Java Speech Grammar Format

The SDK supports command and control grammars specified in a JSGF file. The file defines the grammar using rules and allows to define a less restrictive grammar than what a string list can do. The full rule structure is defined as a W3C standard at http://www.w3.org/TR/jsgf. This section shows a few key methods and examples.

Suppose you want to recognize a sequence of three digits. While the string list approach could be used to specify an appropriate grammar, the grammar would consist of one thousand commands. Needless to say, this is difficult to write and maintain. The JSGF file that recognizes a sequence of three digits contains the following lines:

```
#JSGF V1.0;
grammar ThreeDigits;
public <ThreeDigits> = <digit> <digit> <digit>;
<digit> = ( zero | one | two | three | four | five | six | seven | eight | nine )
```

Each line is terminated with a semicolon. The first line contains the required JSGF identifier. This is required for all JSGF grammars. The next line defines the grammar name.

The rules of the grammar are defined in angle brackets: "<", ">". The first rule is a public rule, that is, it is a rule that is exposed externally from the grammar. Public rules are the sentences or word sequences that the grammar may be used to recognize.

The <ThreeDigits> rule is defined as a sequence of three of the <digits> rules. This means that there needs to be three things matching the <digits> rule to match the <ThreeDigits> rule.

The <digits> rule uses the or operator "|" to define a list of alternatives, and the parentheses "(", and ")" are used to delimit this list. Thus the <digit> rule can match any of the digits from zero to nine.

The JSGF file can specify repeating of words. The following example shows how to recognize a sequence of digits of any length:

```
#JSGF V1.0;
grammar Digits;
public <Digits> = ( <digit> ) + ;
<digit> = ( zero | one | two | three | four | five | six | seven | eight | nine )
```

The new element here is the plus sign "+" at the end of the <Digits> rule. This means that what comes before it should occur one or more times. In this case, what comes before it is what is in the parentheses. Also available in JSGF is the asterisk, "*", which means that what comes before it should occur zero or more times.

As a final demonstration of JSGF, the following example shows how to use grammar tags. Grammar tags are used to locate recognized words or phrases: