Where Does Speech Input Make Sense?
Lessons from the Desktop

Position Paper:
AVIxD 2011 Workshop on Voice Interaction Design -- Where Does Speech Input Make Sense? Lessons from the Desktop

Kimberly Patch
President
Redstart Systems
kim@redstartsystems

A little background

I've lived in the desktop speech arena for the past 16 years. I've used speech, largely hands-free, for my daily work during that time, and I've worked with a lot of other users who need to use speech hands-free. More recently I've worked with doctors who are aiming to speed report generation using speech.

There's a wealth of real-world information in these communities that bears on the question of where speech input makes sense in any context.

Three barriers

We see three main barriers to enabling people with desktop speech recognition:

1. Inability to remember commands -- this is a showstopper. And standard commands aren't necessarily the answer. It's proven difficult to hammer out standard commands partly because they're inappropriate -- speech is still a moving target. Esperanto didn't work either.

2. Fatigue -- vocal and mental.

3. Fear of something going wrong. Speech input has been around long enough that many people have personal experience with something going wrong.

The question

I think the best way to answer the question of where speech input makes the most sense is to give users the same structure or better than we've given them with other input methods, then let the speech input method evolve.

A lot of what we've done is fill the gaps of a flawed system so that users who must use speech can carry out any task by speech in a practical way. Doing this has allowed us to observe people who are generally using speech recognition for all the tasks that they do on the desktop, and identify places where speech can be much faster than the keyboard and mouse -- document, folder and Web navigation, for instance.

We've also been able to observe mixed input.

We've seen people exploit commands in unusual ways.

Here are two examples: the Dragon spell command is only available in places the developers thought you would want to spell, which is generally text input. I've received many complaints from people trying to use the spell command to go through menus more quickly.

Dragon has an in-line command for "New Line", a command that just hits the Enter key. This is so you can say some text and then, without pausing, put the cursor on a new line. I have users who are using this command to go through files -- they say the name of a file and then "New Line" so they don't have to pause before saying Enter.

As designers I think we need to:

- Make full use of all available input methods, make all available input methods discoverable and efficient, then collect information about what users do.

- Distinguish between command set structure and exact command. Command set structure is akin to arranging keyboard keys in an order that makes sense. It creates a framework that makes everything else easier.

- Make commands highly discoverable -- in some cases in multiple ways. In a lot of cases audio prompts would be more useful if the user had a good mental map of the entire system -- or the parts they're likely to use. A map of speech commands can give users an orientation to a new system that makes the experience better forever after because it sets a context so they know what to expect. And in our hyper-connected world, it's becoming increasingly possible for many users to easily look up a map of speech commands for any system.

- Exploit full multimodal, e.g. "I want to go from [pointing gesture] to [pointing gesture]", but also simpler ways of using multiple input methods at once, like selecting text by touch but typing by speech.