Integration von Spracherkennung in einem automotive Multimedia System

Dr. Karim Belhoula [Abdelkarim.belhoula@siemens.com]
Siemens VDO Automotive

Einleitung

Der Bedarf an Spracherkennung im Fahrzeug steigt mit dem Einsatz von modernen Multimediasystemen. Solche Systeme beinhalten eine Fülle von Applikationen die mit herkömmlichen Eingabegeräten schwer zu bedienen sind. Im Rahmen des EU-Projektes INFORM wurde eine JAVA-basierende Multimedia-Platform entwickelt, die es erlaubt Dienste die von verschiedenen Dienstanbietern angeboten werden, zum Fahrzeug zu übertragen und zu installieren. Kern der Mensch-Maschine-Schnittstelle im Fahrzeug bildet ein Spracherkennungssystem, der für die Bedienung von allen Applikationen eingesetzt wurde. In diesem Beitrag wird die Einbindung der Spracherkennung in dem Fahrzeug-Terminal erläutert. Den Ausgangspunkt bildet ein sprecherunabhängiger Spracherkenner mit einer herstellerspezifischen Schnittstelle. Um die Austauschbarkeit der Spracherkenner zu gewährleisten, wurde eine standardisierte JAVA-Schnittstelle implementiert. Ein Spracherkennungsdienst der die Priorisierung zwischen der Platformapplikationen berücksichtigt, wurde entwickelt.

Anforderung an die Spracherkennung im Fahrzeug

Spracherkennung im Fahrzeug stellt nicht nur eine neue Option mit der ein Fahrer seine Geräte wie Navigation, Radio, Telefon uvm bequem bedienen kann, sondern sie trägt dazu bei, die Sicherheit im Strassenverkehr zu steigern. Die Hände des Autofahrers werden somit befreit vom Suchen und Betätigen der richtigen Knöpfe am Armaturenbrett. Die akustische Umgebung im Auto ist geprägt durch Innen- und Außengeräusche, die eine Herausforderung für den Einsatz eines Spracherkenners darstellt. Der Geräuschpegel in einem Auto der Mittelklasse liegt bei 50 km/h zwischen 55 und 58 dB(A) und steigt bei 130 km/h bis zu 75 dB(A). Hinzu kommt die Änderung der Sprachsignalerzeugung in einer Lauten Umgebung; Der sogenannte Lombard Effekt. Aus diesen genannten Gründen ist es erforderlich, einen Spracherkenner mit robuster Geräuschreduktion und Echo-Eliminierung einzusetzen. Weiterhin ist es sehr vorteilhaft eine einfache Austauschbarkeit der Spracherkenner im Gesamtsystem zu gewährleisten. Auf diese Weise lassen sich unterschiedliche kundenorientierte Systeme in kurzer Zeit realisieren. Diese Anforderung kann erfüllt werden durch die Unterstützung einer Standardschnittstelle wie SAPI oder JSAPI. Eine weitere Problematik entsteht wenn der Spracherkenner von mehr als einer Applikation benutzt wird. An dieser Stelle ist ein Konzept für die Verwaltung der Spracherkenner als Systemressource sehr hilfreich.

Zuletzt ist zu beachten dass eine Synchronisation stattfinden muss, wenn andere Eingabegeräte neben der Spracherkennung im System benutzt werden.

Beschreibung des Spracherkenners

Der eingesetzte Spracherkenner namens VOCON stammt von Philips Speech Processing. Hierbei handelt es sich um einen modular aufgebauten Erkenner, der leicht für automotive Zwecke anzupassen ist. Mit seinem geringen Speicherbedarf lässt er sich in automotive Systeme problemlos integrieren.
VOCON verfügt über die folgenden Hauptmerkmale:

- Sprecherunabhängige und Sprecher abhängige Erkennung für Einzelwörter und kontinuierliche Sprache
- Wort- und Phonembasierte Erkennung
- Verbesserte Geräuschreduktion und Echo-Eliminierung
- Schlüsselwort-Aktivierung / Schlüsselwort - Erkennung
- Leichte Portabilität auf unterschiedliche Betriebssysteme

Der Erkenner wird durch eine Quasi-Standardschnittstelle namens VOCAPI gesteuert. Mit der Hilfe eines zusätzlichen Werkzeugs (Vocon Designer) können Applikationsentwickler die notwendigen Vokabularien generieren.

Was ist JSAPI?

JSAPI (Java Speech API) ist eine offene standardisierte Schnittstelle, für die Einbindung von Spracherkenner (command & control und Diktiersysteme) und Text-to-speech-Systeme in Applikationen. Dieses API wurde im Zusammenarbeit mit Sun Microsystems von folgenden Sprachtechnologie- und IT-Führern ausgearbeitet: Apple Computer, Inc. AT&T, Dragon Systems, Inc. IBM Corporation, Novell, Inc., Philips Speech Processing und Texas Instruments Incorporated. Implementierungen dieser Schnittstelle existieren bereits von IBM (basierend auf deren ViaVoice Produkt) und von Lernout & Hauspie (ASR 1600). Die Vorteile von JSAPI bestehen darin, den einfachen Austausch und das Hinzufügen von Spracherkennern und TTS zu ermöglichen. Hierbei kann es sich um Spracherkenner handeln die nativ sind (C, C++) oder um reine Java Module (siehe Abbildung 1). Für die Erstellung und Verwaltung von Vokabularien wurden auch in diesem Zusammenhang die APIs Java Speech Grammar Format (JSGF) und Java Speech Markup Language (JSML) entwickelt.

Im Rahmen des MOTIV Projektes wurde von Siemens VDO die JSAPI Schnittstelle für den VOCON Spracherkenner implementiert und getestet.

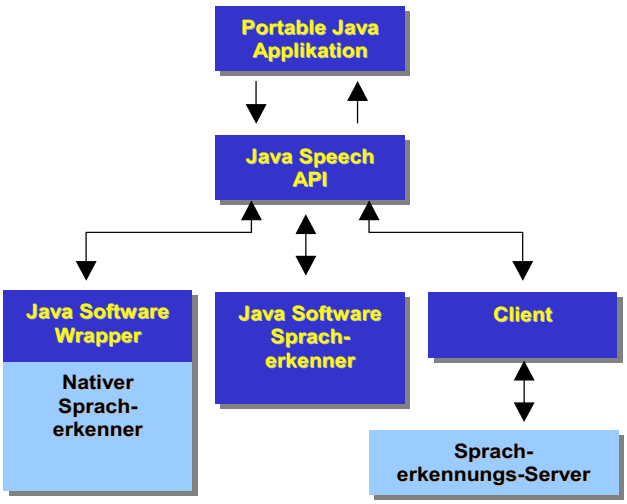

Abbildung 1. ***Anbindung von Spracherkennern an JSAPI***

Die Hauptfunktionen der JSAPI Schnittstelle sind:

- Erzeugung des benötigten Spracherkenners oder Sprachsynthetisators (z.B mit der gewünschten Sprache)
- Allokierung der Ressourcen für den Erkenner
- Laden des entsprechenden Vokabulars
- Aktivierung des Vokabulars
- Starten des Erkenners
- Stoppen des Erkenners
- Deallokierung des Erkenners

Mit Hilfe eines sogenannten „Event Listner“ werden alle Zustände des Erkenners an die sprachgesteurte Applikation mitgeteilt.

Zugriffsverwaltung auf dem Erkenner

Da der Spracherkenner im Gesamtsystem von mehreren Applikationen benutzt wird, ist es sehr vorteilhaft ein Konzept zu entwickeln, dass das Teilen dieser Ressource regeln soll. Die Plattform, der das INFORM-Projekt zugrunde lag, basierte auf eine Client/Server Architektur, in der jede Applikation für sich ein Dienst (Service) darstellt. So besteht beispielsweise die Navigation u.a. aus verschiedenen Services wie den Routenplaner, die Kartendarstellung, die Positionsbestimmung. Alle Dienste werden im System registriert und können somit von anderen Applikationen benutzt werden. Die Generierung und das Allokieren eines Dienstes wird unterstützt durch einen sog. Server. Von einem und demselben Dienst lassen sich mehrere Instanzen bilden. Nach diesem Modell wurde ein Spracherkennungsdienst implementiert. Der Ablauf von der Dienstnachfrage bis zur Benutzung läuft wie folgt (siehe Abbildung 2):

1. Ein Spracherkennungsdienst anfordern
2. Der Dienst wird zur Verfügung gestellt falls er bereits existiert.
3. Ist keine Instanz von dem Spracherkennungsdienst vorhanden, so wird ein Spracherkennungsserver generiert.
4. Der Server erzeugt dann eine Instanz von dem Spracherkennungsdienst
5. Der Dienst wird für die Applikation freigegeben.

Ein Spracherkenner kann auch mit einer bestimmten Priorität angefordert werden. In diesem Fall wird die höher priorisierte Applikation den Erkenner allokieren auch wenn er in Benutzung ist. Der Einfachheit halber wurde in INFORM der Grundsatz „first come, first serve“ verfolgt.

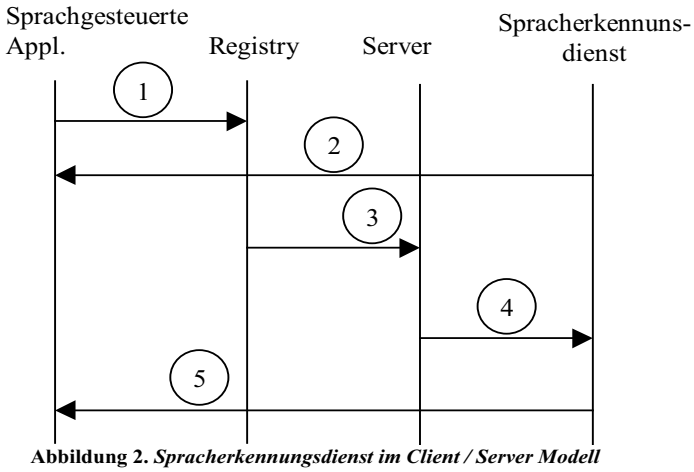

Abbildung 2. ***Spracherkennungsdienst im Client / Server Modell***

Graphische UI vs. Speech UI

Der Startpunkt in der graphischen Benutzerschnittstelle des Systems ist ein Hauptmenü aus dem die unterschiedlichen Applikationen gestartet werden können. Jede Applikation besteht aus einer oder mehreren visuellen Formen, die tief verschachtelt werden können. Um die Erkennungstrefferquote zu verbessern, aktiviert die Applikation immer nur das Vokabular der Form, die den Fokus besitzt. Neben diesen fokusabhängigen Vokabularien, existiert ein globales Vokabular, das immer aktiv ist. Zu diesem Vokabular gehört beispielsweise das Kommando „*Hauptmenü*“ mit dem aus jedem beliebigen Zustand zum Menü zurückgekehrt werden kann. Die Befehle „*Lauter*“ und „*Leiser*“ für die Steuerung des Radios gehören ebenfalls zu dem globalen Vokabular und werden ausgeführt unabhängig davon, welche Applikation den Fokus hat. Neben der Bedienung per Sprache kann der Benutzer zusätzlich Eingaben über externe Tasten oder Touchscreen machen. Dadurch werden die Applikationszustände verändert. Deshalb ist es erforderlich die Umschaltung der Vokabularien entsprechend zu synchronisieren. Diese Aufgabe übernimmt ein sog. „Präsentation Manager“, der die Applikationen über derartige Zustandsänderungen benachrichtigt.

Bibliographie

[1] D. Langmann et. Al: CDSC – The Motiv Car Speech Data Collection, International Conference on Language Resources Engineering, Granada, Spain, May 1998

[2] Langmann, D., Fischer, A., Wuppermann, F., Haeb-Umbach, R., Eisele, T. (1997). Acoustic Front Ends for Speaker-Independent Digit Recognition in Car Environments. In Proceedings of EUROSPEECH'97 (pp. 2571--2574). Rhodes, Greece.

[3] Sun Microsystem Inc. "Java Speech API Programmers Guide" Version 1.0 http://java.sun.com/products/java-media/speech/