Sprach-Suche in einer Musikbibliothek mit klassischer Musik mit der Spracherkennungsbibliothek Microsoft.Speech

Bachelorarbeit

Angewandte Informationswissenschaften (AIW)

Fakultät für Informations- und Kommunikationswissenschaften

Technische Hochschule Köln

vorgelegt von:

Michael Jonathan Janz

am 31.08.2016 bei Prof. Dr. rer. nat. Selma Strahringer

Technology
Arts Sciences
TH Köln

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde.

Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Dies gilt auch für Quellen aus eigenen Arbeiten.

Ich versichere, dass ich diese Arbeit oder nicht zitierte Teile daraus vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

Mir ist bekannt, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs mittels einer Plagiatserkennungssoftware auf ungekennzeichnete Übernahme von fremdem geistigem Eigentum überprüft werden kann.

Ort, den _____

Michael Jonathan Janz

Abstract

Diese Bachelorarbeit befasst sich mit der Thematik der Sprachsuche in einer Musikbibliothek mit klassischer Musik. Klassische Musik hat eine Besonderheit im Gegensatz zu Musik aus anderen Genres. Während beispielsweise bei Rockmusik in der Regel die Band und der Name des Musikstückes die Aufführung bestimmt, spielen bei klassischer Musik durchaus mehrere Faktoren eine Rolle, welche die Aufführung definieren. Dazu gehören nicht nur der Titel des Stückes und die Musiker, in dem Fall das Orchester, sondern auch der Komponist, Solisten und der Dirigent. Diese Arbeit hat das Ziel zu zeigen, wie eine Sprachsuche für eine Musikdatenbank mit klassischer Musik gestaltet werden kann und wie sich diese praktisch realisieren lässt. Hierfür werden die Fähigkeiten der Spracherkennungstechnologie von Microsoft, die Microsoft Speech Platform, untersucht und Datenbanktechnologien besprochen, welche für das Erreichen dieses Ziels relevant sind.

Schlagworte: Spracherkennung, Microsoft Speech, Speech Recognition Grammar Specification, Sprachassistent, Sprachsteuerung

Inhaltsverzeichnis

Abkürzungsverzeichnis

CALO ..... *Cognitive Assistant that learns and organizes*

HAL ..... *Heuristischer Algorithmus*

IPA ..... *International Phonetic Alphabet*

Op. ..... *Opus - Werksnummer*

PLS ..... *Pronunciation Lexicon Specification*

SAPI ..... *System API*

SRGS ..... *Speech Recognition Grammar Specification*

SSML ..... *Speech Synthesis Markup Language*

1 Einleitung

Im Laufe der Digitalisierung wird Arbeit immer mehr von Maschinen verrichtet.[1] Künstliche Intelligenzen übernehmen die Planung von Meetings, scannen Emails nach Informationen oder versorgen den Nutzer mit Informationen.[2][3] Große Unternehmen wie Google, Microsoft oder Apple sind in diesem Feld aktiv und bieten mit ihren Lösungen verschiedene Sprachassistenten, mit unterschiedlichem Leistungsumfang an. Während Apple mit Siri bereits seit 2011 auf dem Markt ist, folgte Google 2012. Microsoft startete den Sprachassistenten Cortana erst 2014, obwohl Microsoft bereits seit Windows XP ein Sprachsteuerungsmodul für den privaten Nutzer anbietet.[4] Seitdem hat Microsoft seine Sprachtechnologien weiterentwickelt, wovon auch die Spracherkennung von Cortana profitiert hat. Microsoft bietet hier für Entwickler die Microsoft Speech Platform und die systemeigene System API (SAPI) für *.NET* kostenlos an. Die Hersteller zeigen hiermit ein breites Interesse und dass Sprachsteuerungen in dem Alltag von Computernutzern angekommen sind. Dies macht dieses Gebiet für zukünftige Entwicklungen interessant.[5]

Klassische Musik hat eine Besonderheit im Gegensatz zu Musik aus anderen Genres. Während beispielsweise bei Rockmusik in der Regel die Band und der Name des Musikstückes die Aufführung bestimmt, spielen bei klassischer Musik durchaus mehrere Faktoren eine Rolle, welche die Aufführung definieren. Dazu gehören nicht nur der Titel des Stückes und die Musiker, in dem Fall das Orchester, sondern auch der Komponist, Solisten und der Dirigent. Musikstücke führen häufig den selben Namen oder besitzen mehrere Namen. Auch unterscheiden sich einige Stücke von der Form der Aufführung deutlich, obwohl es sich inhaltlich um dasselbe Stück handelt, jedoch der

[1] Vgl. (Spät, 2015)
[2] Vgl. (Bosker, 2013)
[3] Vgl. (ZEIT ONLINE, 2016)
[4] Vgl. (Microsoft, 2015)
[5] Vgl. (Bosker, 2013)

Dirigent aufgrund seines Orchesters das Stück nur eingeschränkt spielen kann oder er eine besondere Inszenierung des Stückes gewählt hat.

Um die Aufführung eines Musikstückes exakt zu spezifizieren, was besonders für Kenner von klassischer Musik von Interesse ist, werden mindestens die Daten „Komponist“, “Dirigent“, “Solist“, “Orchester“ und „Titel“ benötigt. Eine Einbindung dieser Daten findet jedoch bereits bei einigen Anbietern nicht statt und verhindert somit eine hochwertige Suche in Musikdatenbanken über Spracheingabe. Diese Arbeit hat das Ziel zu zeigen, wie eine Sprachsuche für eine Musikdatenbank mit klassischer Musik gestaltet werden kann und wie sich diese praktisch realisieren lässt. Zusätzlich werden die Möglichkeiten analysiert, welche sich aus der Nutzung dieser Attribute ergeben. Hierfür werden die Fähigkeiten der Spracherkennungstechnologie von Microsoft, die Microsoft Speech Platform, untersucht und Datenbanktechnologien besprochen, welche für das Erreichen dieses Ziels relevant sind. Hierbei wird die Gestaltung des Suchbefehls auf Basis dieser Attribute im Vordergrund stehen. Somit stellt diese Arbeit einen Leitfaden dar, nach deren Leitung eine Anwendung, für die sprachliche Suche in einer Musikdatenbank mit klassischer Musik, erstellt werden kann

1.1 Motivation

Einige Nutzer wünschen sich einen Sprachassistenten, der immer mehr Aufgaben übernehmen kann und ihnen unangenehme Aufgaben abnimmt. Der Anwendungsbereich eines solchen Sprachassistenten ist theoretisch unbegrenzt, daher ist dies ein interessantes und realitätsnahes Thema, wofür der Autor dieser Arbeit sich in besonderem Maße interessiert. Der ein oder andere Leser dieser Arbeit erinnert sich an die Fernsehserie „Star Trek The Next Generation“, wo ein sprachgesteuerter Computer verschiedene Aufgaben übernehmen konnte und einem ausgereiften Werkzeug glich.

Viele Fernsehserien, welche sich mit der Zukunft beschäftigen galten als Vorbilder, den damals gesetzten technischen Stand auch zu erreichen. Dies sieht man gut an dem Film „Zurück in die Zukunft“ aus dem Jahre 1985, in dem im Jahre 2015 die Menschen sich mit Hoverboards fortbewegten. Einige Firmen haben sich das Ziel gesetzt,

über dem Boden schwebende Hoverboards auf den Markt zu bringen. Dieses hochgesteckte Ziel wurde als technische Vorschau möglich gemacht, aber für den privaten Anwender nur soweit realisiert, dass sich nun motorisierte Hoverboards als Fortbewegungsmittel nutzen lassen.[6] Der Autor dieser Arbeit möchte hiermit einen kleinen Teil für die Forschung in diesem Bereich beitragen und hat ein besonderes Interesse, dass eine Sprachsteuerung wie in „Star Trek" eines Tages Realität wird.

1.2 Abgrenzung

Diese Arbeit wird im Rahmen des Bachelorabschlusses des Studiengangs Angewandte Informationswissenschaft verfasst. Daher wird in dieser Arbeit ein Fokus auf die Theorie und Praxis der technischen Umsetzung gelegt. Das Ziel dieser Arbeit ist zu klären, wie eine Sprachsuche mit der Microsoft Speech Platform in einer Musikbibliothek in Form mit klassischer Musik stattfinden kann. Die Theorie der Spracherkennung wird in Form der Microsoft Speech Platform in ausführlicher Weise beschrieben. Die Tiefe richtet sich nach dem praktischen Nutzen. Somit werden Grundlagen der Spracherkennung wie beispielsweise die Umwandlung von Audiodaten in Text mithilfe von Algorithmen, nur in begrenztem Maß behandelt. Hierzu werden an einigen Stellen theoretische Fähigkeiten der Microsoft Speech Platform nicht praktisch behandelt, da die Nutzung dieser den Umfang der vorliegenden Arbeit überschreitet. Dies ist in den jeweiligen Kapiteln gesondert gekennzeichnet. Eine Erweiterung der Anwendung um andere Sprachen würde neue Herausforderungen in der sprachlichen Gestaltung und der Auswahl der korrekten Technologie bedeuten und somit den Rahmen dieser Bachelorarbeit bei Weitem übersteigen.

Klassische Musik hat viele Attribute, die eine Vorstellung beschreiben. Um diese große Datenmenge für diese Arbeit zu reduzieren, wird die Datenmenge auf die Informationen der Hauptakteure der Aufführung beschränkt. Dies trifft auf die fünf Attribute: „Titel", „Komponist", „Dirigent", „Solist" und „Orchester" zu, mit deren Hilfe ein Musikstück eindeutig identifiziert werden kann.

[6] Vgl. (Thompson, 2015)

Beispielsweise erzwingt die Verwendung des Feldes „Erstelldatum“ eine Auseinandersetzung mit musikalischen Epochen und deren Klassifikationen damit auch Eingaben, wie „Spiele etwas aus der frühen Romantik“ verstanden werden könnten. Ein anderes Beispiel wäre die Stimmung eines Stückes, sodass das Spracherkennungssystem beispielsweise ein fröhliches Musikstück abspielen kann. Hierfür muss geklärt werden, wodurch sich ein fröhliches Musikstück in der klassischen Musik definiert und ob sich diese Definition nach Komponisten, Epochen oder unter anderen Einflüssen verschieden ist. Eine Klärung dieser Fragen erweitert zwar die Fähigkeiten einer Sprachsteuerung einer Musikbibliothek, ist jedoch nicht Thema dieser Arbeit. Über die klassische Musik gibt es viel zu schreiben, was auch den Inhalt dieser Arbeit bereichern würde, dies würde den Rahmen in Kombination mit der technischen Umsetzung jedoch bei Weitem überschreiten. Somit hat diese Arbeit nicht den Anspruch, sich ausführlich mit Fachspezifika der Musik zu beschäftigen und reißt diese daher nur an.

Der Erfolg einer Spracherkennung ist auch von einer entsprechenden Hardware abhängig. Wird ein minderwertiges Mikrofon verwendet, kann dies einen negativen Einfluss auf den Erfolg der Spracherkennung haben. Eine Auseinandersetzung mit dieser Problematik benötigt Tests und einen tieferen Einstieg in Technologie von Aufnahmegeräten. Diese Tiefe kann diese Arbeit nicht leisten, daher wird die Betrachtung der Hardware weitestgehend außer Acht gelassen.

1.3 Struktur der Arbeit

Um die Relevanz dieser Arbeit zu verdeutlichen, wird der aktuellen Hintergrund der Entwicklung von Sprachassistenten mit einbezogen und zuerst auf die Geschichte von Sprachassistenten eingegangen, deren Bedeutung am Markt und deren Fähigkeit zur Steuerung von Musik. Daraufhin werden Einsatzszenarien der Sprachsuche diskutiert, aus denen die Anforderungen für die Software extrahiert werden. Hierbei werden Schwierigkeiten der Sprachsoftware in Bezug auf die verschiedenen Anwendungsszenarien genannt sowie mögliche Lösungen in Form von Soft- und Hardware vorgestellt. In Folge dessen werden die theoretischen Grundlagen der Sprachsuche behandelt. Diese unterteilen sich in die Voraussetzungen an die Sprachsteuerung, die

Theorie der Microsoft Speech Platform sowie in die Strukturierung der Daten. Hierbei werden besonders die Gestaltung der Grammatik sowie die Nutzung der *Semantik* erläutert, da diese für das Ergebnis dieser Arbeit eine hohe Bedeutung besitzen. In dem theoretischen Teil dieser Arbeit wird die Theorie der Microsoft Speech Platform im Detail erläutert und ihre Funktionsweise erklärt. Hierbei wird auch ein Vergleich der Microsoft Speech Platform und SAPI durchgeführt und deren Unterschiede erläutert. Dazu wird näher auf die Theorie von Grammatiken der Speech Recognition Grammar Specification (SRGS) eingegangen, welche von beiden Sprachsystemen benutzt wird sowie die Verwendung von Semantik beispielhaft erklärt. Hierzu werden auch potenzielle Schwierigkeiten bei der Übersetzung von gesprochener Sprache in Text mit der Microsoft Speech Platform behandelt sowie deren Auswirkungen auf die Bildung von Grammatik. Hierzu wird die Thematik der Betonung von Wörtern in Kurzform behandelt.

Nach der Theorie folgt die praktische Durchführung der Arbeit. Es wird methodisch dargestellt, wie eine solche Sprachsuche für eine Sprachbibliothek mit klassischer Musik gestaltet werden kann, wie die Grammatiken für die Sprachsuche gebildet werden und wie das System für die genannten Einsatzzwecke optimiert werden kann. Außerdem werden die theoretisch gezeigten Datenstrukturen praktisch anhand von Beispieldatensätzen, die konkrete Problemfälle darstellen, behandelt.

Diese Arbeit richtet sich an Personen mit technischem Hintergrund und mit Erfahrung im Programmieren. Die Nutzung der Microsoft Speech Platform erfolgt in einer .NET Programmiersprache. Da diese Arbeit sich nicht auf eine Programmiersprache beschränkt, wird auf die Darstellungen von Code verzichtet. Somit wird die Anwendung der Microsoft Speech Platform für .NET allgemein erklärt. Dazu gehören Fachbegriffe aus der Welt des Programmierens und des Datenbankwesens. Dazu ist ein allgemein technisches Verständnis nützlich, da die Thematik einer Sprachsteuerung viele technische Aspekte mit einbezieht. Im Laufe dieser Arbeit werden Begriffe wie „Nutzer“, „Anwender“, oder „Entwickler“ verwendet. Mit diesen Begriffen werden männliche, sowie weibliche Personen angesprochen. Auf die gesonderte Darstellung der weiblichen Form dieser Begriffe wird zugunsten des Leseflusses verzichtet.

2 Sprachassistenten

Sprachassistenten sind eine moderne Weiterentwicklung von Spracherkennungssystemen. Dass diese mittlerweile ausreichend funktionieren, um auf dem Massenmarkt Anklang zu finden, zeigt sich durch die starke Verbreitung von Google Now, Siri und Cortana, welche auf einem großen Teil von Smartphones bereits vorinstalliert sind. In diesem Kapitel wird die Geschichte von modernen Sprachassistenten aufgezeigt sowie deren Bedeutung im Alltag. Hiernach wird kurz auf bekannte Fähigkeiten von Sprachassistenten zur Steuerung von Musik eingegangen.

2.1 Entstehung des Sprachassistenten Siri

Apple stellte 2011 seinen Sprachassistenten Siri vor. Siri gehört mit Google Now und Cortana zu den drei größten Sprachassistenten auf dem Markt und wurde 2011 als erstes dieser drei der Öffentlichkeit präsentiert.[7] Siri selbst hatte zu diesem Zeitpunkt bereits viele Jahre Entwicklung und Vorversionen hinter sich. Ursprünglich wurde das gemeinnützige Institut SRI International vom US Militär beauftragt, einen virtuellen Assistenten zu entwickeln. Dieser Assistent sollte Führungskräfte bei der Organisation unterstützend zuarbeiten, beispielsweise Meetings planen, wichtige Dokumente bereitstellen und Räume organisieren. Dazu sollte der Assistent lernfähig sein und in der Lage sein selbstständig Entscheidungen zu treffen. Dies zeigte sich in der Praxis durch das Analysieren vom Verhalten der Führungskräfte, was dazu führte, diesen automatisch bei Meetings Dokumente zukommen zu lassen, die für das Meeting relevant sind. Dieser Assistent, für den die künstliche Intelligenz HAL (Heuristischer Algorithmus) aus dem Film Odysee im Weltraum aus dem Jahre 2001 ein Vorbild war, wurde beim Militär unter dem Namen CALO (Cognitive Assistant that learns and organizes) geführt. Kann beispielsweise ein Teilnehmer kurzfristig nicht am Meeting teilnehmen, ist CALO in der Lage selbstständig zu entscheiden, ob das Meeting abgesagt und neu organisiert werden muss oder es stattfinden kann. Während des Meeting hört CALO mit und sendet allen Teilnehmern eine Mitschrift des Meetings per Email zu. Die Entwickler um CALO setzten sich zusammen und überlegten sich mit

[7] Vgl. (Brandt, Nutzung von Sprachassistenten in Deutschland, 2016)

einem eigenen Sprachassistenten selbstständig zu machen. Das Ergebnis dieses Treffens war Siri, die 2010 von dem Startup vorgestellt wurde. Ursprünglich war Siri eine eigenständige App und nicht als Suchmaschine vorgesehen, sondern als sogenannte „Do Engine“, als nächstes Level des Zugriffs auf das Internet. Siri sollte eine neue Schnittstelle für den Nutzer sein, er gibt ein was er benötigte und Siri sollte mithilfe dieser Informationen Aktionen ausführen. Hierfür nutzte Siri damals schon 42 verschiedene Webservice, wie Google Maps, Yelp und Wolfram Alpha. „Do Engine“ bedeutete für die Entwickler, dass Siri auch weit über das reine Anfragen hinausgeht. Beispielsweise sollte sie auch in der Lage sein, nach einem Abend mit Alkoholkonsum des Nutzers eigenständig ein Taxi zu rufen. Siri sollte die nächste Generation des Webzugriffs darstellen. Verizon, ein großer Mobilfunkanbieter in den USA, hatte Interesse Siri als App zu erwerben und sie auf allen Android Smartphones vorzuinstallieren. Apple entschied dann selbst, Siri zu kaufen und sie exklusiv auf Apple Telefonen anzubieten und ging als erfolgreicher Bieter aus dem Wettbewerb. Daraufhin entwickelte Google seinen eigenen Sprachassistenten Google Now. Microsoft, das mit dem Kauf von der Mobilfunksparte von Nokia 2013 in den Smartphone Markt einstieg, präsentierte Cortana im Jahr 2014.[8] Somit begann mit Siri der Durchbruch von Sprachassistenten.[9]

2.2 Stand von Sprachassistenten auf dem Markt

Sprachassistenten sind aktuell ein großes Thema, bei dem sich Branchenriesen, wie Apple, Google und Microsoft oder auch Amazon in starker Konkurrenz befinden. Während Apple mit Siri im Jahr 2011 und daraufhin Google mit Google Now im Jahr 2012 den Grundstein für Sprachassistenten legten, stellte Microsoft 2014 die Sprachassistentin Cortana vor. Diese Sprachassistenten zeichnen sich dadurch aus, dass der Nutzer eine beliebige Frage in den Assistenten einsprechen kann und dieser dann je nach Anfrage Informationen liefert oder eine Tätigkeit ausführt, wie das Stellen eines Weckers. Der Sprachassistent wird in der Regel mit einem Druck auf den Homebutton oder ähnlicher Aktionen gestartet. Dazu unterstützen die drei Sprachassistenten alle

[8] Vgl. (Kerkmann, 2014)
[9] Vgl. (Bosker, 2013)

eine Aktivierung per Sprache. Bei Google Now ist dies „Ok Google“[10], bei Siri „Hey Siri“[11] und bei Cortana „Hey Cortana“[12]. Diese Befehle sind in begrenztem Umfang konfigurierbar. Daraufhin kann der Nutzer den gewünschten Befehl einsprechen. Während der Befehl zum Starten des Assistenten beispielsweise bei Siri nur lokal abläuft, wird die Spracherkennung selbst auf den Servern von Apple durchgeführt.[13] Cortana bildet hierbei eine kleine Ausnahme. Ohne Internetverbindung ist es zumindest möglich, den eigenen Pc per Sprache zu durchsuchen.[14]

Allein auf Google Now greifen in Deutschland etwa 17 Millionen Nutzer zu, während dies bei Siri bereits 10.8 Millionen sind. Dahinter folgt Microsofts Cortana mit 6.8 Millionen Nutzern. Amazons Alexa wird der vollständigkeitshalber auch erwähnt, ist mit 0.8 Millionen aber noch ein sehr kleiner Nischenanbieter in dieser Kategorie. Allerdings haben insgesamt erst 47% der Nutzer im deutschen Raum einen Sprachassistenten verwendet. Dies macht deutlich, dass die prozentuale Anzahl der Nutzer von Sprachassistenten zwar noch ausbaufähig ist, sie jedoch in absoluten Zahlen betrachtet für viele Millionen Menschen bereits ein regelmäßig verwendetes Tool darstellen. Mittlerweile wird von Verwendern von Sprachassistenten häufiger ein Anruf per Sprachassistent getätigt als manuell, jedoch wird nur in etwa von 33% der Nutzer die Sprachsteuerung für Entertainmentinhalte, wie z.B. dem Abspielen von Musik verwendet. Ein möglicher Grund liegt darin, dass es dem Anwender schlicht zu peinlich ist, den Sprachassistenten in der Öffentlichkeit zu nutzen. Dies ergab eine Studie von Creative Strategies im Juni 2016. Da viele Nutzer Musik beispielsweise während der Fahrt zur Arbeit oder im Büro hören, ist es ersichtlich, dass die Sprachsteuerung hier nicht genutzt wird. Die Ausnahme bildet hier die Fahrt im Auto, wo 51% der Teilnehmer angaben, eine Sprachsteuerung zu nutzen. Vergleichsweise sind es nur 1,3% bzw. 6% die angaben, die Steuerung auf der Arbeit bzw. in der Öffentlichkeit zu nutzen. Es ist zu erwähnen, dass diese Studie sich nur auf Google Now und Siri bezieht, die wie vorher ausgeführt über die meisten Nutzerzahlen verfügen.

[10] Vgl. (Google.1, kein Datum)
[11] Vgl. (Apple.1, kein Datum)
[12] Vgl. (Microsoft, 2016)
[13] Vgl. (Becker, 2016)
[14] Vgl. (Peters, 2015)

Somit zeigen diese Zahlen, dass das Interesse bei einem Großteil der Nutzer vorhanden ist und somit auch der Wunsch bei Nutzern und Anwendern existiert, diese Technologien zumindest aktuell zum privaten Gebrauch weiter zu entwickeln.[15][16][17]

Sprachassistenten können mittlerweile verschiedenste Aufgaben übernehmen. Sie helfen dem Nutzer bei der Navigation oder können ihm zeigen, wo sie den nächsten Supermarkt finden. Sie liefern ihm Informationen über aktuelle sportliche Events oder stellen den Wecker für den nächsten Tag. Außerdem kann z.B. Google Now dem Nutzer auch mitteilen, wann das bestelle Paket geliefert wird oder Daten zum nächsten Flug des Nutzers anzeigen. Damit dies möglich wird, durchsucht der Dienst von Google Now jegliche Emails des Nutzers, um diesen Informationen abzugewinnen. Dies wird von Datenschutzexperten als kritisch angesehen.[18] Google Now ist stark mit der Google Suche verknüpft. Die Suchmaschine selbst entscheidet je nach Eingabe des Nutzers, ob dies eine Suchaufforderung oder eine Anfrage an Google Now ist, diese Praxis ist bei den drei genannten Sprachassistenten eine gängige Vorgehensweise, um immer eine Antwort, sei sie direkt vom System oder durch eine Suche im Internet, zu liefern. Dazu bieten die Sprachassistenten einen Bildschirm in unterschiedlichen Formen, wo aktuelle Informationen, die für den Anwender wahrscheinlich relevant sind, angezeigt werden. Diese Informationen werden z.T. durch automatisierte Vorgänge, aber auch durch manuelle Eingabe des Nutzers erzeugt. Zum Beispiel analysieren die Sprachassistenten die Wege, welcher der Nutzer gegangen ist, um Bewegungsprofile zu erstellen. So kann der Nutzer sehen, wie lange er morgens zur Arbeit benötigt, da die Software herausfindet, wo der Nutzer wohnt bzw. arbeitet. Manuelle Eingaben können im Kalender getätigt werden und werden daraufhin im Sprachassistenten angezeigt.[19]

[15] Vgl. (Brandt, Anwendungsbereiche von digitalen Sprachassistenten, 2016)
[16] Vgl. (Brandt, Nutzung von Sprachassistenten in Deutschland, 2016)
[17] Vgl. (Milanesi, 2016)
[18] Vgl. (Klausing, 2015)
[19] Vgl. (Google.2, kein Datum)

2.3 Fähigkeiten der Sprachassistenten zur Suche von Musik

Sprachassistenten werden immer weiter entwickelt. Im Internet existieren einige Listen mit Befehlen für Sprachassistenten. Diesen mangelt es jedoch an objektiver Bestätigung ihrer Korrektheit. Nutzer werden von Herstellern gebeten, selbst auszuprobieren, was der Sprachassistent leisten kann.[20] Davon kann das Ziel der Betreiber der am meisten verwendeten Sprachassistenten abgeleitet werden. Durch das Ausprobieren von Befehlen ist der Nutzer aufgrund fehlender Erwartungshaltung weniger enttäuscht, sollte ein Befehl nicht funktionieren. Aber er ist positiv überrascht, wenn er funktionsfähig ist. Auch unterscheiden sich die Fähigkeiten der Sprachassistenten je nach Sprachraum, da Befehle für jede Sprache einzeln entwickelt werden müssen. Um diese Menge an Informationen auf objektiver Ebene zu betrachten, werden an dieser Stelle nur Befehle erwähnt, die bei offiziellen Quellen in deutscher Sprache gefunden wurden. Manche Sprachassistenten, wie Siri, bieten die Möglichkeit gezielt Titel, Alben oder ein Genre abzuspielen.[21] Hierfür wird der Befehl „Spiele… ab“ verwendet. Bei Siri sind bereits erweiterte Funktionen zur Steuerung von Musik vorhanden. So kann bei einem Aktivieren von Siri durch den Homebutton die Musik pausiert oder der nächste bzw. vorherige Titel abgespielt werden.[22] Während des Sprechvorgangs wird die Wiedergabe pausiert. Microsoft bietet für Cortana leider keine ausreichende Befehlsliste an. Google Now bietet nur die Wiedergabe eines Musikstückes mit Angabe des Titels an[23].

Alle drei Anbieter sind in der Lage, Musik durch Zuhören zu erkennen. Cortana und Siri greifen für diesen Zweck auf die Datenbanken von Shazam zurück[24][25]. Google Now geht eine zweiseitige Strategie und bietet einen eigenen Suchdienst für Musik, nutzt aber auch die Dienste von Shazam.[26][27] Voraussetzung hierfür ist jedoch, dass die Shazam App installiert ist, was bei Siri und Cortana nicht notwendig ist. Dies ist

[20] Vgl. (Google.3, kein Datum)
[21] Vgl. (Apple.1, kein Datum)
[22] Vgl. ebd.
[23] Vgl. (Google.4, kein Datum)
[24] Vgl. (Apple.1, kein Datum)
[25] Vgl. (windowsunited, 2015)
[26] Vgl. (Google.5, kein Datum)
[27] Vgl. (Google.6, kein Datum)

ein Beispiel dafür, wie die Funktionalitäten von Apps, wie Shazam in die Sprachassistenten direkt eingebunden werden, was im schnellsten Fall ohne ein gesondertes Öffnen der App abläuft. Beispielsweise analysiert Google Now nach Aktivierung der Sprachaufnahme die Umgebungsgeräusche. Hört es dabei potenzielle Musik, bietet Google Now mit einem weiteren Klick an, die Musik zu analysieren. Ist dies erfolgreich, werden der Titel des Liedes sowie das Album und der Interpret dem Nutzer angezeigt. Zudem wird zugleich die Möglichkeit angeboten, diesen bei dem eigenen Dienst Google Play zu erwerben.

3 Einsatzmöglichkeiten einer Sprachsteuerung für Musik

Es stellt sich die Frage, in welchen Situationen die Vorteile einer Sprachsteuerung gegenüber anderen Steuerungsmethoden überwiegen. Laut der Studie von Creative Strategies ist eine Sprachsteuerung für den durchschnittlichen Anwender in jenen Situationen interessant, wo er diese allein nutzen kann. In diesem Kapitel werden diese und weitere Einsatzmöglichkeiten sowie die zugehörigen Charakteristika erläutert.

3.1 Menschen mit besonderem Förderungsbedarf

In einigen Fällen kann eine Sprachsteuerung als eine Art Luxus angesehen werden. Der Anwender kann entscheiden, ob er diese oder eine manuelle Eingabe bevorzugt, wobei die Entscheidung je nach Technologie unterschiedlich ausfällt. Ist ein Anwender jedoch soweit körperlich eingeschränkt, dass eine Sprachsteuerung nicht mehr als nur eine weitere Eingabemethode anzusehen ist, sondern nahezu die einzige verfügbare Eingabemöglichkeit darstellt, nimmt ihre Bedeutung einen viel höheren Stellenwert ein. Ein Beispiel hierfür sind Menschen, die nicht mehr ihre Hände bewegen können oder im schlimmsten Fall sogar querschnittsgelähmt sind. Die übliche Steuerung durch eine Tastatur und Maus oder durch den Touchscreen am Smartphone ist nicht mehr möglich. Dies ist ein Einsatzzweck für Sprachsteuerungen, die dem Anwender nicht nur ermöglichen, alle Funktionen am Endgerät trotz körperlicher Einschränkungen in vollem Umfang zu nutzen, sondern dass diese auch reibungsfrei funktionieren, da hiervon die Fähigkeiten des Nutzers im Alltag und seine Einschränkungen im Leben abhängig sein können. Ebenso können Menschen mit einer Einschränkung in der Beweglichkeit von einer Sprachsteuerung profitieren, wie

beispielsweise alte oder gehbehinderte Menschen. So kann beispielsweise ein Computer oder eine Audioanlage mithilfe von Sprache gesteuert werden. Auf diese Weise wird vermieden, dass der Anwender einen für ihn erschwerten Fußweg bewältigen muss. Zu dieser Kategorie zählen ebenso Nutzer, die aufgrund von eingeschränkten motorischen Fähigkeiten nicht in der Lage sind, ein Smartphone oder ein anderes Gerät zu bedienen. Eine Sprachsteuerung ist für diese Nutzergruppe ein geeignetes Mittel, um trotzdem Musik abzuspielen.

3.2 Bedienen von Fahrzeugen

Die Bedienung von Funktionen während der Fahrt im Fahrzeug ist in der Automobilbranche bereits ein eigenes Feld. Dies zeigt der Sprachsteuerungstest des ADAC, in dem die Automobilhersteller selbst Lösungen anbieten.[28] Dies zeigt bereits, wie wichtig Sprachsteuerungen für diese Branche geworden sind.

Eine Herausforderung für die Hersteller von Sprachsteuerungssystemen für Automobile liegt darin, ein System zu kreieren, welches auch beim Bedienen des Fahrzeuges einfach und mit wenig Ablenkung funktioniert, sodass der Fahrer sich auf den Verkehr konzentrieren kann. Negativbeispiele sind hier die Bedienung des Radios oder der illegalen Nutzung des Smartphones am Steuer während der Fahrt. Dazu ist eine hohe Erkennungsgenauigkeit notwendig da während der Fahrt, besonders unter hohen Geschwindigkeiten, ein erhöhter Geräuschpegel auftritt, der die Erkennung erschwert. Diesen Störgeräuschen kann mit einem gerichteten Mikrofon, einer Dämmung des Fahrzeugs und softwareseitigen Geräuschfiltern vorgebeugt werden.

In einer Studie des Institutes für Wirtschaftsinformatik Braunschweig wurden Teilnehmer u.a. zu deren Meinung zu verschiedenen Eingabemöglichkeiten im Fahrzeug befragt. Die Sprachsteuerung selbst wurde negativ bewertet, schlechter als die Steuerung über Touchscreen oder Lenkradtasten. Die Forschungsgruppe deutete das Ergebnis so, dass die Nutzer mit der Qualität der aktuellen Sprachsteuerungsangebote nicht zufrieden sind, hier existiert also Verbesserungsbedarf.[29] Aus dieser Studie

[28] (ADAC, kein Datum)
[29] Vgl. (Wolf, Hess, & Benlian, 2012)

geht auch hervor, dass Nutzer den Wunsch verspüren nicht zu sehr durch weitere Dienste, wie Multimediaelemente abgelenkt werden wollen. Gerade in Bezug auf die Situation der Führung eines Fahrzeuges werden also auch an die Sprachsteuerung hohe Erwartungen gesetzt. Sonst wird diese schlicht nicht genutzt.

3.3 Verschiedene Anwendung im alltäglichen Leben

Eine Sprachsteuerung ist für jene Situationen im alltäglichen Leben interessant, in denen es dem Nutzer erschwert ist, übliche Eingabemethoden zu verwenden. Es können also unter anderem alle Tätigkeiten näher betrachtet werden, die die Nutzung der Hände einschränkt.

Beim Kochen ist so eine Situation gegeben. Der Anwender schneidet beispielsweise Fleisch und möchte es vermeiden, mit seinen Händen das Smartphone oder Endgerät zu berühren. Hierfür ist eine Sprachsteuerung ein geeignetes Einsatzmittel, um die Nutzung der Hände zu vermeiden. Zu beachten ist jedoch, dass gerade beim Kochen andere Geräuschquellen auftreten können. Eine Möglichkeit dieses Problem zu vermeiden ist die Verwendung eines Headsets mit eingebautem Mikrofon, welches einen Fokus auf nahe Geräuschquellen legt.

Sämtliche handwerklichen Arbeiten, in denen eine Verschmutzung der Hände stattfindet, bieten sich ebenfalls für die Verwendung einer Sprachsteuerung an. Sollte der Anwender unter der Dusche Musik hören wollen, wäre hier ebenfalls eine Anwendung denkbar, um einen Kontakt des Gerätes mit Wasser vorzubeugen. Gerade beim Duschen ersteht auch ein starker Geräuschpegel, der die Nutzung einer Sprachsteuerung erschwert. Die Nutzung eines Headsets ist aufgrund des Wasserkontaktes nur mit dafür ausgelegten Geräten möglich. Hier müssen also Optimierungen des Mikrofones und intelligente Geräuschfilter vorhanden sein, die das Geräusch des fließenden Wassers unterdrücken, sodass eine Erkennung überhaupt möglich wird.

Ebenso sind Situationen, in denen der Nutzer aufgrund besonderer Konzentration nicht abgelenkt werden möchte, für eine Sprachsteuerung denkbar. Sollte der Nutzer bei der konzentrierten Arbeit Musik zur Steigerung der Konzentration hören wollen, ist es denkbar, dass der Hauptfokus auf der Arbeit liegen soll und die Auswahl eines

Musikstückes oder einer Playlist schnell und unkompliziert funktionieren soll. Dies ist ein Kriterium, um die Sprachsteuerung in diesem Feld einsetzen zu wollen. Ob ein Nachfragen der Anwendung nach Korrektheit des Titels das konzentrierte Umfeld bereits beeinträchtigt, muss untersucht werden, bzw. vom Anwender selbst entschieden werden.

Eine andere Verwendungsmöglichkeit für eine Sprachsteuerung ist beispielsweise bei sportlicher Tätigkeit wie beim Joggen gegeben. Als passendes Beispiel für diese Arbeit sei der Ausdauerläufer erwähnt, der Musik beim Laufen hören möchte. Leichte Musikabspielgeräte, die aufgrund ihres Gewichtes bei Joggern beliebt sind, bieten nur beschränkte Eingabemöglichkeiten, die dem Gewicht und der Größe geschuldet sind. Besonders bei einer hohen Anzahl von Musiktiteln dauert es länger, den gewünschten Titel mit der eingeschränkten Steuerung des Gerätes abzuspielen. Dieses Problem kann eine Sprachsteuerung mit einer gezielten Suche lösen. Die Herausforderung liegt darin, eine funktionsfähige Sprachsteuerung auf Geräte mit geringer Größe und Leistungsfähigkeit zu portieren. Smartphones, welche hierzu in der Lage sind, sind deutlich schwerer und benötigen hierfür bei allen drei Sprachassistent Anbietern einen Internetzugang.

4 Theoretische Grundlagen der Sprachsuche

Aus dem vorherigen Kapitel sind nun einige Situationen genannt worden, aus denen sich die Kriterien für eine funktionsfähige Sprachsteuerung herleiten lässt. In diesem Kapitel wird analysiert, was realistische Voraussetzungen für eine Sprachsteuerung einer Musikbibliothek sind und was durch die Microsoft Speech Platform realisiert werden kann.

4.1 Voraussetzungen an die Sprachsteuerung

Aus den Szenarien, die im vorherigen Kapitel erörtert wurden, können folgende Voraussetzungen abgeleitet werden. Die Sprachsteuerung muss leicht aktivierbar sein, um die Verzögerung bis zur Befehlseingabe zu minimieren. Sollte der Nutzer gezwungen sein, erst eine App zu öffnen, in der er die Sprachsteuerung nutzen kann, ist der manuelle Weg, Musik händisch in der App zu steuern / zu kontrollieren, mindestens

gleichschnell. Dieser Vorteil einer Sprachsteuerung ergibt somit keinen Nutzen. Außerdem wäre die Nutzung der Hände ein weiteres Ausschlusskriterium für sämtliche Situationen, in denen die Hände des Nutzers anderweitig beschäftigt sind, bzw. aufgrund von körperlichen Einschränkungen oder Verschmutzung etc. nicht nutzbar sind. Eine mögliche Lösung für das Problem wäre das permanente Zuhören der Sprachsteuerung, wie es beispielsweise bei Google Now durch das Schlüsselwort „Ok Google“ geschieht. Unter Microsofts Cortana wäre dies mit „Hey Cortana“ möglich. Sollte der Nutzer jedoch Bedenken bzgl. seines Datenschutzes haben und aufgrund dessen das permanente Zuhören nicht aktivieren wollen, gibt es noch die Möglichkeit über den Annäherungssensor des Smartphones per Geste die Sprachsteuerung zu aktivieren. Hierbei werden die Hände zwar benötigt, aber das Smartphone bleibt frei von direkter Berührung und somit von Verschmutzung. Die Unterstützung von solchen Gesten durch den Annäherungssensor im Smartphone und eine Verknüpfung dieser mit der Sprachsteuerung, ist für diese Vorgehensweise notwendig.

Eine andere Voraussetzung für die Sprachsteuerung ist die korrekte akustische Erkennung des Gesprochenen des Nutzers, möglichst auch bei lauten Nebengeräuschen. Mikrophone von Smartphones sind für das Telefonieren optimiert. Dies bedeutet, dass diese Mikrophone zumeist nicht gerichtet, sondern kugelförmig aufnehmen. Dies führt zu einer vermehrten Mitaufnahme von Störgeräuschen, die im Nachhinein softwaretechnisch, soweit möglich, entfernt werden müssen. Dazu werden das eingebaute Mikrophon und dessen Fähigkeiten oft nicht deklariert, was es dem Nutzer enorm erschwert, ein Smartphone für den Zweck der Sprachsteuerung zu erwerben. Selbst ein gutes Mikrophon ist keine Garantie für ein gutes Verständnis des Sprachassistenten. Amazon hat mit seinem Homespeaker einen Versuch unternommen, seinen Sprachassistenten Alexa in die Wohnzimmer der Nutzer zu bringen. Obwohl das Gerät 7 Mikrophone besitzt, ist es dennoch nicht mehr in der Lage den Nutzer zu verstehen, sobald es Musik abspielt. Ansätze wie die „non-negative matrix factorization“ bieten eine Lösung für dieses Problem, jedoch lässt sich aufgrund des Betriebsgeheimnisses der jeweiligen Anbieter nur vermuten, ob dieser Ansatz auch

in der Praxis Verwendung findet. Daher ist dies ein Kriterium, dessen Lösung die Grenzen dieser Arbeit überschreiten und deren technische Tiefe den Rahmen sprengen würde.[30][31][32]

Die Sprachsteuerung sollte ebenfalls ein Feedback für den Nutzer unterstützen, sollte die Erkennung kein genaues Ergebnis geliefert haben. Hierfür muss eine Grenze gezogen werden, ab wann das Feedback eintreten soll oder ob dies bei jeder Anfrage an den Sprachassistenten geschehen soll. Mithilfe eines Feedbacks soll das Ziel erreicht werden, dass die Sprachsteuerung weniger Fehler produziert und somit den Vorgang beschleunigt. Sollte das Programm nun bei jeder Anfrage eine Rückfrage stellen, selbst wenn der Nutzer den Song nur pausieren wollte, kann dies zur Unzufriedenheit des Nutzers und somit zur Unzufriedenheit mit der Sprachsteuerung führen. Es wird also ein objektiver Wert benötigt, der die Wahrscheinlichkeit einer korrekten Erkennung darstellt. Dazu ist es notwendig, dass die Anwendung mit dem Nutzer kommunizieren kann, sodass dieser seine Suche bestätigen oder korrigieren kann.

4.2 Theorie der Spracherkennung mit der Microsoft Speech Platform

Microsoft hat im Laufe der Zeit seine Spracherkennungslösungen immer weiter überarbeitet. Während das Spracherkennungsmodul von Cortana aufgrund des Betastatus noch in Veränderung ist, bietet Microsoft jedoch die gut dokumentierten Sprachbibliotheken SAPI und Microsoft Speech Platform für die Verwendung von Spracherkennung an. In diesem Kapitel werden diese beiden Bibliotheken kurz verglichen, daraufhin wird die Microsoft Speech Platform genauer erläutert und deren Fähigkeiten vorgestellt.

4.2.1 Vergleich SAPI und Microsoft Speech Platform

Beide Sprachbibliotheken, SAPI und die Microsoft Speech Platform bieten Möglichkeiten für Spracherkennung an. Die Bestandteile sind in vielen Punkten identisch,

[30] Vgl. (billiger-telefonieren.de, 2013)
[31] Vgl. (Pierce, 2015)
[32] Vgl. (Raj, Virtanen, Chaudhuri, & Singh)

dennoch gibt es einige Unterschiede die nun näher analysiert werden. Die SAPI Bibliothek wird bereits seit Windows Vista mit dem Betriebssystem mitgeliefert und entspricht der integrierten Sprachsteuerung vom Betriebssystem. Microsoft bietet hier an, diese vorhandene Bibliothek in eigene Programme einzubinden und somit nutzbar zu machen. Dies bedeutet aber auch, dass die Sprachbibliothek im System fest eingebunden ist und nicht mit Programmen mitgepackt werden kann. Die Microsoft Speech Platform hingegen ist als externe DLL gekennzeichnet und muss manuell installiert werden, kann deshalb jedoch paketiert und somit Programmen angehangen werden. Während dies für Windows Computer keinen sonderlichen Unterschied bedeutet, ist dies für andere Plattformen, wo eine Sprachbibliothek nicht vorinstalliert ist, wie beispielsweise bei Linux mit installierter .NET Umgebung oder Windows Mobile absolut notwendig, da das Programm sonst nicht auf die Sprachbibliothek zugreifen kann. Hier ist die Verwendung von der Microsoft Speech Platform also alternativlos.[33]

Inhaltliche Unterschiede sind ebenfalls vorhanden. Beide Sprachbibliotheken verwenden Grammatiken, um die Eingabe des Nutzers zu beschränken. Auf das Thema Grammatiken wird in dem Kapitel 4.3 eingegangen. Hier ist wichtig zu wissen, dass diese Grammatik für die Verwendung der Microsoft Speech Platform definiert werden muss. Ohne definierte Grammatik ist diese nicht funktionsfähig. Die SAPI Sprachbibliothek funktioniert auch ohne feste Grammatik im freien Diktat. Ein weiterer Unterschied ist die Fähigkeit zum Training der Spracherkennung. Die Microsoft Speech Platform Bibliothek ist so gestaltet, dass ein Nutzertraining nicht nötig ist, was besonders für den Mehrnutzerbetrieb geeignet ist. Die SAPI Bibliothek bietet ein Training an und ist darauf ausgelegt, für einen Nutzer zu funktionieren.[34]

Außerdem ist mithilfe der Microsoft Speech Platform möglich, die Spracherkennung auf einen externen Server auszulagern. Somit kann die Spracherkennung für mehrere

[33]Vgl. (Microsoft.1, kein Datum)
[34] Vgl. (Microsoft.1, kein Datum)

Teilnehmer auf einen Server ausgelagert werden, was in großem Umfang bei den Sprachassistenten Siri, Google Now und Cortana eine gängige Vorgehensweise ist.[35]

Für diese Arbeit soll die Microsoft Speech Platform Sprachbibliothek genutzt werden, um möglichst nahe an die gängige Praxis der Sprachassistenten zu kommen. Diese Sprachbibliothek ist zwar zuletzt 2011 aktualisiert worden, dennoch bietet sie einen guten Einblick in die Funktionsweise von Cortana und ihre eventuellen zukünftigen Möglichkeiten und bietet alle benötigten Werkzeuge für eine Sprachsteuerung kostenlos und gut dokumentiert an. Die Verfügbarkeit und freie Verwendbarkeit macht sie zu einer geeigneten Testplattform, die Technologie der Spracherkennung zu erforschen.

4.2.2 Systemanforderungen

Um die Microsoft Speech Platform nutzen zu können, werden drei Bestandteile benötigt:

- Das Microsoft Speech Platform SDK[36]
- Die Microsoft Speech Platform Runtime[37]
- Eine Microsoft Speech Platform Sprache[38]

Microsoft bietet 26 verschiedene Sprachen an.[39] Außerdem bietet Microsoft für einige Sprachen auch verschiedene Akzente an, wie beispielsweise amerikanisches oder britisches Englisch. Es wird mindestens eine Sprache benötigt. Die Nutzung der Microsoft Speech Platform geschieht über eine Managed Code API über .NET oder über eine native Code API via einer COM Schnittstelle. Zweites bietet tiefer liegende Einstellungen an, die über .NET nicht angeboten werden, welche laut Microsoft jedoch für die Mehrheit an Sprachanwendungen ausreichend ist.[40] Für die Verwendung

[35] Vgl. Ebd.
[36] Vgl. (Microsoft, 2012)
[37] Vgl. (Microsoft, 2011)
[38] Vgl. (Microsoft, 2011)
[39] Vgl. (Microsoft.2, kein Datum)
[40] Vgl. (Microsoft.3, kein Datum)

in dieser Arbeit ist die native Code API ausreichend und wird daher auch in der praktischen Durchführung genutzt.

Die Microsoft Speech Platform kann auf .NET oder COM Technologie verwendet werden und unterstützt somit jegliche Programmiersprachen auf .NET Basis wie C# oder Visual Basic. Für C++ kann eine COM Schnittstelle der API genutzt werden.[41] Für alle drei Bestandteile werden mindestens Windows Vista oder Windows 2003 Server benötigt, 32-bit und 64-bit werden beide unterstützt. Dazu wird eine Entwicklungsumgebung wie Visual Studio benötigt und es muss mindestens die Version 4.0 von .NET installiert sein. Als Hardware wird mindestens eine 1 GHz CPU, 512 MB Ram, ein Ethernet Network Adapter und eine DirectX9 fähige Grafikkarte verlangt sowie USB 2.0. Die Relevanz der Voraussetzung von USB 2.0 kann in dieser Arbeit nicht überprüft werden.[42]

4.2.3 Technische Fähigkeiten und Funktionen der Microsoft Speech Platform

Die Microsoft Speech Platform bietet Werkzeuge um gesprochene Sprache in Text und Text mithilfe einer Sprachsynthese in gesprochene Sprache umzuwandeln. Die Plattform besteht aus drei Bestandteilen, dem Microsoft Speech Platform SDK, der Microsoft Speech Platform Runtime und mindestens einer Microsoft Speech Platform Sprache. Die Runtime und die Sprache sind die Hauptbestandteile der Microsoft Speech Platform. Programme, welche diese Plattform nutzen, sind ohne diese nicht lauffähig. Sollen mit der Plattform auch Programme entwickelt werden, wird das SDK ebenfalls benötigt. Microsoft bietet laut eigener Aussage die folgenden 26 Sprachen an, Akzente werden allerdings auch als eigene Sprache betrachtet: Catalanisch, Dänisch, Deutsch, Englisch (Australien), Englisch (Canada), Englisch (Großbritannien), Englisch (Indisch), Englisch (USA), Spanisch (Spanien), Spanisch (Mexiko), Finnisch, Französisch (Canada), Französisch (Frankreich), Italienisch, Japanisch, Koreanisch, Norwegisch, Niederländisch, Polnisch, Portugiesisch (Brasilien),

[41] Vgl. (Microsoft.4, kein Datum)
[42] Vgl. (Microsoft.5, kein Datum)

Portugiesisch (Portugal), Russisch, Schwedisch, Chinesisch (China), Chinesisch (Hongkong) und Chinesisch (Taiwan).[43]

Die Spracherkennung-Engine der Microsoft Speech Platform nutzt festgelegte Grammatiken, um die Spracherkennung zu ermöglichen. Sie bilden das Fundament für die Spracherkennung und setzen genaue Regeln fest. Diese Grammatiken werden in Kapitel 4.2.4 näher erläutert. Das bei der Spracherkennung aufgenommene Audio wird in numerische Segmente unterteilt, die voraussichtlich eine Stimme enthalten. Die numerischen Werte enthalten dabei die Eigenschaften der Stimme. Danach werden die aufgenommenen Daten anhand von drei Datenbanken verglichen. Diese drei Datenbanken unterteilen sich in eine Akustik-Modell Datenbank, in ein Lexikon und in eine Sprachmodell Datenbank. Die Akustik-Modell Datenbank enthält Daten zur Akustik von Sprache und Umgebungen. Dieser Teil kann auf die Stimme eines Nutzers oder einer Arbeitsumgebung trainiert werden. Die Lexikon-Datenbank enthält Wortlisten und deren Betonung, während in der Sprachmodell-Datenbank gespeichert ist, wie Wörter in einer Sprache verknüpft werden. Sollte nach dem Vergleich der Datenbanken eine Spracheingabe erkannt werden, wird ein Event ausgelöst, welches programmintern bearbeitet werden kann. Das Audio kann aus einem Mikrofon, einer .WAV Datei oder einem Datenstream entnommen werden. Dazu werden Werkzeuge angeboten, mit denen das Audio analysiert werden kann[44][45].

Außerdem ist ein Bezug auf die Praxis der Anwendung von hoher Bedeutung. Die Sprachsuche in einer Musikbibliothek soll in der Lage sein, Musik nach unterschiedlichen Kriterien zu finden. Es ist nicht Aufgabe dieser Sprachsuche, beispielsweise ein Taxi bestellen zu können oder online Musik zu suchen. Dies ist eine große Herausforderung für die modernen Sprachassistenten, denn einerseits vergrößert sich die Anzahl der Einsatzmöglichkeiten, andererseits wird die Erkennung aufgrund dessen

[43] Vgl. (Microsoft.6, kein Datum)
[44] Vgl. (Microsoft.7, kein Datum)
[45] Vgl. (Microsoft.8, kein Datum)

immer schwieriger. Sollte die Sprachsuche in der Musikbibliothek eine weitere Aufgabe bekommen, wird die Genauigkeit der Erkennung sinken.[46]

Die Microsoft Speech Platform bietet die Möglichkeit, die „Confidence“ einer verstandenen Sprachphrase zu analysieren und mit Alternativen zu vergleichen. Der Confidence Wert bewegt sich im Zahlenraum von Null bis 1 im Format einer Gleitkommazahl. Der Confidence Wert stellt nicht die absolute Wahrscheinlichkeit einer korrekten Spracherkennung dar, sondern ist als Vergleichswert zu verstehen, mit dem verschiedene Phrasen verglichen werden können. Eine Phrase mit einer Confidence von 0.8 sagt aus, dass eine Phrase am ehesten dem Gesprochenen entsprechen könnte als andere Phrasen mit einer niedrigeren Confidence. Der Wert der Confidence ist von der verwendeten Spracherkennung-Engine abhängig, was dazu führt, dass Confidence Werte aus unterschiedlichen Spracherkennung-Engines nicht vergleichbar sind. Mit dem Confidence Wert werden bei einer erfolgreichen Spracherkennung die verstandene Phrase und mögliche Alternativen mitgeliefert. So kann je nach Kontext vergleichen werden, ob der Confidence Wert mit dem kontextuellen Zusammenhang eine sinnvolle Anweisung enthält oder ob eine andere Phrase in dem aktuellen Zusammenhang mit einem niedrigeren Confidence Wert eine sinnvollere Anweisung enthält. Auch wenn nur eine Phrase verstanden wird, ist dieser eine Confidence angeheftet. Während der Erstellung dieser Arbeit hat sich gezeigt, dass Phrasen mit einer geringen Confidence z.T. erheblich nicht dem Gesprochenen entsprachen. Je nach Anwendung ist es sinnvoll, verschiedene Eingaben auf ihre Confidence zu testen, um mit dessen Hilfe Verbesserungen am System vorzunehmen, mittels der Verwendung eines besseren Aufnahmegerätes, Reduzierung von Umgebungsgeräuschen oder durch Anpassung der Grammatik.[47]

4.2.4 Grammatik

Da Anwendungen in der Regel nur eine begrenzte Anzahl an Befehlen brauchen, nutzt die Microsoft Speech Platform Grammatiken, um die Struktur der Befehle zu definieren. Diese lassen sich vom Nutzer selbst gestalten und somit genau für den

[46] Vgl. (Microsoft.9, kein Datum)
[47] Vgl. (Microsoft.10, kein Datum)

Anwendungszweck optimieren. Dies steigert nicht nur die Genauigkeit der Erkennung, sondern stellt auch sicher, dass das Verstandene eine Bedeutung für die Anwendung hat und somit Irrelevantes nicht bearbeitet werden muss.[48] Microsoft bezeichnet die Grammatik folgendermaßen:

"A speech recognition grammar consists of a structured list of rules that identify words or phrases that the speech recognition engine should attempt to identify in the spoken input."[49]

Diese strukturierte Liste von Regeln legt der Nutzer in seiner Anwendung selbst fest. Dies kann durch drei verschiedene Formate geschehen, welche aller auf der Speech Recognition Grammar Specification Version 1.0 des W3C aufbauen.[50] Microsoft bietet die Darstellung in einer VoiceXML oder programmintern durch Klassen der Namespaces von SRGSGrammar und Grammarbuilder an. Strukturell unterscheiden sich die drei Möglichkeiten nicht.[51] Abbildung 1 stellt den Aufbau eines Befehls mit einer Verzweigung in VoiceXML dar:

```
<?xml version="1.0"?>
- <grammar xml:lang="de-DE" xmlns="http://www.w3.org/2001/06/grammar" root="abspielen"
mode="voice" version="1.0">
  - <rule scope="public" id="abspielen">
    - <one-of>
        <item> Spiele </item>
        <item> Spiel </item>
      </one-of>
    - <one-of>
      - <item>
          <ruleref uri="#komponist"/>
        </item>
      - <item>
          <ruleref uri="#titel"/>
        </item>
      </one-of>
      ab.
    </rule>
  - <rule scope="public" id="komponist">
    - <one-of>
        <item> Bach </item>
        <item> Beethoven </item>
        <item> Liszt </item>
      </one-of>
    </rule>
  - <rule scope="public" id="titel">
    - <one-of>
        <item> 9. Sinfonie</item>
        <item> Goldberg Variationen </item>
        <item> Liebestraum </item>
      </one-of>
    </rule>
</grammar>
```

Abbildung 1 Beispiel einer Grammatik mit VoiceXML

[48] Vgl. (Microsoft.7, kein Datum)
[49] Vgl. (Microsoft.11, kein Datum)
[50] Vgl. (Microsoft.12, kein Datum)
[51] Vgl. (Microsoft.13, kein Datum)

Mit dem Knoten “grammar“ beginnt das Grammatikkonstrukt. Als Attribute werden dem Knoten das Sprachlayout, hier de-DE für Deutsch in Deutschland, und den benötigten XML Namespace, der über W3C bereitgestellt wird. Dazu wird das Wurzelelement “abspielen“ definiert, welcher die Einstiegsregel und das Fundament des Sprachkonstruktes darstellt und das optionale Attribut „mode“ mit dem Wert “voice“. Dies bedeutet, dass diese Grammatik für gesprochenen Text bestimmt ist. Eine Alternative für „voice“ ist „dtmf“ was „dual tone multi frequenzy“ entspricht und somit auch für Telefonanlagen verwendet werden kann.[52][53]

Im Weiteren wird auf eine tiefgehende Erläuterung von Attributen und Knotenpunkten aufgrund der technischen Tiefe verzichtet. Die Grammatik in Abbildung 1 zeigt die grammatikalische Konstruktion eines Befehls zum Abspielen von Musik. Der Befehl beginnt mit dem Wort “Spiele“ oder “Spiel“. Daraufhin kann der Nutzer wählen, ob er einen Komponisten, hier „Bach“, „Beethoven“ oder „Liszt“ wählt oder ob er ein Musikstück wählen möchte, hier „9. Sinfonie“, „Goldberg Variationen“ oder „Liebestraum“. Diese Listen sind beliebig erweiterbar und werden als eigene Regel behandelt. Die Gestaltung der Regeln erlaubt es auch, Elemente wiederholbar oder optional zu gestalten. So kann beispielsweise eine weitere Regel eingefügt werden, die die folgenden Befehle erlaubt:

- „Spiele etwas von Beethoven ab“
- „Spiele Komponist Beethoven ab“

Die Worte „etwas von“ enthalten für die Anwendung selbst keinen Mehrwert. Diese Wortwahl lässt zwar implizieren, dass ein Stück von einem Komponisten abgespielt werden soll, jedoch gibt es in dem Befehlssatz aus Abbildung 1 kein Stück, was den Namen Beethoven trägt. Somit ist eine Kongruenz ausgeschlossen. Es ist klar, dass ein Stück von dem Komponisten Beethoven abgespielt werden soll. Der Befehl „Spiele Komponist Beethoven ab“ spiegelt in diesem Zusammenhang den gleichen

[52] Vgl. (IBM, kein Datum)
[53] (Microsoft.13, kein Datum)

Sachverhalt wieder. Dies gilt solange, wie der Begriff „Beethoven“ in der Datensammlung eindeutig ist. Sollte es einen Titel geben, der ebenfalls den Namen „Beethoven“ trägt, ist diese Aussage nicht mehr eindeutig. Klarer wird das Problem an dem Beispiel, dass ein Musiker auch Stücke komponiert, wie Herr Jörg Widmann.[54] Sollte der Sprachanwendung keine weiteren Informationen mitgeteilt werden, ist die Phrase „Spiele etwas von Jörg Widmann“ mehrdeutig. Er kann die Rolle des Musikers oder die Rolle des Komponisten einnehmen. Um dieses Problem der Mehrdeutigkeit zu behandeln, bietet die Microsoft Speech Platform die Verwendung von alternativen Phrasen an. Somit können in der Anwendung zwei von der Wortwahl her gleiche Phrasen, jedoch mit unterschiedlichen semantischen Informationen, verarbeitet werden. Um diesen Fall zu behandeln, muss die Anwendung die erhaltene Phrase mit ihren Alternativen vergleichen. Sollten zwei Phrasen den gleichen Wortlaut haben, ist eine Rückfrage der Anwendung via Sprachsynthese sinnvoll, um dieses Problem der Mehrdeutigkeit vollends zu lösen.

Obwohl einige optionale Bestandteile keine Information tragen, ist es dennoch sinnvoll, diese einzubinden, um Umgangssprache oder Slang ebenfalls verstehen zu können. Der Gebrauch seiner gewohnten Sprache ist für den Nutzer angenehmer und verhindert ein Auswendiglernen der Befehle der Anwendung. So können auch Eingaben, wie „Spiele bitte was von Beethoven ab“ und „Spiele etwas von Beethoven ab“ verstanden werden. Dies kann je nach Wunsch des Entwicklers weiter geführt werden, solange es die Struktur und die semantischen Informationen nicht behindert. Je mehr Informationen der Sprachbefehl enthält, desto sinnvoller kann es je nach Sprachbefehl sein mit optionalen Bestandteilen zu arbeiten. Damit können möglichst viele sprachliche Einschübe berücksichtigen werden, deren Nichtberücksichtigung die Erkennung erschwert oder sogar unmöglich macht. Allein schon die Erweiterungen mit „bitte“ und „etwas von“ machen vier verschiedene Spracheingaben möglich. Wird das optionale Wort „Komponist“ auch noch eingefügt, sind es bereits sieben mögliche Kombinationen. Um dem Nutzer auch ein angenehmes Gefühl beim Verwenden von einer längeren Spracheingabe wie „Spiele Komponist Beethoven und

[54] Vgl. (Deutschlandradio Kultur, 2015)

Dirigent Celibedache und Orchester Wiener Philharmoniker ab“ zu gewährleisten ist es sinnvoll, vor die informationstragenden Begriffe diese genannten optionalen Bestandteile einzusetzen. Somit kann die genannte Spracheingabe auch wie folgt ausgesprochen werden: „Spiele etwas von dem Komponisten Beethoven mit Celibedache und dem Wiener Philharmoniker ab“. Durch die Begriffe „mit“, „und dem“ hört sich der Satz natürlicher an.

Es stellt sich nun die Frage, wie die Anwendung nun durch die unbekannte Anzahl optionaler Wörter die für die Ausführung des Befehls relevanten Schlüsselwörter erkennt, welchen eine semantische Bedeutung zugeordnet ist. Eine Möglichkeit ist die Durchsuchung der Spracheingabe nach Schlüsselworten, welche in einer Datenbank abgelegt sind. So wird beispielsweise jedes Wort in der Spracheingabe mit „Beethoven“ verglichen, um die Identität des Künstlers festzustellen. Diese Lösung ist jedoch mit einer großen Menge an Rechenleistung verbunden und müsste vom Entwickler manuell gestaltet werden. Die Microsoft Speech Platform bietet für diese Problematik die Nutzung von Semantik an, um somit sprachlichen Bestandteilen eine Bedeutung anzuheften. Auf Semantik wird in dem Kapitel 4.2.6 näher eingegangen.

Mit dieser Form der Grammatik lassen sich also Verzweigungen, Querverweise, Wiederholungen und optionale Bestandteile definieren. Dazu wird auch die Verwendung von Betonung und Semantik unterstützt.

4.2.5 Betonung

Um die gesprochene Sprache mit Text im System abgleichen zu können, wird hierfür eine Schnittstelle benötigt. Diese Schnittstelle wird in der Microsoft Speech Platform durch die Lexikon-Datenbank durchgeführt. Hier sind Wörter und deren Betonung abgespeichert. Die Betonung findet je nach Sprache und Akzent unterschiedlich statt, was es auch erforderlich macht, für die gewünschte Sprache die entsprechende Runtime zu installieren. Die Lexikon-Datenbank basiert auf der Pronunciation Lexicon Specification (PLS) Version 1.0 des W3C.[55]

[55] Vgl. (Bagshaw, Burnett, Carter, & Scahill, 2008)

Microsoft bietet dem Nutzer an, eigene Betonungen festzulegen. Um den Sinn dessen zu klären sei das Beispiel des französischen Komponisten Frederic Chopin erwähnt. Dieser Name wird nicht in Deutsch, sondern in Französisch betont. Um dennoch die französische Aussprache mit einer deutschen Runtime zu ermöglichen, kann ein Eintrag in der Lexikon-Datenbank angelegt werden. Die Aussprache eines Begriffes wird mithilfe eines phonetischen Alphabetes dargestellt. Microsoft überlässt hier dem Nutzer die Wahl zwischen drei verschiedenen phonetischen Alphabeten. Für die Sprachsynthese wird das International Phonetic Alphabet (IPA) genutzt. Es zeichnet sich durch die Zuweisung der Aussprache eines Lautes zu nur einem Zeichen aus. Dies macht dieses Alphabet maschinell nutzbar. Das Wort „Chopin“ wird jemand, der nicht mit der französischen Herkunft des Komponisten vertraut ist, beispielsweise mithilfe von IPA als „ç ʁ ɸ ʃ n„ aussprechen. Hierfür bezieht sich die Sprachsynthese und die Spracherkennung auf die Lexikon Datenbank, in der mithilfe des Sprachlayouts die Betonung des Wortes generiert wird. Verwendet man hierfür die Deutsche Runtime, geniert das Programm auch eine deutsche Betonung. Das PLS unterstützt die Verwendung von mehreren Sprachen in einer Grammatik, dies wurde jedoch nicht von der Microsoft Speech Platform aufgegriffen. Da je Spracherkennung-Engine bzw. für jeden Sprachsyntheziser nur ein Sprachlayout gewählt werden kann, können und müssen bei Bedarf gesonderte Betonungen selbst gestaltet werden. Soll also die französische Betonung des Namens „Chopin“ ebenso verstanden werden, kann „ʃo.pɑ̃ˈn“ manuell der Lexikon Datenbank zugefügt werden, was in etwa die französische Betonung von „Chopin“ darstellt. Damit die Betonung jedoch genutzt werden kann, müssen Informationen bzgl. der Betonung auch vorhanden sein. Eine automatisierte Erstellung der Betonung ist nicht möglich, da rein aus dem Text „Chopin“ nicht hervorgeht, wie dieser ausgesprochen wird. Aufgrund der fehlenden Daten bzgl. der Betonung und mangelnder Unterstützung seitens der Microsoft Speech Platform wird die Thematik der manuellen Betonung in dieser Arbeit nicht weiter aufgegriffen.[56][57]

[56] Vgl. (Microsoft.14, kein Datum)
[57] Vgl. (Microsoft.15, kein Datum)

4.2.6 Semantik

Wie bereits in Kapitel 4.2.4 angesprochen wurde, kann Textbestandteilen mithilfe von Semantik eine Bedeutung zugewiesen werden.[58] Dies ermöglicht eine schnelle automatisierte Weiterbearbeitung des Befehls, da somit eine weitere Interpretation erleichtert oder nicht benötigt wird. Beispielsweise kann mithilfe von Semantik bei der Nutzung einer SQL Datenbank die Variable, welche die semantische Information enthält, welcher Begriff gesucht werden soll, direkt mit übergeben werden. Ersichtlich wird dies an dem folgenden Beispiel, in dem „<suchbegriff>“ eine Variable darstellt: „Spiele <suchbegriff> ab“. Die einzige relevante Information für die Suche ist der Suchbegriff, die anderen Wörter der Phrase werden für die weitere Suche nicht benötigt, da sie in diesem Kontext keine weitere Informationen enthalten. Hieraus kann also ein SQL Statement erstellt werden: „Select datei from Musikstuecke where komponist = <suchbegriff> or dirigent = <suchbegriff> or titel = <suchbegriff> or solist = <suchbegriff> or orchester = <suchbegriff>“. Über dieses Statement werden mehrere Felder in der Datenbank mit dem Inhalt der Variablen „<suchbegriff>“ verglichen und ausgegeben. Somit ist eine weitere Interpretation überflüssig. Die entsprechenden Dateien werden über das SQL Statement zurückgegeben und können abgespielt werden.

Die Rolle eines Künstlers kann, je nach Wahl der Gestaltung der Grammatik, direkt mit dem Namen verknüpft werden. Somit werden dem Befehl „Spiele Chopin ab“ zwei semantische Informationen mitgeliefert, der Name des Künstlers und dessen Rolle als Komponist. Somit ist eine eindeutige Zuordnung gewährleistet. Wird diese aus besonderen Gründen nicht gewünscht, kann auch bewusst nicht zwischen den Feldern Komponist, Dirigent, Titel, Solist oder Orchester unterschieden werden. Der Nutzer erhält gegebenenfalls eine Liste, wo der Suchbegriff in unterschiedlichsten Bedeutungen vorkommt. Jedoch existiert an dieser Stelle das Problem der Doppeldeutigkeit. Ein Stück kann ebenfalls den Namen eines Komponisten oder Dirigenten enthalten. In Kapitel 4.2.4 wurden optionale Parameter angesprochen. Mit diesen kann eine Suche weiter spezifiziert werden. Gegeben sei das Beispiel: „Spiele Komponist Bach ab“ und „Spiele Titel Liebestraum ab“. Ohne eine semantische Angabe,

[58] Vgl. (Tichelen & Burke, 2007)

gibt es keine Entscheidung, ob es sich bei „Bach“ und „Liebestraum“ um einen Komponisten handelt oder um einen Titel. Eine simple Suche in allen Datenbankfeldern wird, wie bereits erwähnt, durchgeführt. Die optionalen Wörter, welche in diesem Beispiel die semantische Information bzgl. des Suchfeldes enthalten, „Komponist“ und „Titel“ ermöglichen eine Abänderung des oben genannten SQL Statements und können die Suche genauer eingrenzen, da nicht nur die semantische Bedeutung des Begriffes, sondern auch die semantische Bedeutung der Interpretation des Begriffes überliefert wird. Das SQL Statement verändert sich mit der Beispielphrase „Spiele <suchfeld> <suchbegriff> ab“ zu: „Select datei from Musikstuecke where <suchfeld> = <suchbegriff>“. Somit wird bei der Angabe der Rolle von Liszt sichergestellt, dass Liszt ein Komponist ist und nur Titel von diesem Komponisten gewünscht sind. Abbildung 2 zeigt ein VoiceXML Dokument mit semantischen Informationen.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <grammar xml:lang="de-DE" tag-format="semantics/1.0" xmlns="http://www.w3.org/2001/06/grammar" root="personennennen" mode="voice" version="1.0">
    <tag>var titelausgabe = new Array;var personenausgabe = new Array;</tag>
  - <rule scope="public" id="personennennen">
    - <item>
        <ruleref uri="#person"/>
      </item>
    - <item repeat="0-5">
        und
        <ruleref uri="#person"/>
      </item>
      <tag>out.titel=titelausgabe;out.personen=personenausgabe;</tag>
    </rule>
  - <rule scope="public" id="person">
      <tag>temp=""</tag>
    - <one-of>
      - <item repeat="0-1">
          Herr
          <tag>temp="Herr"</tag>
        </item>
      - <item repeat="0-1">
          Frau
          <tag>temp="Frau"</tag>
        </item>
      </one-of>
      <tag>titelausgabe.push(temp);</tag>
    - <one-of>
      - <item>
          Müller
          <tag>personenausgabe.push("Müller");</tag>
        </item>
      - <item>
          Schmidt
          <tag>personenausgabe.push("Schmidt");</tag>
        </item>
      - <item>
          Schneider
          <tag>personenausgabe.push("Schneider");</tag>
        </item>
      - <item>
          Fischer
          <tag>personenausgabe.push("Fischer");</tag>
        </item>
      </one-of>
    </rule>
</grammar>
```

Abbildung 2 Nutzung von semantischen Informationen in einem VoiceXMLDokument

Mithilfe des VoiceXML Dokumentes in Abbildung 2 kann der Nutzer die Namen von Personen nennen. Hierbei kann er mehrere Namen über ein „und“ nennen und optional die Anrede „Herr“ oder „Frau“ verwenden. Um semantische Informationen zu verwalten, wird das tag „<tag>“ verwendet. In dem tag „<tag>“ kann ECMAScript verwendet werden, um die semantischen Informationen zu strukturieren und

auszugeben.[59] Ausgegeben werden semantische Informationen über den Zugriff auf das Objekt „out“. Diesem Objekt können beliebige Felder hinzugefügt werden. In diesem Beispiel werden die Arrays „titelausgabe“ und „personenausgabe“ erzeugt und daraufhin den Feldern „personen“ und „titel“ zugewiesen. Der Entwickler kann nun in seiner Anwendung auf die semantische Information zugreifen, welche in zwei Objekte mit semantischen Informationen aufgeteilt ist. Diese Objekte entsprechen den Feldern „personen“ und „titel“. Um diese Objekte mit Daten zu füllen, kann im ECMAScript die Methode „.push()“ verwendet werden. Mit dieser Methode wird dem Array ein neues Feld hinzugefügt und die Übergabeparameter werden in diese neue Zelle eingefügt.[60] Somit wird in dieser Grammatik mit der Anweisung „<tag> personenausgabe.push(“Müller“);</tag>“ die Zeichenkette Müller am Ende des Arrays „personenausgabe“ eingefügt. In dem Array „titelausgabe“ werden die Anreden der jeweiligen Personen gespeichert. Um diese in der Sprachanwendung korrekt zuweisen zu können, ist der Index der Arrays ein geeigneter Verbindungspunkt. Somit kann in Array „titelausgabe“ das Wort „Herr“ und in dem Array „personenausgabe“ das Wort „Müller“ gespeichert werden. In diesem Beispiel jedoch ist die Verwendung einer Anrede optional gehalten. Dies führt zu folgendem Problem: Der Nutzer spricht die Phrase „Herr Müller und Fischer und Frau Schmidt“. Da der Index die Daten verknüpfen soll, aber für „Fischer“ keine Anrede verwendet wird, verschiebt sich der Index und die Sprachanwendung registriert für Index Nr.2 „Frau Fischer“. Um dieses Problem zu vermeiden, müssen Platzhalter verwendet werden, sodass der Index eindeutig die zwei Arrays verbindet. In diesem Beispiel wird ein leerer String als Platzhalter eingefügt. Dies löst das genannte Problem jedoch nur soweit, bis der Nutzer eine Phrase wie „Müller und Schmidt und Frau Fischer“ spricht. Für „Müller“ wird ein Platzhalter eingefügt. Da dieser Platzhalter in Form eines leeren Strings jedoch bereits vorhanden ist, wird er für „Schmidt“ kein zweites Mal eingefügt. Das Ergebnis ist die Verbindung von „Frau“ und „Schmidt“. Somit wird ein Platzhalter benötigt, der die Information über den Index direkt beinhaltet. Mit der Methode „.lenght“ kann die Größe eines Arrays abgerufen werden, dieser kann als Index verwendet werden. Somit ist der Platzhalter immer eindeutig. Dies muss jedoch in der Sprachanwendung

[59] Vgl. (Microsoft.16, kein Datum)
[60] Vgl. (Mozilla, 2016)

beachtet werden. Eine andere Möglichkeit ist die stärkere Verwendung von ECMAScript, um Objekte zu erzeugen, welche den Titel und den Namen der Person speichern. Diese können ebenfalls mit der „push()“ Methode an die Sprachanwendung gesendet werden. Dies sei vollständigkeitshalber erwähnt. Mit ECMAScript gibt es viele weitere Möglichkeiten, die Grammatik zu gestalten, dies übertrifft jedoch den Rahmen dieser Arbeit.

In einem VoiceXML Dokument kann die Semantik auf verschiedene Weisen zugewiesen und in der Sprachanwendung verwendet werden. Semantische Informationen können mit dem „<tag>“ Tag angehängt und verarbeitet werden. Hierbei ist die Nutzung von Variablen möglich um die gewünschte Zeichenkette als Semantische Information anzuhängen.[61]

4.2.7 Sprachsynthese

Neben der Spracherkennung bietet Microsoft auch eine Sprachsynthese an. Mit dieser kann Text mithilfe einer künstlichen Stimme ausgegeben werden. Bei dieser können Parameter wie Sprache, Geschlecht, Geschwindigkeit, Betonung, Stimmhöhe und Lautstärke geändert werden. Die Sprachsynthese beruht auf der Speech Synthesis Markup Language (SSML), welche von W3C entwickelt wurde. Die Verwendung einer Sprachsynthese ermöglicht es für das Ziel dieser Arbeit, Rückfragen verbal an den Nutzer zu stellen. Dies ist gerade in dem Fall nützlich, wenn der Nutzer keine genaueren Angaben zu dem gewünschten Stück macht und somit Doppeldeutigkeiten entstehen. Für die Sprachsynthese wird eine eigene Runtime Language zusätzlich zur Spracherkennung-Runtime Language benötigt. Wiedergegeben werden können normale Zeichenketten, Zeichenketten mit Betonungsangabe im IPA Format, sogenannte „Prompts“, die einen komplexen Aufbau von Satzteilen ermöglichen und Zeichenketten im SSML Format, welches ebenfalls komplexere Aufbauten ermöglicht, sowie als VoiceXML hinterlegt werden kann. Für die Sprachsynthese kann die

[61] Vgl. (Tichelen & Burke, 2007)

Ausgabe in einem Ausgabegerät, wie einem Lautsprecher oder einer .WAV Datei erfolgen.[62][63]

4.3 Strukturierung der Daten

Dieses Kapitel befasst sich mit der Frage, wie die Metadaten der Musiktitel strukturiert und normalisiert werden können, sodass die Daten für die Spracherkennung verwendet werden können. Daraufhin wird tiefer auf die sprachliche Gestaltung der Daten eingegangen und wie mithilfe dieser, die Daten in einer SQL Datenbank zur direkten Benutzung in einer Anwendung gespeichert werden können.

4.3.1 Gestaltung der Datenbank

Für eine Musikbibliothek werden an die Datenbank selbst keine besonderen Anforderungen gestellt. Die Nutzung einer Datenbank in SQL ist genauso möglich, wie die Nutzung eines Ordners mit Musikdateien. Hierbei muss die Musikbibliothek folgende Anforderungen erfüllen:

- Die Musikdateien müssen speicherbar und zugreifbar sein

Die Speicherbarkeit und Zugreifbarkeit stellt sicher, dass die Titel nach dem Auffinden auch abgespielt werden können. Eine Zuweisung der Metadaten zu den Musikstücken kann auf verschiedene Weise stattfinden. In einer SQL Datenbank gestaltet sich dies beispielsweise durch eine Struktur, wie sie in Abbildung 3 ersichtlich ist.

[62] Vgl. (Burnett, Walker, & Hunt, 2004)
[63] Vgl. (Microsoft.17, kein Datum)

ID	*Titel*	*Komponist*	*Solist*	*Dirigent*	*Orchester*	*Datei*
1	Liebestraum	Liszt	Jorge Bolet	-	-	…
2	9. Sinfonie, Sinfonie Nr.9	Beethoven	Camilla Tilling, Gerhild Romber…	Riccardo Chailly	Gewandhausorchester	…

Abbildung 3 Struktur einer Musikbibliothek in SQL

Eine andere Möglichkeit ist die Nutzung eines Ordners, der Musikdateien enthält. Hierbei müssen die Musikdateien jedoch so gewählt sein, dass ein Taggingformat, wie beispielsweise ID3, unterstützt wird. Mithilfe eines Taggingformates werden Metadaten an Musikstücke angehangen. Hier entfällt die Nutzung von SQL Statements. Folglich muss die Suche über die jeweilige Programmiersprache geschehen, was den Programmieraufwand erhöht. Dies erfüllt das zweite Kriterium.

- Es muss eine Zuweisung der Metadaten der Musikstücke zu den Musikdateien existieren

Eine weitere Anforderung ist, dass die Daten normalisiert oder ein Thesaurus zum Synonymabgleich vorhanden ist. Um die Problematik zu verdeutlichen, wird das Beispiel des Komponisten Bach verwendet. Johann Sebastian Bach war seinerzeit einer der berühmtesten Komponisten und ist auch heute für seine Werke bekannt. Johann Sebastian Bach hatte viele Kinder, von denen u. a. Wilhelm Friedemann Bach auch Komponist war. Spricht der Nutzer den Befehl „Spiele Bach ab“ in einer kleinen Musikbibliothek, wo nur Stücke von Johann Sebastian Bach vorhanden sind, ist keine Mehrdeutigkeit zu erwarten. Sollte jedoch mindestens ein Stück von seinem Vater vorhanden sein, werden Stücke von beiden Komponisten in die Wiedergabeliste gelegt. Um dieses Problem zu lösen müssen die Namen voll ausgeschrieben werden, um eine Unterscheidung zu ermöglichen. Eine hohe Qualität der Daten ist also wünschenswert, da davon der Erfolg der Spracherkennung abhängt. Zudem ist es an dieser Stelle sinnvoll, per Sprachsynthese den Nutzer um eine Spezifizierung des Komponisten zu bitten. Ein Beispiel hierfür ist die Phrase: „Für den Begriff Bach existieren 2 Komponisten in der Datenbank: Johann Sebastian Bach und Wilhelm

Friedemann Bach, welcher Komponist soll abgespielt werden?“. Ein Thesaurus in welchem Synonyme behandelt werden ist dann sinnvoll, sollte ein Musiker einen Künstlernamen verwenden. Zusätzlich soll auch der echte Name des Künstlers genutzt werden. Hierfür ist die Verwendung eines komplexeren Datenbanksystems, wie SQL empfehlenswert, mit dem die Funktion eines Thesaurus, wie beispielsweise mehrere Namen für eine Person, abgebildet werden kann. Dies zum Zwecke der Vollständigkeit genannt, in dieser Arbeit aber nicht weiter behandelt. Damit ist das dritte Kriterium erfüllt.

- Es wird ein System benötigt, welche die Verwaltung von Synonymen sicherstellt, dies kann auch im Sinne einer Normalisierung der Daten geschehen.

4.3.2 Sprachliche Gestaltung

In der klassischen Musik gibt es viele verschiedene Wege, wie Künstler ihre Werke sauber und konstant strukturiert haben, z.B. mithilfe einer Werksnummer (Op.). Andere Künstler waren jedoch selbst nicht konstant in ihrer Vorgehensweise und es kam vor, dass sie unregelmäßig die Benennung ihrer Werke änderten oder Werke nachträglich hinzugefügt wurden. Dies führt in der Strukturierung der Datenbank zu einigen Problemen, die die erfolgreiche Durchführung einer Sprachsuche erschweren.[64]

Ein Fall ist die Nutzung einer doppelten Bezeichnung, als Beispiel wird hier das Stück „Turandot Akt 3“ von Puccini erwähnt. Dieses Stück ist auch unter dem Namen „Nessun Dorma“ bekannt. Der Nutzer kennt eventuell nicht beide Namen und er möchte das Stück über einen Namen abspielen können. Dies kann mit einer SQL Datenbank über Beziehungen abgebildet werden, sodass ein Stück mehrere Namen enthalten kann. Dies muss bei der Bildung der Grammatik beachtet werden, wodurch die Spracherkennung mit beiden Namen funktioniert. Außerdem kann ein Stück auch mehrere Solisten enthalten. Dies kann ebenfalls über eine entsprechende Beziehung abgebildet werden.

[64] Vgl. (Zaslaw, 1997)

Ein anderer Fall ist die Nutzung des Namens von Solisten, Dirigenten oder Komponisten. Nicht immer ist die volle Angabe des Namens notwendig. Der Befehl „Spiele Komponist Beethoven ab“ ist bei einmaligen Vorkommen in den Daten eindeutig und die Nutzung des vollen Namens mit Vornamen und Jonkheer bringt keinen Mehrwert. An anderer Stelle kennt der Nutzer den vollen Namen des Künstlers nicht und lässt die Nutzung des Vornamens daher aus. Bei Doppeldeutigkeit kann das Programm eine Rückfrage zur Spezifizierung des Künstlers stellen. Um das Problem der optionalen Nutzung der Vornamen zu lösen, kann mithilfe einer Beziehung auf Datenbankebene der Name in Vorname und Nachname aufgeteilt werden. In der Spracherkennungsanwendung wird jeder Vorname als optionaler Parameter deklariert. Dies muss jeweils für die Attribute Dirigent, Komponist und Solist geschehen. Für das Orchester ist dies soweit nicht sinnvoll, da ein Orchestername in der Regel eindeutig ist und eine Kürzung des Namens für den Nutzer keinen Mehrwert bringt. Viele Orchesternamen tragen noch den Beinamen einer Stadt im Namen, beispielsweise „Londoner Sinfonie Orchester“. Die Daten können entsprechend angepasst werden, sodass Geoinformationen, wie Stadt oder Land genutzt werden können und der Befehl „Spiele etwas aus London ab“ möglich ist. Dies kann je nach Nutzerwunsch verwendet werden, wird in dieser Arbeit jedoch nicht weiter behandelt.

Es existieren Musikstücke, welche nicht alle Daten, die in dieser Arbeit besprochen werden, besitzen. Ein Beispiel ist das Stück Liebestraum von Frank Liszt. Dies ist ein Stück für einen Solopianisten, und besitzt somit kein Orchester oder Dirigenten. Die Sprachanwendung muss so gestaltet sein, dass diese fehlenden Daten die Erkennung des gesuchten Stückes nicht erschweren.

Außerdem muss geklärt werden, wie mit unterschiedlichen Sprachen umgegangen wird. Ein Titel sollte aufgrund seiner Wiedererkennbarkeit in seiner ursprünglichen Sprache gehalten werden, da eine Übersetzung manuell durchgeführt werden muss. Dazu muss der übersetzte Titel dem Nutzer auch bekannt sein, was die praktische Anwendbarkeit drastisch verschlechtert. Anders ist dies bei dem Orchestereintrag. Es ist durchaus möglich, dass ein Nutzer ein Stück von dem „Londoner Symphonieorchester“ oder dem „London Symphony Orchestra“ hören möchte. Dies ist eine

Entscheidung, die jeder Anwender je nach persönlichem Empfinden treffen muss. Jedoch muss hier bedacht werden, dass die gewählte Sprache eine Information ist, welche bei der Übersetzung verloren geht. Dies wird bei dem Beispiel des „National Philharmonic Orchestre“ deutlich. Die Wahl der englischen Sprache impliziert, dass es sich auch um ein Orchester aus England oder einem anderen englischsprachigen Staat handelt. Durch eine Übersetzung dieses Titels würde diese Information verloren gehen und es bleibt unklar, um welches nationales Philharmonie Orchester es sich handelt. Dies sollte bei der Entscheidung überdacht werden. Für anderssprachige Einträge kann im besten Fall eine Betonung manuell eingestellt werden, sodass die englische oder französische Spracheingabe verstanden werden kann. Dies bedarf jedoch für jeden anderssprachigen Eintrag eine Betonung, was einen enormen Aufwand darstellt, da hierfür keine automatisierten Werkzeuge verfügbar sind. Daher wird die Nutzung der Betonung in dieser Arbeit ausgelassen.

Abkürzungen sind ein weiteres Problem, welches behandelt werden muss. Sinfonie Nr.3 wird als Sinfonie Nummer 3 ausgesprochen. Dies wird nicht automatisch von der Anwendung behandelt und muss manuell eingepflegt werden. In dieser Arbeit geschieht dies durch die Normalisierung der Daten, indem „Nr.“ als „Nummer“ ausgeschrieben wird. Dazu sei erwähnt, dass viele Anwender „Sinfonie Nr.9“ als „neunte Sinfonie“ aussprechen können. Hierfür kann der Begriff „9. Sinfonie“ als weiterer Titel dem Musikstück zugeordnet werden, wie in Abbildung 3 ersichtlich ist. Dazu fällt auch die Werksnummer, die mit Op. abgekürzt wird. Damit die Sprachanwendung den Begriff „Opus“ erkennen kann, muss dieses ausgeschrieben werden. Die Anwendung dieser sprachlichen Fälle kann in Anhang 1 betrachtet werden.

5 Praktische Durchführung

In diesem Kapitel wird die vorher behandelte Theorie praktisch angewandt, sodass diese Arbeit ebenfalls ein Beispiel für die Gestaltung einer Sprachsuche in einer Musikbibliothek mit klassischer Musik anbietet. Der erste Schritt hierbei ist die Sammlung von Musik sowie deren Indexierung und das Einpflegen in eine Datenbank. Die Vorteile einer SQL Datenbank wurden in der Theorie bereits behandelt. Daher wird eine solche auch verwendet. Die Indexierung der Musik geschieht manuell, um

eine Normalisierung der Daten zu gewährleisten. Hiermit ist die Nutzung eines gesonderten Thesaurus bzgl. der Verwaltung anderer Schreibweisen oder Synonymen überflüssig. Daraufhin wird auf das grammatikalische Konstrukt der Sprachsuche in Bezug auf die gewählten Titel erstellt. Zusätzlich wird geklärt, wie eine automatisierte Generierung der Grammatik stattfinden kann.

5.1 Auswahl von Musiktitel

Um die theoretischen Grundlagen praktisch anzuwenden, werden 24 Musiktitel für die Datenbank verwendet. Die Titel wurden dabei so ausgewählt, um die Lösungen der theoretisch angesprochenen sprachlichen Probleme in einem praxisnahen Umfeld anwenden zu können. Eine volle Liste der Titel kann im Anhang 1 eingesehen werden. Die Stücke wurden so ausgewählt, dass folgende Kriterien erfüllt sind:

- Stücke mit mehreren Namen sind vorhanden (z.B. Sinfonie Nr.3, Eroica)
- Anderssprachige Orchester sind vorhanden (z.B. London Symphony Orchestra)
- Namen aus unterschiedlichen Sprachen sind vorhanden (z.B. Giacomo Puccini)
- Datensätze mit fehlenden Daten sind vorhanden (z.B. Liszt, Liebestraum. Wird ohne Orchester gespielt)
- Mehrere Stücke eines Komponisten sind vorhanden (z.B. Beethoven, Sinfonie 3 und 9)
- Sinfonien mit gleicher Nummer sind vorhanden, die sich durch andere Attribute unterscheiden (z.B. Sinfonie Nr.3 von Beethoven, Sinfonie Nr.3 von Mahler)
- Es ist mindestens ein Datensatz vorhanden, der sich nur eindeutig über mindestens drei Parameter bestimmen lässt (Hier: Sinfonie Nr.3, dirigiert von Bernstein mit dem Wiener Philharmonie Orchester, jeweils vom Komponisten Beethoven und Mahler)
- Es sind Datensätze vorhanden, die sich über ihre Op. identifizieren lassen (z.B. Nocturne Op9 Nr.1)
- Es sind mehrere Stücke von einem Solisten vorhanden (z.B. Jonas Kaufmann)

- Es sind mindesten zwei Stücke vorhanden, deren Komponisten den selben Nachnamen tragen (Hier: Johann Sebastian Bach, Wilhelm Friedemann Bach)
- Es ist ein Stück mit einem Wert in mehreren Feldern vorhanden (Widmann, Komponist & Widmann Solist)

Diese Kriterien sollen mögliche Szenarien im Alltagsbetrieb einer Sprachsuche in einer Musikdatenbank mit klassischer Musik darstellen .Um die praktische Relevanz der Kriterien zu belegen, wurden 24 Musikstücke gesammelt und deren Metadaten manuell normiert und aufbereitet. Diese Musikstücke stellen die besonderen Anforderungen und Probleme dar, die es in den nächsten Kapiteln praktisch zu behandeln gilt.

5.2 Füllen der Datenbank

Wendet man die theoretischen Erkenntnisse aus Kapitel 4.3 an, so ergibt sich ein Datenbankschema, welches für das Fortlaufen dieser Arbeit verwendet wird. Dieses Schema kann in Abbildung 4 betrachtet werden.

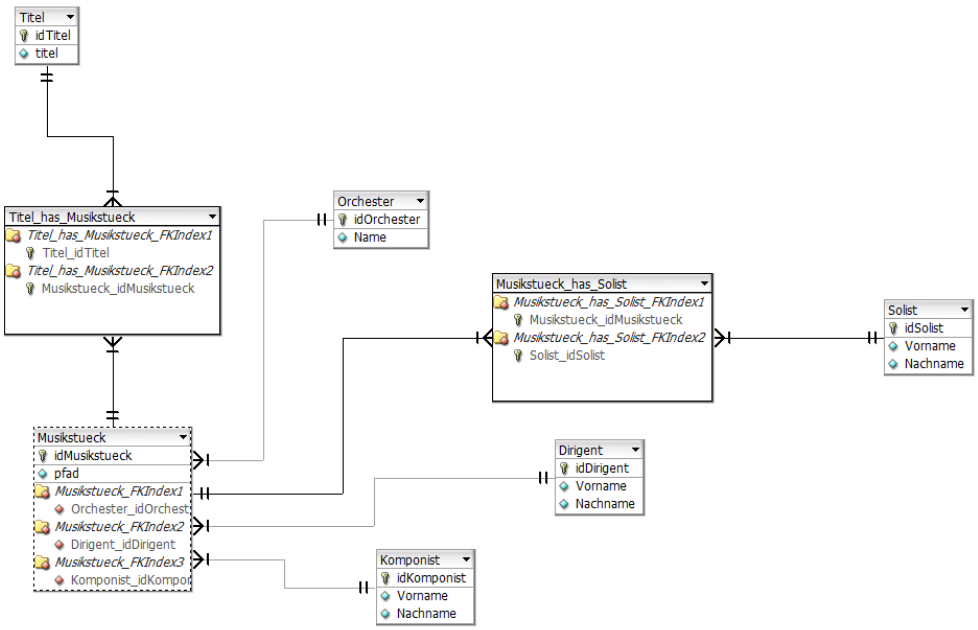

Abbildung 4 Schema einer für eine Sprachsuche optimierten Musikdatenbank

Abbildung 4 zeigt das Datenbankschema einer Musikbibliothek, die für die Spracherkennung optimiert wurde. Die Tabelle Musikstueck dient als Sammeltabelle, die alle Attribute enthält die eine Aufführung eines Musikstückes spezifizieren. Zu diesen Daten wird der Pfad zu der Datei gespeichert, die den Titel enthält. Alternativ kann die Datei z.B. in SQL auch als Datentyp „Blob“ direkt in der Datenbank gespeichert werden, die Nutzung des Pfades erleichtert jedoch für den Zweck dieser Arbeit den Zugriff auf die Datei. Aus der Tabelle „Titel“ geht hervor, dass ein Musikstück mehrere Titel besitzen kann und ein Titel zu mehreren Musikstücken gehören kann. Dies hat den Hintergrund, dass doppelte Einträge in Datenbanken Redundanzen entsprechen, die es zu vermeiden gilt, um die Konsistenz der Datenbank zu sichern. Da z.B. „Sinfonie Nr.9“ in den Beispieldaten zweimal vorhanden ist, bringt es für die Anwendung keinen Mehrwert, den Titel „Sinfonie Nr.9“ mehrfach zu speichern. Erkennt die Spracherkennung den Begriff „Sinfonie Nr.9“, werden automatisch alle entsprechenden Musikstücke ausgegeben. Eine N:M Beziehung wird auch zwischen der Tabelle „Musikstueck“ und der Tabelle „Solist“ verwendet. Ein Solist ist an mehreren Musikstücken beteiligt und ein Musikstück benötigt gegebenenfalls mehrere Solisten. In der Tabelle Solist wurde der Name in Vorname und Nachname gesplittet. Hierdurch kann während der Generierung der Grammatik, der Vorname als optionaler Parameter gekennzeichnet werden.

Die Aufteilung von Name in Vorname und Nachname geschieht ebenfalls bei den Tabellen „Dirigent“ und „Komponist“. Diese Tabellen werden mit der Tabelle Musikstueck mit einer 1:N Beziehung verknüpft, da ein Musikstück nur einen Komponisten und Dirigenten hat, unter einem Komponisten oder unter einem Dirigenten viele unterschiedliche Musikstücke zu finden sind. Dasselbe gilt auch für die Tabelle „Orchester“. Diese Beziehungsform kann sich je nach Datengrundlage ändern, ist für die genutzten Beispielsdaten in dieser Arbeit aber ausreichend.

Die Füllung der Datenbank erfolgt in dieser Arbeit manuell, über entsprechende Insert Statements in SQL. Für einen automatisierten Import der Daten in die Datenbank benötigt es einen Crawler, welcher automatisch neue Stücke der Bibliothek hinzufügen kann. Beispielsweise durchsucht der Crawler in regelmäßigen Abständen ein

Verzeichnis nach allen Musikdateien, entnimmt die Datensätze aus den Metadaten und fügt diese in die Datenbank ein. Damit dies funktioniert, müssen die Daten vorher bereits normalisiert sein, sodass die Datenbank keine redundanten Einträge wie „Ludwig van Beethoven“ und „L. Beethoven“ enthält, welche bei der Sprachsuche zu fehlerhaften Ergebnissen führen können. Je nach Anbieter der Musikdateien mit ihren Metadaten ist eine solche automatische Indexierung der Dateien möglich, für diese Arbeit werden aufgrund des benötigten Aufwandes die Daten manuell eingefügt.

Mithilfe dieser Datenstruktur können die Metadaten der Musiktitel abgespeichert und so strukturiert werden, dass im nächsten Schritt die Grammatik anhand dieser erzeugt werden kann.

5.3 Erstellung der Grammatik

Bevor eine Grammatik, in dieser Arbeit in VoiceXML erstellt wird, müssen im ersten Schritt alle Informationen gesammelt werden. Hierzu muss geklärt werden, wie weit optionale Bestandteile, wie zusätzliche Spezifizierungen, Freundlichkeitsfloskeln, spezifizierende Begriffe oder Verknüpfungsbegriffe in die Anwendung integriert werden sollen, um möglichst viele Befehle des Anwenders verständlich zu machen. Um das Konzept dieser optionalen Wörter vorzustellen, werden Beispielphrasen verwendet. Im Weiteren wird auf die allgemeine Struktur eines Sprachbefehls näher eingegangen und dessen sprachliche Gestaltung näher erläutert. Zum Schluss dieses Kapitels wird die Grammatik, bestehend aus den Beispieldaten in Anhang 1 mithilfe der sprachlichen Zusatzkomponenten, die in diesem Kapitel behandelt werden, gebildet. Es wird bewusst auf eine absolut korrekte grammatikalische Phrasenbildung verzichtet, um gewohnte Sprache des Nutzers ebenfalls zu berücksichtigen sowie minimalistisch gehaltene Befehle ebenfalls bearbeiten zu können. So kann der Befehl „Spiele Komponist Beethoven“ genau so verstanden werden wie der Befehl „Spiele etwas von Komponist Beethoven ab“. Es gibt in der Realität eine unzählbar große Anzahl von Möglichkeiten einen Befehl zum Abspielen von Musik zu bilden. Daher wurden für diese Arbeit einige Beispielphrasen verwendet, um die syntaktische Gestaltung dieser beispielshaft zu zeigen.

5.3.1 Sprachliche Zusätze

Als zusätzliche Spezifizierungen sind Wörter oder Phrasen gemeint, die für den Nutzer eine Information enthalten, für die Spracherkennung jedoch nicht relevant sind. Ein Beispiel hierfür ist die Phrase „etwas von“: „Spiele etwas vom Komponist Wiedeman ab“. Ist in der Datensammlung der Begriff „Widmann“ nur als Komponist vorhanden, ist diese Information bereits eindeutig, da Herr Widmann jedoch auch als Musiker tätig ist, wurde seine Rolle in dieser Phrase mit dem Begriff „Komponist“ spezifiziert. Diese Spezifizierung wird von der Anwendung dadurch abgefangen, dass durch die Verwendung des Begriffes „Dirigent“ die Alternative, in der Widmann als Komponist angesehen wird, nicht mit ausgegeben wird und somit entfällt. Das Wort „etwas vom“ findet hierbei keine zusätzliche Bedeutung, wird aber gegebenenfalls durch den Anwender aufgrund seiner gewohnten Sprachwahl verwendet. Dieser Satz ist so grammatikalisch nicht ganz korrekt. Die grammatikalische korrekte Phrase lautet: „Spiele etwas vom Komponisten Widmann ab“. Daraus folgt, dass sowohl der Begriff „Komponist“ als auch die Mehrzahl„Komponisten“ verstanden werden müssen, sodass die Anwendung Umgangssprache, aber auch grammatikalisch korrekte Sprache behandeln kann. Die Phrase „etwas vom“ wird als Beispiel in der Grammatikgestaltung verwendet. Da diese Phrase für das Feld „Titel“ nicht verwendet werden kann, wird für Titel das optionale Wort „mit“ hinzugefügt, sodass der Befehl „Spiele etwas von Komponist Beethoven und mit Titel 9. Sinfonie ab“ verstanden werden kann. Folgende zusätzlichen Spezifizierungen werden in dieser Arbeit verwendet:

- „Etwas von“, „etwas vom“, „etwas von dem“
- „Von“,“vom“,“von dem“
- „Mit“, „mit dem“

Es ist ebenso zu berücksichtigen, dass der Anwender eine Freundlichkeitsfloskel, wie „bitte“ verwendet. Sollte die Anwendung dies nicht verstehen, kann dies zu Enttäuschung bei dem Anwender und zu einer Unzufriedenheit mit dem Programmes führen. Das Wort „bitte“ wird als Beispiel in dieser Anwendung verwendet, sodass der Befehl „Spiele bitte Beethoven ab“ funktioniert.

Eine andere Kategorie sind spezifizierende Begriffe. Hierunter zählen Begriffe, die auf Felder in der Datenbank verweisen, wie „Komponist“, „Solist“, „Orchester“, etc.. Mithilfe dieser Begriffe kann die Suche bei mehrdeutigen Phrasen auf ein Feld in der Datenbank verweisen und somit eine Nachfrage beim Nutzer unnötig machen.

Als letzte Kategorie werden Verknüpfungsbegriffe behandelt, die die Angabe von mehreren Suchparametern natürlicher gestalten sollen. Der Befehl „Spiele Komponist Beethoven Dirigent Bernstein ab“ kann so verwendet werden, wirkt aber aufgrund der fehlenden grammatikalischen Korrektheit unnatürlich. Dies kann durch ein „und“ verbessert werden, sodass der Befehl „Spiele Komponist Beethoven und Dirigent Bernstein ab“ möglich ist. In Kombination mit einem zusätzlich spezifizierenden Begriff, wie „etwas vom“ kann die Phrase „Spiele etwas vom Komponisten Beethoven und mit dem Dirigenten Celibedache ab“ gebildet werden, was eher natürlicher Sprache entspricht.

Um die Fähigkeiten innerhalb des Rahmens dieser Arbeit zu behandeln, wird sprachlichen Zusätzen keine semantische Bedeutung zugewiesen. Dies ermöglicht eine Vermischung dieser sprachlichen Zusätze und schrumpft die Struktur auf ein leichter verständliches Maß. Somit werden nur die semantischen Informationen der Suchfelder und der Suchbegriffe behandelt, welche mit sprachlichen Zusätzen kombiniert werden, um verschiedenste sprachliche Anweisungen des Nutzers verstehen zu können. So wird der Befehl „Spiele mit Komponisten Beethoven und von Dirigent Celibedache“ ebenfalls verstanden, ohne eine inhaltliche Unterscheidung von dem Befehl „Spiele Komponist Beethoven und Dirigent Celibedache ab“.

5.3.2 Struktur des Sprachbefehls

Um die Struktur des Sprachbefehls zu analysieren, werden vorab alle Bestandteile gesammelt:

- Das Schlüsselwort „Spiel“
- Freundlichkeitsfloskel
- Zusätzliche Spezifizierung
- Spezifizierungsbegriff

- Suchbegriff
- Verknüpfungsbegriff
- Das optionale Schlusswort „ab“

Der Sprachbefehl startet immer mit dem Schlüsselwort „Spiel“. Daraufhin erfolgt optional eine Freundlichkeitsfloskel sowie eine zusätzliche Spezifizierung, wie „etwas von“. Auch optional ist ein Spezifizierungsbegriff, wie „Komponist“, absolut notwendig ist ein Suchbegriff, z.B. „Mahler“. Ein Verknüpfungsbegriff, wie „und“, kann bei der Angabe von mehreren Suchbegriffen verwendet werden. Der Befehl endet optional mit dem Schlusswort „ab“.

Das Schlüsselwort „Spiel“, das Schlusswort „ab“ und die Freundlichkeitsfloskel werden nicht wiederholt. Je nach Eingabe des Nutzers werden aber die zusätzliche Spezifizierung, der Spezifizierungsbegriff, der Suchbegriff und der Verknüpfungsbegriff wiederholt, sollte der Anwender mehrere Suchfelder und Suchbegriffe angeben wollen. Abbildung 5 zeigt die das grammatikalische Konstrukt des Sprachbefehls in grafischer Form.

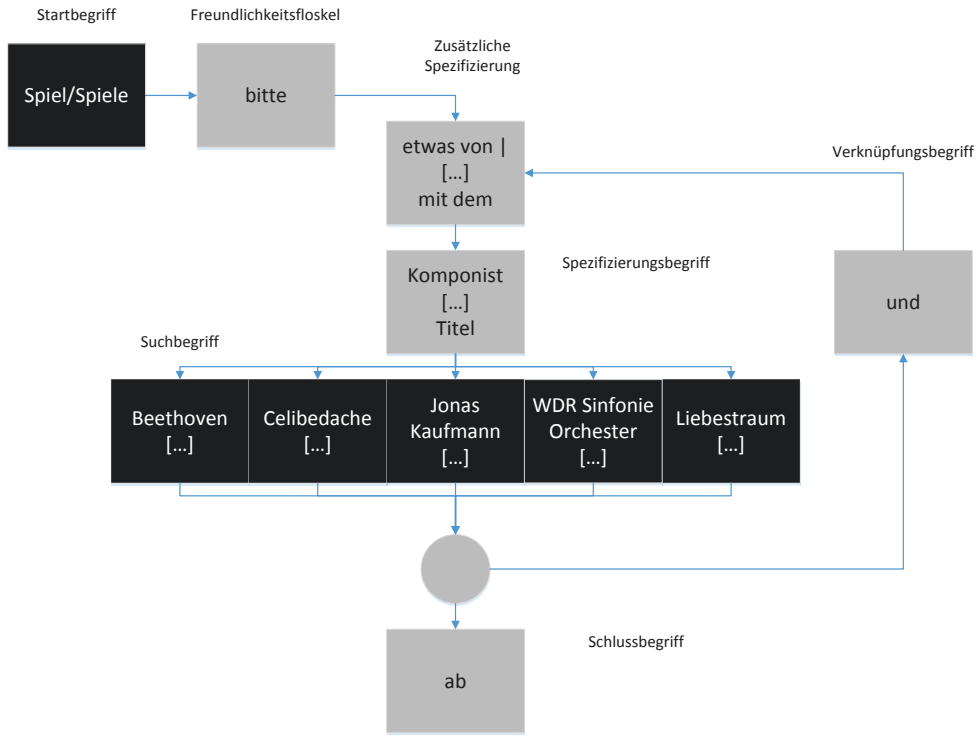

Abbildung 5 Grafische Struktur des Suchbefehls

Diese Abbildung zeigt die Struktur des Suchbefehls in komprimierter Form. Die schwarzen Blöcke entsprechen festen Bestandteilen des Suchbefehls und müssen vom Sprecher verwendet werden. Blöcke in grauer Farbe stellen optionale Begriffe dar und tragen keine semantische Information. Hierbei sind die spezifizierenden Begriffe soweit eine Ausnahme, dass sie in Sonderfällen, in denen ein Suchbegriff jeweils in mehreren Suchfeldern vorkommt, diesen sicher spezifiziert. Da der Verknüpfungsbegriff optional ist, kann dieser verwendet werden oder auch nicht. Eine Anknüpfung der Begriffe ist auch ohne einen Verknüpfungsbegriff möglich. Nach Abbildung 5 sind auch Eingaben, wie „Spiele etwas von Komponist Beethoven und Komponist Mahler ab“ möglich. Da es jedoch in der Datensammlung kein Stück gibt, welches zwei Komponisten aufweist, ist ein solcher Befehl kontraproduktiv, da das Suchergebnis immer leer sein wird. Auch sind Eingaben, wie die Phrase „Spiele etwas von Dirigent Beethoven ab“ möglich. Beethoven ist in der Datensammlung als Dirigent nicht vorhanden, es kann aber von der Sprachsuche verstanden werden. Dies eröffnet Fehlerquellen und kann das korrekte Verstehen des Gesprochenen erschweren. Diese Probleme können gelöst werden, indem alle möglichen Kombinationen von Suchbegriffen mit Suchfeldern bereits vordefiniert wird. Die automatische Erstellung dieser Struktur wird in Kapitel 5.3.3 erläutert. Somit ist eine Suchanfrage „Spiele etwas von Komponist Beethoven und vom Komponist Mahler“ nicht möglich, da eine solche Suchanfrage in der Grammatik schlicht nicht vorgesehen ist. Mit fünf möglichen Suchbegriffen ergibt sich so für die Anzahl an möglichen Wegen in der Grammatik die Rechnung: !5 = 120 (Fakultät von 5). Ein Beispiel für diese Strukturierung zeigt Abbildung 6.

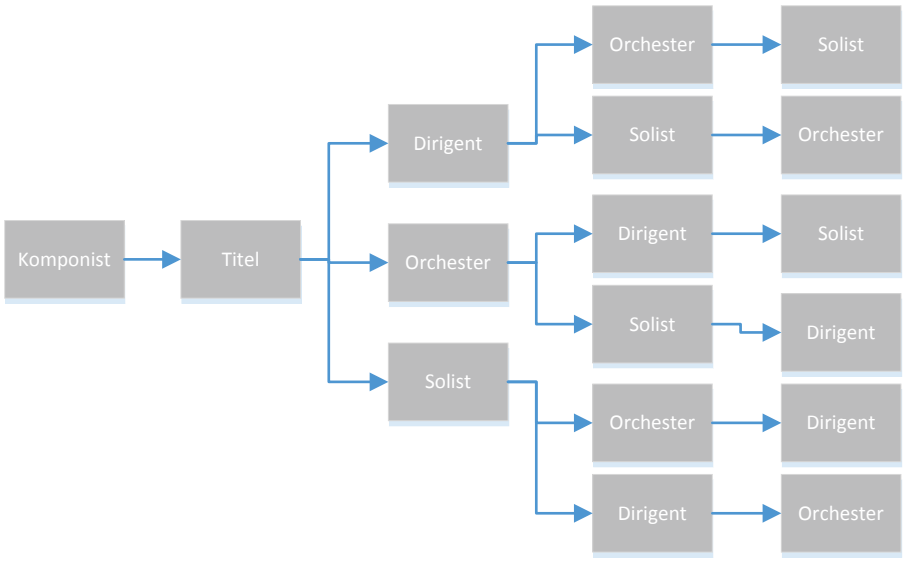

Abbildung 6 Mögliche Wege mit den verwendeten Feldern Komponist und Titel

Abbildung 6 zeigt einen Ausschnitt des Suchbefehls in grafischer Form, in dem bereits die Suchbegriffe „Komponist“ und „Titel“ verwendet wurden. Eine Phrase für den Ausgangszustand kann lauten: „Spiele etwas von Beethoven mit Titel 9. Sinfonie und…“. In der Grammatik selbst werden somit alle möglichen Wege fest strukturiert. Somit ist es auch möglich, dem Suchbegriff ein Suchfeld direkt zuzuweisen. Hat der Anwender bereits den Namen eines Komponisten verwendet, so ermöglicht diese Struktur die Nutzung des Feldes „Komponisten“ nicht erneut, sondern nur die der vier anderen Felder „Titel“, „Dirigent“, „Orchester“ und „Solist“. Wird also der Begriff „Widmann“, welcher Musiker und Komponist ist und der Begriff „Beethoven“ verwendet, so ist nur der folgende grammatikalische Weg möglich: Musiker, Komponist. Für den Sonderfall, dass nur die Phrase „Spiele Widmann ab“ gesprochen wird, muss die Anwendung zur Spezifikation eine Rückfrage stellen. In Abbildung 6 sind weitere optionale Begriffe bewusst ausgelassen worden, um eine bessere Darstellbarkeit zu ermöglichen.

5.3.3 Automatische Generierung der Grammatik

Um die Musikdatenbank dynamisch zu gestalten, sodass neue Musikstücke ohne manuelle Änderung der Grammatik eingefügt werden können, benötigt es ein System, welches die Grammatik automatisiert generiert.

Die Grammatik besteht aus statischen und dynamischen Bestandteilen. Statische Bestandteile sind neben dem Xml- und dem Grammar Header das Schlüsselwort „Spiel“, zusätzliche Spezifizierungen, Verknüpfungswörter und das optionale Schlüsselwort „ab“. Solange sich die Anzahl der Attribute in der Datenbank nicht verändern und somit der Zweck der Anwendung unverändert bleibt, sind Spezifizierungsbegriffe ebenfalls statische Bestandteile der Grammatik. Die Elemente können je nach Nutzerwunsch vom Entwickler angepasst oder verändert werden. Es wird hier jedoch ein Zustand erreicht, in dem es keiner weiteren Veränderung bedarf. Dies unterscheidet die statischen von den dynamischen Bestandteilen. Der dynamische Teil der Anwendung ist das Feld „Suchbegriff“. Die Datengrundlage dieses Feldes verändert sich mit jeder neuen Datei, welche der Datenbank hinzugefügt wird.

Um eine solche Grammatik zu erzeugen kann eine beliebige Programmiersprache verwendet werden. Diese muss Zugriff auf das gewählte Datenbanksystem ermöglichen und Dateioperationen unterstützen. Die statischen Bestandteile werden in einer Zeichenkette im Programm gespeichert, wo diese bei Bedarf editierbar sind. Um die Struktur des dynamischen Teiles der Grammatik, wie in Kapitel 5.3.2 erläutert, automatisch zu erstellen, bedarf es bei n Attributen n-1 Schleifen. Jede Schleife steht für ein Attribut. Für die fünf genutzten Attribute in dieser Arbeit, werden also vier Schleifen benötigt. In jeder Schleife wird ein Teil der Zeichenkette gebildet, die dem statischen Teil angehängt wird. Jede weitere Schleife muss sicherstellen, dass ein Wert nicht bereits vorgekommen ist, sodass jeder Weg einzigartig ist, wie in Abbildung 6 gezeigt wird.

5.3.4 Alternative Grammatikgestaltung

Die in Kapitel 5.3.3 ausgeführte Gestaltung bringt einen Nachteil mit sich, dessen Relevanz der Diskussion bedarf. Diese Form der Suche ermöglicht es dem Nutzer, unterschiedlichste Datensätze mit einander zu verknüpfen. So kann der Nutzer die Sprachanwendung dazu auffordern, etwas vom Komponisten Beethoven abzuspielen mit der Solisten Anna Larsson. Ein Datensatz, welcher dies unterstützt, ist in den Daten nicht vorhanden, dennoch bildet die Grammatik diese Suche ab. Um diese

Problemstellung zu lösen, muss Abstand genommen werden von der in dieser Arbeit verwendeten Erstellung der Grammatik und ein anderer Ansatz verwendet werden.

In dieser Arbeit sind die Struktur der Grammatik sowie deren mögliche Tiefe vordefiniert. Die Form der Struktur ändert sich nicht, wenn neue Musiktitel der Grammatik hinzugefügt werden. Dies hat den Vorteil, dass die Grammatik vergleichsweise einfach erstellt sowie strukturiert dargestellt werden kann. Die alternative Gestaltung der Grammatik sieht vor, dass die Struktur allein aus den Daten der Musiktitel erstellt wird. Dies hat jedoch den Nachteil, dass potenzielle fehlverstandene Phrasen stärker ins Gewicht fallen. Ein Beispiel hier wird in Abbildung 7 gezeigt.

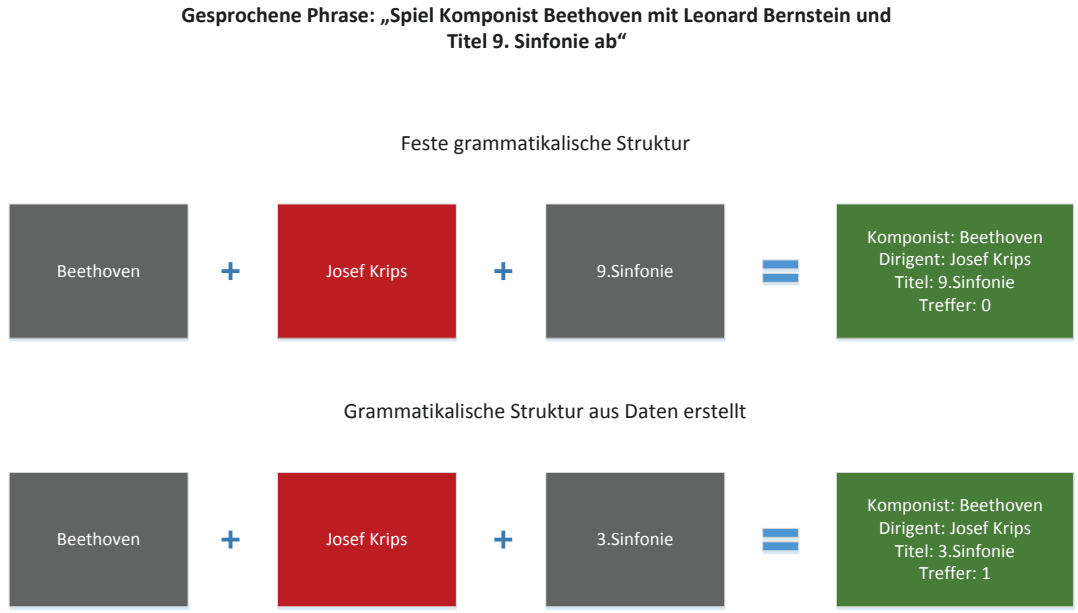

Abbildung 7 Beispiel für fehlerhafte Spracherkennung in unterschiedlichen Grammatikstrukturen

Abbildung 7 zeigt ein Beispiel für eine falsch verstandene Phrase des Nutzers. Der Nutzer spricht die Phrase „Spiel Komponist Beethoven mit Leonard Bernstein und Titel 9. Sinfonie ab“. Aufgrund der Falschen Aussprache des Nutzers, Hintergrundgeräuschen oder ähnlichen Gründen erkennt die Spracherkennung-Engine den Begriff „Leonard Bernstein“ nicht korrekt und interpretiert ihn als „Josef Krips“. Dieses Fehlverstehen wird mit einer Grammatik mit fester Struktur sicher behandelt, da diese die nötigen Freiheiten bei der Sprachgestaltung bereitstellt. Es ist in der Datensammlung kein Stück vorhanden, in dem der Komponist „Beethoven“,

der Dirigent „Josef Krips“ und der Titel „9. Sinfonie“ lautet. Somit ergibt die Treffermenge Null und es wird kein Stück abgespielt. Wird die Grammatik jedoch anhand der Daten dynamisch erstellt, kann dies zum Abspielen eines ungewollten Stückes führen. In dieser Form der Grammatik sind die Wege je nach dem vorherigen Begriff ausgehend strukturiert. Wird also „Beethoven“ und „Josef Krips“ verstanden, so führt der einzige mögliche Weg in der Grammatik auf den Titel „3. Sinfonie“. „9. Sinfonie“ und „3. Sinfonie“ sind sich ähnlich genug, sodass die Anwendung dieses Stück für das Gesuchte hält. Somit wird der Titel abgespielt und der Nutzer muss die Wiedergabe erst stoppen, bevor er erneut eine Sprachsuche tätigen kann. Dies kann mit einer Prüfung des Confidence Wertes abgefangen werden. Jedoch bleibt das Problem bestehen, dass aufgrund der geringen Menge an festgelegten Wegen in der Grammatik die Wahrscheinlichkeit einer Fehlinterpretation steigt.

Ein weiteres Problem ist die Durchführung einer Sprachsuche mit der Eingabe von mehreren Solisten. Da mit der Verwendung einer Grammatik mit fester Struktur jedes Suchfeld nur einmal durchsucht werden kann, ist die Eingabe von einer Phrase, wie „Spiel etwas von Solist Miah Persson und von Solist Anna Larsson ab“ nicht möglich. Dies ist mit einer dynamischen Gestaltung der Struktur aus den Daten möglich. Eine andere Alternative ist das Aufheben der Beschränkung der jeweiligen Suchfeldanzahl von eins, was zu anderen bereits genannten Problem führt oder nur das Zulassen von mehreren Solisten. Da dies jedoch eher ein Sonderfall ist, wird dies in dieser Arbeit nicht näher behandelt.

Jede Form der Gestaltung der Grammatik hat seine Vor- und Nachteile. Diese gilt es vor der Gestaltung der Anwendung gemäß den Wünschen des Nutzers und der jeweiligen Datenlage abzuwägen, um die entsprechenden Nachteile ausgleichbar zu machen.

5.4 Sprachsynthese

Für Nachfragen jeglicher Art bietet die Microsoft Speech Platform eine Sprachsynthese an. Für die Zwecke diese Arbeit kann die Sprachsynthese ohne besondere Gestaltung von sprachlichen Ausdrücken kreiert werden. Um die Sprachsynthese in

der Anwendung zu verwenden, wird die Klasse „Speechsynthesizer“ benötigt. Das Ausgabegerät muss auf das Standard Ausgabegerät festgelegt und eine Stimme ausgewählt werden. Für die Verwendung in der deutschen Sprache bietet sich die Stimme „Hedda“ an.[65] Sind diese Schritte vollzogen, kann die Sprachsynthese verwendet werden. Mithilfe der „Speak“ Methode spricht der Sprachsynthesizer über das angegebene Ausgabegerät die eingegebene Zeichenkette aus.

Die Sprachsynthese wird u.a. für Szenarien benötigt, in denen der Nutzer über Fehler oder Mehrdeutigkeiten informiert werden muss. Als Beispiel hierfür gilt die Phrase „Spiele Widmann ab“, wie bereits in Kapitel 5.3.2 behandelt wurde. Für diesen Fall kann die Sprachwiedergabe lauten: „Meinten Sie Widmann als Komponisten oder Widmann als Solisten?“. Die Suchfelder, die einer Spezifizierung bedürfen, können den alternativen Phrasen entnommen werden. Besitzen mindestens zwei Fragen den gleichen Wortlaut, so muss die semantische Information des Suchfeldes spezifiziert werden. So lautet die Phrase mit den Variablen „<suchfeld1>“, “<suchfeld2> und „<suchbegriff>“: „Meinten Sie <suchbegriff> als <suchfeld1> oder <suchbegriff> als <suchfeld2>?“. Diese Phrase wird je nach Anzahl der unklaren Suchfelder erweitert.

Ein anderer Anwendungsfall ist die Nachfrage bei einer Phrase mit einem niedrigen Confidence Wert, der nach Nutzerwünschen festgelegt wurde. Für die Beispielphrase „Spiele Beethoven und Celibedache ab“ kann die Schreibweise mit der Variablen „<suchkombination>“, welche die Kombinationen aus Suchbegriffen und Suchfeldern enthält, folgendermaßen aussehen: „Ist das Stück <suchkombination1> und <suchkombination2> korrekt?“. Ausgeschrieben spricht die Sprachsynthese die Phrase „Ist das Stück Komponist Beethoven und Dirigent Celibedache korrekt?“ aus.

Es gibt ebenfalls den Fall, dass für eine Suchphrase mehrere Ergebnisse möglich sind. Dies trifft für allgemeine Suchphrasen wie „Spiele Beethoven ab“ ein. Es gibt hierbei verschiedene Möglichkeiten diesen Fall zu bearbeiten. So ist es möglich, dass grundsätzlich das erste gefundene Stück. abgespielt wird. Eine andere Möglichkeit ist das Nachfragen der Anwendung via Sprachsynthese. Hierbei kann die Anwendung über

[65] Vgl. (Microsoft, 2011)

eine Phrase wie „Bitte spezifizieren Sie Ihre Suche“ den Anwender um eine Wiederholung der Suchphrase mit mehr Informationen bitten. Eine andere Möglichkeit ist die Ausgabe aller gefundenen Stücke. Es ist sinnvoll die Anzahl dieser Musikstücke, welche von der Ausgabe betroffen sind, je nach Anwender individuell festzulegen, da der Zeitraum bis zum Abspielen des Titels ansonsten ein gewünschtes Maß übersteigt. Daraufhin kann der Anwender seine Phrase erneut mit einer Spezifikation aussprechen. Um dieses Problem zu lösen gibt es weitere Möglichkeiten u.a. mit der Einbindung einer neuen Grammatik, welche klare Spezifizierungsphrasen zulässt wie „Ich meinte Widmann als Komponist“. Dies wird jedoch nicht weiter behandelt.

Diese Anwendungen der Sprachsynthese benötigen eine Beantwortung des Nutzers. Hierfür wird ebenfalls eine Grammatik benötigt, welche Ja/Nein Antworten enthält. Eine mögliche Vorgehensweise ist das Laden dieser Antworten-Grammatik in die Spracherkennung-Engine und das temporäre Deaktivieren der Suchgrammatik. Der Nutzer spricht seine Antwort. Daraufhin wird die Suchgrammatik wieder aktiviert. Je nach Eingabe des Nutzers wird das Stück abgespielt oder der Nutzer wird per Sprachsynthese gebeten, das Stück erneut abzuspielen. Dies trifft auch für den Fall zu, dass keine Spracheingabe erkannt wurde. Ein Beispiel hierfür lautet „Bitte wiederholen Sie ihre Suche“.

Durch die Verwendung der Sprachsynthese kann die Kommunikation mit dem Nutzer rein über Sprache abgewickelt werden. Somit gelten weiterhin sämtliche Vorteile, die eine Sprachsteuerung anbietet.

6 Fazit

In dieser Arbeit wurde untersucht, wie mithilfe der Microsoft Speech Platform eine Sprachsuche für eine Musikbibliothek mit klassischer Musik gestaltet werden kann und wie sich diese praktisch realisieren lässt. Hierfür wurden die Fähigkeiten der Microsoft Speech Platform analysiert sowie deren Anwendung in der Praxis ausgearbeitet. Hierbei hat sich gezeigt, dass die Microsoft Speech Platform durchaus starke Werkzeuge bietet um eine Sprachsuche zu gestalten. Besonders die Gestaltung der

Grammatik sowie der Semantik hat einen Großteil des Aufwandes für diese Arbeit für sich beansprucht.

Um eine Sprachsuche für eine Musikbibliothek mit der Microsoft Speech Platform zu erstellen, muss diese zuerst mit den benötigten Bestandteilen auf dem Zielsystem installiert werden. Dazu wird ebenfalls ein Datenbanksystem benötigt. Ist die Installation der Bestandteile abgeschlossen, wird die Datenbank anhand der Musikdateien und gewünschten Funktionen an die Anwendung erstellt. Hierbei werden ebenfalls die Daten analysiert sowie mit den Nutzerwünschen verknüpft. Aus dieser Kombination wird die Gestaltung der Grammatik festgelegt, inklusive sämtlicher zusätzlichen Spezifizierungen und Begriffen. Daraufhin wird die Anwendung nach den Einsatzszenarien und Wünschen des Nutzers entwickelt. Dabei muss die Anwendung in der Lage sein, die Daten aus der Datenbank zu importieren und hieraus die Grammatik automatisiert zu erstellen. Dies kann in einer VoiceXML gespeichert werden, ist für den weiteren Betrieb der Datenbank aber nicht zwangsläufig notwendig. Die erstellte Grammatik wird in die Spracherkennung-Engine geladen. Diese wird gestartet und somit ist die Suche einsatzbereit. Dazu benötigt die Anwendung eine Möglichkeit, das gefundene Musikstück abzuspielen.

Mithilfe dieser Arbeit kann eine voll funktionstüchtige Anwendung geschrieben werden, mit dessen Hilfe der Nutzer eine Sprachsuche in einer Musikbibliothek sprachlich eingeben kann. Hierbei zeigt sich, dass die Verwendung von der Microsoft Speech Platform mit SRGS enorm viel Potenzial bietet und bereits im Jahre 2011 ein mächtiges Werkzeug für die Erstellung von Sprachsystemen war. Die praktische Nutzung von SRGS durch die Microsoft Speech Platform wurde ebenfalls vergleichsweise einfach gestaltet. Sind Programmierkenntnisse beim Entwickler vorhanden, so kann er bereits ohne großen Aufwand eine Anwendung mit der Microsoft Speech Platform erstellen. Die Schwierigkeit kann jedoch je nach Ansprüchen an die Anwendung enorm steigen. Durch die Nutzung von ECMAScript, welche in dieser Arbeit nur in kleinem Rahmen verwendet wurden, sind bei Weitem komplexere Anwendungen möglich, als in dieser Arbeit behandelt wurden. Dies zeigt, dass Microsoft mit der

Microsoft Speech Platform eine wirklich gute Plattform für Sprachanwendungen gelungen ist.

Bedauerlich war jedoch die mangelhafte Einbindung der Betonung in der Microsoft Speech Platform. Die Möglichkeit, mehrere Sprachen in einer Grammatik zu nutzen ist in dieser Plattform nicht gegeben. Somit müssen Betonungen immer manuell eingetragen werden, was einen enorm hohen Aufwand darstellt und in der Praxis kaum Verwendung finden dürfte. Die Möglichkeit für ein anderssprachiges Wort nur die Sprache dieses Wortes anzugeben und somit automatisch eine korrekte Betonung nutzen zu können, ist nach der SRGS gegeben, wurde aber nicht von Microsoft aufgegriffen. Dies wäre wünschenswert gewesen, da somit ein Großteil aller Betonungsproblematiken hätte abgefangen werden können. Als Alternative bleibt nur die manuelle Eingabe. Hier zeigt die Microsoft Speech Platform durchaus Verbesserungspotenzial. Unklar ist, ob die .COM Schnittstelle der Microsoft Speech Platform mehr Möglichkeiten anbietet. Dies setzt jedoch weitere Erfahrung in Programmierung voraus und stellt somit eine Barriere für alle unerfahreneren Programmierer dar.

Außerdem hat sich in der Praxis gezeigt, dass die Spracherkennung-Engine bereits gut in dem Verständnis von Spracheingaben ist. Gut bedeutet aber auch Verbesserungswürdig. So hat es bei Tests zwar funktioniert, es gab jedoch oft kleine Abweichungen von dem wirklich Gesprochenen. Für den Erfolg dieser Arbeit trugen diese keine Relevanz. Um zu einer validen, wissenschaftlichen Bewertung dessen zu kommen, ob eine solche Anwendung ihren Zweck auch in der alltäglichen Nutzung, besonders in den Szenarien, welche in Kapitel 3 angesprochen wurden, erfüllt, sind weitere Untersuchungen notwendig. Ebenso gibt es bei der Gestaltung weiterhin viele offene Fragen. Um die Anwendung in der Praxis für verschiedene Dialekte, Kulturen und Regionen zu gestalten, muss die Gestaltung der Grammatik besonders ausführlich und zusätzlich erforscht werden. Im Optimalfall würde für jedes Vorkommen eines Wortes eine Wahrscheinlichkeit festgelegt, ob es erscheinen kann. Dies kann je nach Persönlichkeitsprofil des Nutzers oder der Region eingestellt werden. Die Erstellung und die Fähigkeiten der Grammatik sind außerordentlich und übertreffen den Zweck

dieser Arbeit bei Weitem. Dennoch ist es vollständigkeitshalber notwendig, diesen Punkt im Fazit aufzunehmen.

Die Vollautomatisierung ist bei normierter Datenlage möglich. Mit einem Crawler können diese Daten gesammelt und über insert-Statements in die Datenbank automatisch eingefügt werden. Daraufhin wird die Grammatik neu generiert und neu in die Anwendung geladen. Spricht der Nutzer nun eine Phrase, kann die Suche mit einer aktualisierten Datenlage stattfinden. Bei Erfolg kann über verschiedene Wege das entsprechende Musikstück abgespielt werden. Ist die Anwendung mit der Datenbank erst einmal eingerichtet, kann sie vollautomatisch funktionieren, sollte dies gewünscht sein.

Aufgrund der Komplexität der Fähigkeiten der Grammatik hat diese Arbeit nur einen Teil der Funktionalitäten verwenden können. Durch die Nutzung von ECMS Skript ist die Nutzung einer Skriptsprache innerhalb der Grammatik möglich. Diese wurde zum erfolgreichen Verwenden der Semantik verwendet, jedoch wurden alle weiteren Möglichkeiten ausgelassen. Weitere Forschung in diesem Gebiet ist äußerst wünschenswert. Ebenso ist es wünschenswert, das Ergebnis dieser Arbeit in einer großen Musiksammlung praktisch zu testen, um somit ihren Nutzung praktisch zu untermauern.

In dieser Arbeit wurde sich auf die Attribute „Komponist“, „Dirigent“, „Solist“, „Orchester“ und „Titel“ beschränkt. Die nahezu unendliche Auswahl an möglichen Attributen machte diese Einschränkung notwendig. Hinter Phrasen wie „Spiele etwas aus der frühen Romantik“ verbirgt sich eine große Komplexität und es bleibt zu hoffen, dass das Ergebnis dieser Arbeit um andere Attribute erweitert werden kann. Neben zeitlichen Attributen existieren auch viele weitere inhaltliche Attribute wie z.B. die Stimmung des Stückes oder die Art des Stückes. Anfragen, wie „Spiele ein ruhiges Klavierstück“ wären somit möglich.

Ebenso ist eine weitere Auseinandersetzung mit der Technologie von Aufnahmegeräten und Filtermethoden notwendig. Es bleibt zu hoffen, dass diese Technologien

sich weiterentwickeln, um der Spracherkennung-Engine auch konstant saubere Daten vorlegen zu können, sodass die Spracherkennung nicht an minderwertigen Mikrophonen oder an mangelhaften Geräuschfiltern scheitert.

7 Ausblick

Im Laufe der letzten 100 Jahre hat die Industrialisierung und die Digitalisierung die Welt verändert. Maschinen nahmen Menschen Arbeit ab, was einerseits zu Arbeitslosigkeit, andererseits zur Verlagerung von Arbeitskraftbedarf geführt hat. Es werden immer mehr Techniker benötigt, was sich durch den aktuellen Mangel an Arbeitskräften im IT Bereich auf dem deutschen Markt wiederspiegelt.[66] Dieser Trend trifft nun durch das Angebot von Sprachassistenten auf den privaten Sektor. Zum Beispiel lassen sich Eigenheime per Sprache steuern, soweit Anbieter dies unterstützen.[67] Es bleibt abzuwarten, wohin diese Entwicklung in der Zukunft führt. Möglich ist eine weitere Vernetzung von Diensten und Angeboten mit Sprachassistenten. Diese werden immer leistungsfähiger und kommen der „Do Engine“, wie sie sich die Entwickler von Siri ursprünglich vorgestellt haben, immer näher. Die Fähigkeiten dieser Assistenten wird voraussichtlich immer weiter zunehmen und ebenfalls die Suche im Internet, aber auch in privaten Musikdatenbanken verfeinern. Sie werden mit immer mehr Daten umgehen können. Somit ist es nur eine Frage der Zeit, bis die Sprachassistenten in tiefere Details eintauchen und auf Datenbanken zugreifen können, die Aufführungen von Musikstücken mit vielen Attributen verknüpft haben.

Diese Arbeit ist ein Teil in dieser Entwicklung. Somit können beispielsweise in Automobilsystemen Anwendungen entwickelt werden, die eine erweiterte Sprachsuche nach Musikstücken ermöglichen. Mit der Nutzung von weiteren Attributen kann die Wiedergabe immer weiter spezialisiert werden, bis zu einem Detailgrad, der grundlegende Eigenschaften des Musikstückes ändern kann. So ist der Computer aus der Fernsehserie „Star Trek The Next Generation“ in der Lage, Musikstücke inhaltlich zu manipulieren und akzeptiert somit Befehle wie „Etwas mehr Gitarre bitte“. Mithilfe von künstlicher Intelligenz und weiterer Forschung auf diesem Gebiet könnte eine

[66] Vgl. (Demmer, 2014)
[67] Vgl. (Apple, kein Datum)

Anwendung Bestandteile von Musikstücken, welche inhaltlich Gemeinsamkeiten besitzen, vermischen und somit ein neues Stück kreieren. Mithilfe von Sensoren, welche den Zustand des Nutzers erfassen, sowie beispielsweise auf Basis von Kalenderdaten kann eine Anwendung spekulieren, welche Art von Musik der Nutzer gerne hören möchte. Somit spielt der Nutzer nach einem anstrengenden Tag des Nutzers ruhige Musik zum Entspannen ab. Die Anwendung kann uns Musikstücke empfehlen, welche uns in unserer aktuellen Lage und nach unserem persönlichen Musikgeschmack, Entspannung oder Freude bringen können.

Um dies zu ermöglichen, wird viel von Wissenschaft, der Gesellschaft und vom privaten Anwender verlangt. Die Wissenschaft muss die benötigte Forschung liefern. Es werden normierte Datenstrukturen benötigt sowie ausführlichste Beschreibungen der einzelnen Musikstücke. Es werden Programme benötigt, die diese Informationen interpretieren können. All dies muss entweder von einem Anbieter oder international genormt geschehen, sodass alle Menschen diese Technologie nutzen können. Von der Gesellschaft wird eine Weiterentwicklung im Bereich der Ethik benötigt. All die Informationen, die ein solcher digitaler privater Assistent benötigt, werden aktuell von Unternehmen aller Art gesammelt. So manchem scheint es, dass eine Art Rennen begonnen zu haben, welcher der IT-Unternehmen die größte Datensammlung vorweisen kann. Solange mit diesen Daten gehandelt wird und diese Daten über den Erfolg und Misserfolg von Unternehmen entscheiden, wird die Entwicklung von Anwendungen allein nach den Wünschen derer kontrolliert, die sich einen eigenen Vorteil davon erhoffen. Sollte die Gesellschaft diesen Punkt erreicht haben, wird vom privaten Anwender Vertrauen in die Anwendung benötigt. Eine solche Anwendung funktioniert erst dann vollständig, wenn der Anwender ihr alle möglichen Daten zukommen lässt und diese private Überwachung zulässt. Wenn die Wissenschaft diesen Stand der Forschung erreicht und diese Überwachung nicht von Parteien missbraucht wird und der Anwender darauf vertraut, dass diese Anwendung für sein Wohl gestaltet wurde, dann kann eine solche Anwendung auch existieren.[68]

[68] Vgl. (Frick, 2016)

Diese Arbeit ist nur ein kleiner Schritt in diese Richtung. Es ist wünschenswert, dass diese Thematik weiter erforscht wird, mit sämtlichen datenschutzrechtlichen Aspekten und allen Vor- und Nachteilen, sodass am Ende dieser Entwicklung nicht die wirtschaftlichen Aspekte von Konzernen diese Forschung bestimmen, sondern die Wünsche des privaten Nutzers. So spielt in einer Welt, in der kein Geld benötigt wird wie in der Serie „Star Trek The Next Generation“, auch der Datenschutz keine finanzielle Rolle mehr und die Entwicklung und Forschung kann sich darauf konzentrieren, was wirklich zählt, auf die technische Machbarkeit und die Wünsche des Nutzers.

8 Literaturverzeichnis

ADAC. (kein Datum). *ADAC Test Sprachsteuerung 2014*. Abgerufen am 30. 8 2016 von adac.de: https://www.adac.de/infotestrat/tests/assistenzsysteme/sprachsteuerung_2014/

Apple. (kein Datum). *HomeKit*. Abgerufen am 30. 8 2016 von apple.com: http://www.apple.com/de/ios/homekit/

Apple.1. (kein Datum). *„Hey Siri, wo gibt es das beste Sushi der Stadt?“*. Abgerufen am 30. 8 2016 von apple.com: http://www.apple.com/de/ios/siri/

Bagshaw, P., Burnett, D. C., Carter, J., & Scahill, F. (14. 10 2008). *Pronunciation Lexicon Specification (PLS) Version 1.0.* Abgerufen am 30. 8 2016 von w3.org: https://www.w3.org/TR/pronunciation-lexicon/

Becker, L. (25. 5 2016). *Hey Siri: Bundesdatenschutzbeauftragte warnt vor lauschenden Sprachassistenten*. Abgerufen am 25. 5 2016 von heise.de: http://www.heise.de/mac-and-i/meldung/Hey-Siri-Bundesdatenschutzbeauftragte-warnt-vor-lauschenden-Sprachassistenten-3218349.html

billiger-telefonieren.de. (17. 7 2013). *Taugt das Smartphone als Aufnahmegerät?* Abgerufen am 30. 8 2016 von billiger-telefonieren.de: http://www.billiger-telefonieren.de/handy/nachrichten/taugt-das-smartphone-als-aufnahmegeraet_33398.html

Bosker, B. (22. 1 2013). *SIRI RISING: The Inside Story Of Siri’s Origins — And Why She Could Overshadow The iPhone*. Abgerufen am 30. 8 2016 von huffingtonpost.com: http://www.huffingtonpost.com/2013/01/22/siri-do-engine-apple-iphone_n_2499165.html

Brandt, M. (31. 5 2016). *Anwendungsbereiche von digitalen Sprachassistenten*. Abgerufen am 30. 8 2016 von de.statista.com: https://de.statista.com/infografik/4928/anwendungsbereiche-von-digitalen-sprachassistenten/

Brandt, M. (20. 4 2016). *Nutzung von Sprachassistenten in Deutschland*. Abgerufen am 30. 8 2016 von de.statista.com: https://de.statista.com/infografik/4686/nutzung-von-sprachassistenten-in-deutschland/

Burnett, D. C., Walker, M. R., & Hunt, A. (7. 9 2004). *Speech Synthesis Markup Language (SSML) Version 1.0*. Abgerufen am 30. 8 2016 von w3.org: https://www.w3.org/TR/speech-synthesis/#S3.1.8

Demmer, C. (10. 3 2014). *"Deutschland fehlen IT-Experten"*. Abgerufen am 30. 8 2016 von sueddeutsche.de: http://www.sueddeutsche.de/karriere/fachkraefte-in-der-it-branche-deutschland-fehlen-it-experten-1.1908381

Deutschlandradio Kultur. (12. 7 2015). *Das Lied der Klarinette*. Abgerufen am 30. 8 2016 von deutschlandradiokultur.de: http://www.deutschlandradiokultur.de/joerg-widmann-als-solist-und-komponist-das-lied-der.1091.de.html?dram:article_id=323917

Frick, W. (7. 1 2016). *BIG DATA, BIG CONFUSION, BIG DISTANCE*. Abgerufen am 30. 8 2016 von capital.de: http://www.capital.de/themen/big-data-big-confusion-big-distance.html

Google.1. (kein Datum). *"Ok Google" auf Ihrem Android-Gerät aktivieren*. Abgerufen am 30. 8 2016 von support.google.com: https://support.google.com/websearch/answer/6031948?hl=de

Google.2. (kein Datum). *Sag einfach, was du wissen willst.* Abgerufen am 30. 8 2016 von google.de: http://www.google.de/search/about/features/

Google.3. (kein Datum). *App herunterladen*. Abgerufen am 30. 8 2016 von google.de: http://www.google.de/search/about/download/

Google.4. (kein Datum). *Was die Google App sonst noch alles kann*. Abgerufen am 30. 8 2016 von google.de: http://www.google.de/search/about/learn-more/

Google.5. (kein Datum). *Google Now integrations*. Abgerufen am 30. 8 2016 von google.com: https://www.google.com/landing/now/integrations.html

Google.6. (kein Datum). *Titel erkennen, die in der Umgebung abgespielt werden*. Abgerufen am 30. 8 2016 von support.google.com: https://support.google.com/googleplay/answer/2913276?hl=de

IBM. (kein Datum). *Adding DTMF input*. Abgerufen am 30. 8 2016 von ibm.com: http://www.ibm.com/support/knowledgecenter/SSMQSV_6.1.1/com.ibm.voicetools.sed.editor.doc/tvxaddtmf.html

Kerkmann, C. (25. 4 2014). *Der langsame Abschied von Nokia*. Abgerufen am 30. 8 2016 von handelsblatt.com: http://www.handelsblatt.com/unternehmen/it-medien/microsoft-kauft-handysparte-der-langsame-abschied-von-nokia/9801562.html

Klausing, H. (15. 12 2015). *Was weiß Google über mich?* Abgerufen am 30. 8 2016 von haz.de: http://www.haz.de/Nachrichten/Medien/Netzwelt/Die-Suchmaschine-und-der-Datenschutz-Was-weiss-Google-ueber-mich

Microsoft. (30. 12 2011). *Microsoft Speech Platform - Runtime (Version 11)*. Abgerufen am 30. 8 2016 von microsoft.com: https://www.microsoft.com/en-us/download/details.aspx?id=27225

Microsoft. (25. 10 2011). *Microsoft Speech Platform - Runtime Languages (Version 11)*. Abgerufen am 30. 8 2016 von microsoft.com: https://www.microsoft.com/en-us/download/details.aspx?id=27224

Microsoft. (13. 1 2012). *Microsoft Speech Platform - Software Development Kit (SDK) (Version 11)*. Abgerufen am 30. 8 2016 von microsoft.com: https://www.microsoft.com/en-us/download/details.aspx?id=27226

Microsoft. (6. 12 2015). *How To Use Speech Recognition in Windows XP*. Abgerufen am 30. 8 2016 von support.microsoft.com: https://support.microsoft.com/en-us/kb/306901

Microsoft. (29. 7 2016). *What is Cortana?* Abgerufen am 30. 8 2016 von support.Microsoft.com.

Microsoft.1. (kein Datum). *Microsoft Speech Platform Overview*. Abgerufen am 30. 8 2016 von msdn.Microsoft.com: https://msdn.microsoft.com/en-us/library/jj127858.aspx

Microsoft.10. (kein Datum). *RecognizedPhrase.Confidence Property*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/microsoft.speech.recognition.recognizedphrase.confidence.aspx

Microsoft.11. (kein Datum). *Grammars Overview (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/hh378458(v=office.14).aspx

Microsoft.12. (kein Datum). *Grammar Rules (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/office/hh362887(v=office.14).aspx

Microsoft.13. (kein Datum). *grammar Element (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/hh362814(v=office.14).aspx

Microsoft.14. (kein Datum). *Using Custom Pronunciations*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/hh378403.aspx

Microsoft.15. (kein Datum). *Consonants (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/hh362821(v=office.14).aspx

Microsoft.16. (kein Datum). *tag Element (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/hh378486(v=office.14).aspx

Microsoft.17. (kein Datum). *Control Voice Attributes (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/office/hh362932(v=office.14).aspx

Microsoft.2. (kein Datum). *Microsoft Speech Platform*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/vs/alm/hh361572(v=office.14).aspx

Microsoft.3. (kein Datum). *Benefits of Using the Microsoft Speech Platform SDK 11*. Abgerufen am 30. 8 2016 von msdn.microsoft.com:

https://msdn.microsoft.com/en-us/library/office/hh362943(v=office.14).aspx

Microsoft.4. (kein Datum). *Microsoft Speech Platform Native Code API Documentation*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/jj127857

Microsoft.5. (kein Datum). *Microsoft Speech Platform SDK 11 Requirements and Installation*. Abgerufen am 30. 8 2016 von msdn.Microsoft.com: https://msdn.microsoft.com/en-us/library/vs/alm/hh362873(v=office.14).aspx

Microsoft.6. (kein Datum). *Language Support*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/hh378476(v=office.14).aspx

Microsoft.7. (kein Datum). *How Speech Recognition works (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.Microsoft.com: https://msdn.microsoft.com/en-us/library/office/hh378337(v=office.14).aspx

Microsoft.8. (kein Datum). *Audio Input for Recognition (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/office/hh378436(v=office.14).aspx

Microsoft.9. (kein Datum). *Purpose of Grammars (Microsoft.Speech)*. Abgerufen am 30. 8 2016 von msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/office/hh378342(v=office.14).aspx

Milanesi, C. (3. 6 2016). *Voice Assistant Anyone? Yes please, but not in public!* Von http://creativestrategies.com/: http://creativestrategies.com/voice-assistant-anyone-yes-please-but-not-in-public/ abgerufen

Mozilla. (22. 7 2016). *Array.prototype.push()*. Abgerufen am 30. 8 2016 von developer.mozilla.org: https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/Array/push

Peters, M. (15. 10 2015). *Windows 10: Cortana offline verwenden – so funktioniert's*. Abgerufen am 30. 8 2016 von praxistipps.chip.de: http://praxistipps.chip.de/windows-10-cortana-offline-verwenden-so-funktionierts_43906

Pierce, D. (19. 1 2015). *Amazon Echo review: listen up*. Abgerufen am 30. 8 2016 von theverge.com: http://www.theverge.com/2015/1/19/7548059/amazon-echo-review-speaker

Raj, B., Virtanen, T., Chaudhuri, S., & Singh, R. (kein Datum). *NON-NEGATIVE MATRIX FACTORIZATION BASED COMPENSATION OF MUSIC FOR AUTOMATIC SPEECH RECOGNITION.* Abgerufen am 30. 8 2016 von Tampere University of Technology: http://www.cs.tut.fi/sgn/arg/music/tuomasv/nmf_compensation.pdf

Spät, P. (9. 2 2015). *Adieu, Jobs! Willkommen, Maschine!* Abgerufen am 30. 8 2016 von Zeit.de: http://www.zeit.de/karriere/2015-01/kapitalismus-arbeitsplaetze-digitalisierung-maschinen

Thompson, C. (24. 6 2015). *Lexus says it built a real hoverboard*. Abgerufen am 30. 8 2016 von businessinsider.com: http://www.businessinsider.com/lexus-hoverboard-2015-6?IR=T

Tichelen, L., & Burke, D. (5. 4 2007). *Semantic Interpretation for Speech Recognition (SISR) Version 1.0.* Abgerufen am 30. 8 2016 von w3.org: https://www.w3.org/TR/semantic-interpretation/

windowsunited. (30. 10 2015). *Musik-Erkennungsdienst Shazam erhält Cortana Support unter Windows 10*. Abgerufen am 30. 8 2016 von windowsunited.de: https://windowsunited.de/2015/10/30/musik-erkennungsdienst-shazam-erhaelt-cortana-support-unter-windows-10/

Wolf, A. E., Hess, T., & Benlian, A. (2012). *Nutzen digitaler Mehrwertdienste.* Abgerufen am 30. 8 2016 von Digitale Bibliothek Braunschweig: http://rzbl04.biblio.etc.tu-bs.de:8080/docportal/servlets/MCRFileNodeServlet/DocPortal_derivate_00027372/Beitrag276.pdf

Zaslaw, N. (17. 12 1997). *Der neue Köchel*. Abgerufen am 30. 8 2016 von mozartproject.org (über web.archive.org): https://web.archive.org/web/20110717002801/http://www.mozartproject.org/essays/zaslaw.html

ZEIT ONLINE. (6. 1 2016). *Google soll keine E-Mails mehr scannen dürfen*. Abgerufen am 30. 8 2016 von Zeit.de: http://www.zeit.de/digital/datenschutz/2016-01/verbraucherschutz-google-gmail-abmahnung-werbung

9 Abbildungsverzeichnis

10 Anhang

Titel	Orchester	Solist	Dirigent	Komponist
Turandot Akt. 3 | Nessun Dorma	BBC Symphonie Orchester	Jonas Kaufmann	Martin Alsop	Giacomo Puccini
Turandot Akt. 3 | Nessun Dorma	-	Paul Potts	-	Giacomo Puccini
Turandot Akt. 3 | Nessun Dorma	London Symphony Orchestra	Luciano Pavarotti	-	Giacomo Puccini

Die Walküre, erster Aufzug | „Ein Schwert verhieß mir der Vater“	Orchester der Deutschen Oper Berlin	Jonas Kaufmann	Donald Runnicles	Richard Wagner
Tannhäuser Akt. 3 | Inbrunst im Herzen	Orchester der Deutschen Oper Berlin	Jonas Kaufmann	Donald Runnicles	Richard Wagner
Oh holy night	-	Paul Potts	-	Adolphe Adam
Narrhalla Marsch	Das große Original Karnevalsorchester	-	-	Adolphe Adam
Oh holy night	National Philharmonic Orchestre	Luciano Pavarotti	Kurt Herbert Adler	Adolphe Adam
Symphonie Nummer 2 | Auferstehung | 2. Sinfonie	Great Simon Bolivar Symphony Orchestra	Miah Persson, Anna Larsson	Gustavo Dudamel	Gustav Mahler
Symphonie Nummer 3 |	Wiener Philharmonie Orchester	Christa Ludwig	Leonard Bernstein	Gustav Mahler
Symphonie Nummer 3 | Eroica | 3. Sinfonie	Wiener Philharmonie Orchester	-	Leonard Bernstein	Ludwig van Beethoven
Symphonie Nummer. 9 | 9. Sinfonie	London Symphony Orchestra	Jennifer Vyvyan, Shirley Verret, Rudolph Petrak, Donaldson Bell	Josef Krips	Ludwig van Beethoven
Symphonie Nummer. 9 | Aus der Neuen Welt | 9. Sinfonie	Münchner Philharmoniker	-	Sergiu Celibidache	Antonin Dvorak
Symphonie Nummer 5 | 5. Sinfonie	City of Birmingham Symphony Orchestra	-	Andris Nelsons	Antonin Dvorak

Nocturne Op. 9 Nummer 1 | Nocturne Opus 9 Nummer 1	-	Yiundi Li	-	Frederic Chopin
La Campanella	-	Yiundi Li	-	Franz Lizst
Liebestraum	-	Lang Lang	-	Franz Liszt
Nocturne Op. 55 Nummer 1 | Nocturne Opus 55 Nummer 1	-	Virna Kljakovic		Frederic Chopin
Symphonie Nummer 7 | 7. Sinfonie	Münchner Philharmoniker	-	Sergiu Celibidache	Anton Bruckner
Symphonie Nummer 9 | 9. Sinfonie	Münchner Philharmoniker	-	Sergiu Celibidache	Anton Bruckner
Symphonie Nummer 3 | 3. Sinfonie	Münchner Philharmoniker	-	Sergiu Celibidache	Anton Bruckner
Goldberg Variationen	-	Grigory Sokolov	-	Johann Sebastian Bach
Cantata F80| Lasset uns ablegen die Werke der Finsternis	Rheinische Kantorei	Barbara Schlick, Wilfried Jochens, Claudia Schubert, Stephan Schreckenberger	Herrman Max	Wilhelm Friedeman Bach
Fantasie für Klarinette	-	Jörg Widmann	-	Jörg Widmann

Anhang1 Auflistung der genutzten Musikstücke