by speech that you can with the keyboard and mouse -- often faster.

SLIDE 37: A structured grammar makes commands

To sum things up, structured grammar makes commands easy to remember and use, and enables combined commands. It's the right tool for the human-machine interface.

SLIDE 38: For more information

For more information, feel free to contact me at kim@redstartsystems.com.

The Human-Machine Grammar rules and dictionary are freely available at the Redstart Systems Site at www.redstartsystems.com.

There are also talks posted on the site that go into more detail about the thinking and science behind Human-Machine Grammar.

And there's also advice for software makers who want to make sure they are ready for speech input.

We're encouraging everyone who writes speech commands to use Human-Machine Grammar.

What you've seen here today is only the tip of the iceberg. When you go to our site you'll find a mature structured grammar that was developed over the past decade in a real-world environment. Although Utter Command allows you to do anything on the computer that you can using the keyboard and mouse we think of it as a baseline technology that will enable much more.

Thank you very much.