BROKER-SOFTWARE ZUR GENERISCHEN REALISIERUNG MULTIMODALER APPLIKATIONEN AUF KONVENTIONELLEN SPRACHPLATTFORMEN

Michael Buschbeck[1], Klaus Kasper[2], Herbert Reininger[1],
Lubos Krejsa[1], Martin Wagner[1], Sven Lehnen[1], Matthias Thiel[1], Lars Both[1]

[1]atip GmbH, [2]Hochschule Darmstadt
michael.buschbeck@atip.de

Abstract: Sprache kann in der Mensch-Maschine-Kommunikation häufig sinnvoll durch weitere Ein- und Ausgabemodalitäten ergänzt werden. Während für die Umsetzung reiner Sprachdialoge der Einsatz von Standardsoftware bereits etabliert ist, herrschen für multimodale Anwendungen noch proprietäre Lösungen vor. Der vorliegende Beitrag stellt einen Ansatz vor, wie konventionelle Sprachplattformen durch Einbindung einer Broker-Software in die MRCP-Kommunikation zwischen Dialogmanager, Spracherkenner- und Sprachsynthesesoftware auch zur Entwicklung multimodaler Applikationen verwendet werden können.

1 Einleitung

Natürliche Sprache allein ist in vielen Fällen nicht ausreichend, um bestimmte Sachverhalte zu vermitteln. Auch in der alltäglichen Kommunikation verwenden wir daher häufig Gestik, Mimik oder Skizzen, um Gesprochenes zu unterstützen. Das Konzept von Multimodalität (für die Aufnahme von Informationen) und Multimedialität (für ihre Darstellung) wird daher auch in der Mensch-Maschine-Kommunikation bereits seit längerer Zeit mit Interesse verfolgt. Bis heute entwickelte Lösungen sind jedoch zumeist entweder nur sehr spezifisch für bestimmte Ein- und Ausgabemedien geeignet (z.B. XHTML+Voice [1]) oder verwenden weitgehend proprietäre Software (z.B. die *Mobile Interaction Platform* [2]).

Für die Entwicklung reiner Sprachdialoge hat sich in den letzten Jahren eine Reihe von Industriestandards etabliert, darunter VoiceXML [3] zur Programmierung des Dialogablaufs und MRCP [4] als Kommunikationsprotokoll zwischen Sprachplattformen und Spracherkennungs- und Sprachsynthesesoftware. Basierend auf diesen und verwandten Standards existiert mittlerweile eine Vielzahl von generischen Softwarelösungen, die kommerzielle Reife erreicht haben und produktiv eingesetzt werden. Entsprechende generische Lösungen für multimodale und multimediale Dialoge sind noch nicht verfügbar.

VoiceXML stellt eine Abstraktion des tatsächlichen Dialogablaufs dar, die sich ohne Weiteres auch auf Dialoge mit multimodalen Eingaben und multimedialen Ausgaben abbilden lässt. Eine konventionelle Sprachplattform, die MRCP unterstützt, abstrahiert darüber hinaus auch Ein- und Ausgaben auf die Ebene eines wohldefinierten und weitgehend vom Medium Sprache unabhängigen Transportprotokolls. Der MRCP-Datenstrom zwischen Sprachplattform und Spracherkenner/-synthesesoftware kann also über eine Broker-Software geleitet werden, die ihn mit Eingaben beliebiger Modalitäten (Sprache, grafische Zeigeoperationen usw.) anreichert und Ausgaben des von der Sprachplattform ausgeführten Dialogs zerlegt und an verschiedene Ausgabemedien (Sprache, visuelle Darstellung usw.) weitergibt.

Der vorliegende Beitrag beschreibt eine Umsetzung einer Broker-Software, die diesen Ansatz verfolgt, und erläutert die dabei eingesetzten Verfahren zur Darstellung multimodaler Eingaben und multimedialer Ausgaben im Rahmen der Möglichkeiten einer konventionellen Sprachplattform, die über VoiceXML programmiert werden kann und über MRCP mit Spracherkenner- und Sprachsynthesesoftware kommuniziert.

2 Architektur und technische Grundlagen

2.1 Aufbau einer konventionellen Sprachplattform

Eine konventionelle Sprachplattform, die die in diesem Beitrag vorausgesetzten Anforderungen erfüllt, umfasst mehrere Komponenten:

- einen Dialogmanager, der nach Maßgabe eines in VoiceXML programmierten Dialogablaufs Eingaben anfordert, ihre Ergebnisse verarbeitet und Ausgaben initiiert;
- einen Spracherkenner, der über MRCP vom Dialogmanager gesteuert wird, um zu gegebener Zeit mit Hilfe der im Dialogablauf definierten Grammatiken Spracheingaben entgegenzunehmen und ihre Ergebnisse an den Dialogmanager weiterzureichen;
- eine Sprachsynthesesoftware, die ebenfalls per MRCP an den Dialogmanager angebunden ist und von ihm Anweisungen zur sprachlichen Textausgabe entgegennimmt.

Die Benutzerinteraktion mit einer Sprachplattform findet typischerweise über Telefon statt. Da IP-Telefonie (VoIP) kommerziell zunehmend an Bedeutung gewinnt, unterstützen viele Sprachplattformen mittlerweile neben der direkten Anbindung über dedizierte Telefoniehardware auch den Empfang und die Durchführung von Sprachanrufen über SIP [5].

2.2 Modellierung von Benutzerinteraktion in VoiceXML

Benutzerinteraktion in VoiceXML folgt einem formularorientierten Modell: Jeder Dialogschritt wird durch ein Formular dargestellt, das ein oder mehrere Eingabefelder umfasst. Die Reihenfolge, in der die Felder mit Eingaben gefüllt werden, ist dabei nicht starr vorgegeben, sondern kann in eingeschränktem Rahmen vom Benutzer selbst bestimmt werden (*mixed initiative* und *multi-slot filling*). Der Entwickler des Dialogs kann zu diesem Zweck sowohl für jedes einzelne Eingabefeld als auch das ganze Formular Prompts und Grammatiken festlegen, um dem Benutzer bzw. dem System zu beschreiben, welche Eingaben erwartet werden.

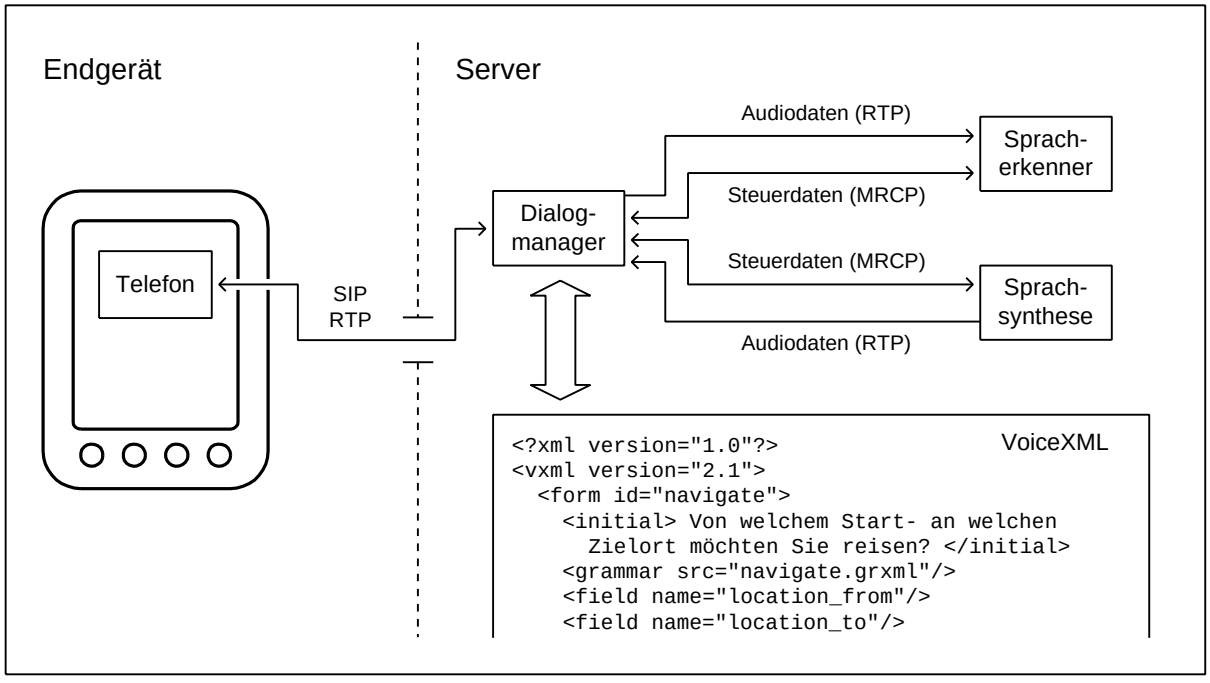


Abbildung 1 – Aufbau einer konventionellen Sprachplattform mit Dialogmanager, Spracherkenner- und Sprachsynthesesoftware, die über MRCP miteinander kommunizieren. Das hier ebenfalls dargestellte Endgerät hat eine Verbindung zur Sprachplattform über SIP hergestellt.

Betritt der Benutzer ein Formular, dann spielt ihm der Dialogmanager zunächst den initialen Prompt vor, der dem gesamten Formular zugeordnet ist, und aktiviert die auf Formularebene definierten Grammatiken. Der Benutzer kann dann eine Eingabe machen, die ein oder mehrere (oder alle) Eingabefelder mit Informationen füllt. Falls danach noch Eingaben offen sind, fragt der Dialogmanager Schritt für Schritt nach den noch fehlenden Informationen und verwendet dabei die den einzelnen Eingabefeldern zugeordneten Prompts und Grammatiken.

Darüber hinaus können global verfügbare *Hyperlinks* definiert werden, die bei jeder Eingabe implizit aktiv sind und mit deren Hilfe der Benutzer jederzeit z.B. an eine andere Stelle des Dialogs springen oder den Dialog beenden kann.

2.3 Multimodality Broker

Der hier vorgestellte *Multimodality Broker* (MMB) fügt sich in die MRCP-Datenströme ein, über die der Dialogmanager mit Spracherkenner- und Sprachsynthesesoftware kommuniziert.

MMB stellt sich dem Dialogmanager als MRCP-Server dar, der eine Spracherkenner- und eine Sprachsynthese-MRCP-Ressource zur Verfügung stellt. Diese beiden Ressourcen sind Schnittstellen zu *Input Broker* und *Output Broker*, denen per Konfiguration Treibermodule für jeweils ein oder mehrere Eingabemodalitäten bzw. Ausgabemedien zur Kenntnis gegeben werden können. In diesem Beitrag wird das Konzept anhand der folgenden von entsprechenden Treibermodulen unterstützten Protokolle illustriert:

- MRCP, um die im Grundaufbau vorhandene Spracherkenner- und Sprachsynthesesoftware und damit Sprache als Ein- und Ausgabemedium verwenden zu können;
- HTTP, um die zusätzliche visuelle Ein- und Ausgabe mit einem grafischen Webbrowser auf dem Endgerät zu ermöglichen.

Über MMB hinaus kommt auf Server und Endgerät nur Standardsoftware zum Einsatz. Eine Anwendung zum Durchführen von IP-Telefonanrufen über SIP und ein grafischer Webbrowser sind auf vielen modernen Mobiltelefonen bereits werkseitig vorinstalliert.

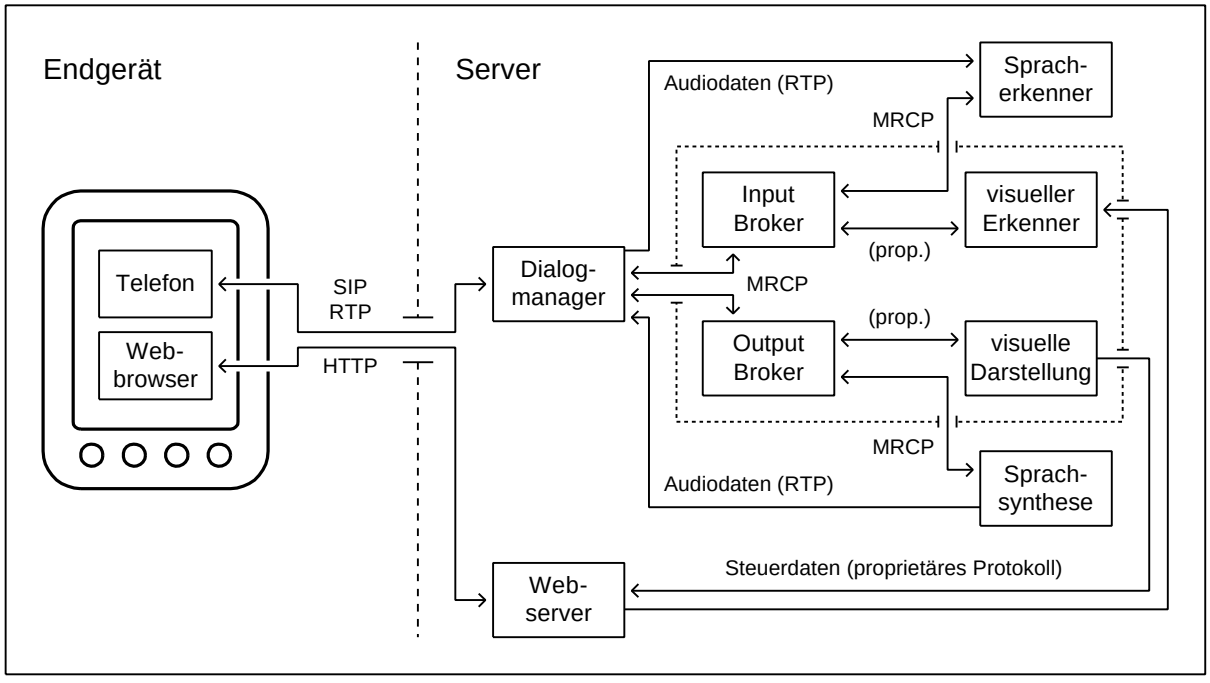


Abbildung 2 – Eingliederung des *Multimodality Broker* (MMB) in die MRCP-Datenströme zwischen den Komponenten einer konventionellen Sprachplattform. Ebenfalls dargestellt ist die Kommunikation der Client-Software auf dem Endgerät mit der Sprachplattform und MMB.

3 Multimodale Eingaben

Die Funktionsweise des MMB sei an einem konkreten Beispiel illustriert: Eine hypothetische Software zur Routenplanung (Abbildung 3) möchte dem Benutzer erlauben, einfache Befehle sprachlich oder durch Berühren entsprechender Knöpfe abzugeben. Komplexere Anweisungen, wie das Anfordern einer Routenbeschreibung zwischen zwei Orten, sollen durch eine Kombination aus dem Antippen der beiden Orte auf der Karte und sprachlich gegebenem Kontext eingegeben werden können.

3.1 Unimodale und multimodale Grammatiken

Jeder unterstützte Erkenner hat ein spezifisches, unimodales Grammatikformat, in dem gültige Eingaben und Eingabekombinationen beschrieben und ihre semantischen Interpretationen definiert werden. Ein verbreiteter Standard zur Darstellung von Sprachgrammatiken ist SRGS [6]. Für grafische Eingaben verwenden wir hier eine entsprechende Form von Grammatik, die antippbare Flächen auf dem Bildschirm definiert und ihnen symbolische Namen zuordnet, die als semantische Interpretation zurückgeliefert werden.

Im Gegensatz zu solchen unimodalen Grammatiken stehen multimodale Grammatiken, die modalitätsspezifische Teilgrammatiken in Form einer Metagrammatik verknüpfen und auf diese Weise modalitätsübergreifend gültige Eingabekombinationen und deren semantische Interpretation festlegen. Die Syntax multimodaler Grammatiken orientiert sich stark an der XML-Form von SRGS und stellt ebenfalls ihre Grundelemente zur Verfügung: Abfolge von Teileingaben, optionale und mehrfach nacheinander durchzuführende Teileingaben und die Auswahl unter mehreren alternativ möglichen Teileingaben. Die Teileingaben selbst sind entweder ihrerseits durch Grundelemente der multimodalen Grammatik oder durch in XML eingebettete modalitätsspezifische Teilgrammatiken definiert.

Die multimodale Grammatik zur Anforderung einer Routenbeschreibung soll es dem Benutzer erlauben, zwei Punkte als Start- bzw. Zielort der Route auf der dargestellten Karte anzu-

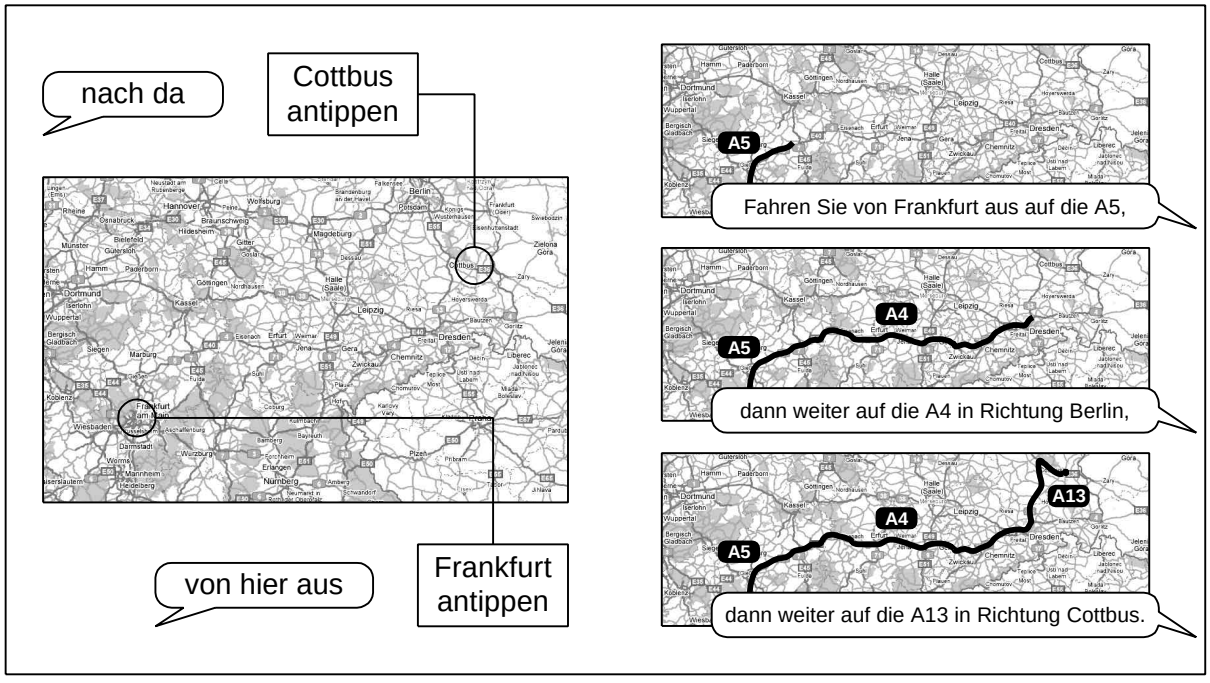


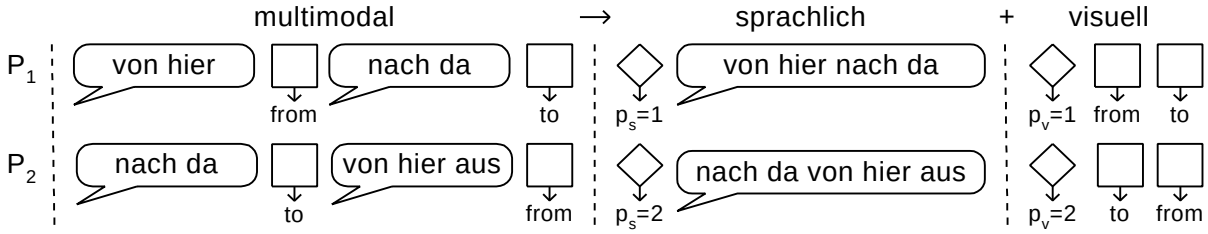
Abbildung 3 – Eine hypothetische Software zur Routenplanung: Der Benutzer kann Anweisungen sprechen und Punkte auf der Karte antippen. Eine Routenbeschreibung wird über einen sprachlichen Befehl und das Antippen zweier Orte auf der Karte angefordert.

tippen. Um eine freiere Bedienung zu ermöglichen, soll aus dem gesprochenen Kontext erschlossen werden, ob Start- oder Zielort zuerst angetippt wurde („von hier nach da“, „nach da, von hier aus“). Daraus ergibt sich die folgende multimodale Grammatikdefinition:

```
<grammar xmlns="http://atip.de/2007/multimodal-grammar"
         xmlns:visual="http://atip.de/2007/visual-grammar"
         xmlns:speech="http://www.w3.org/2001/06/grammar">

  <one-of>

    <item>
      <speech:item> von hier </speech:item>                ← sprachlich
      <visual:area id="map" slot="location_from"/>         ← grafisch
      <speech:item> nach da </speech:item>                 ← sprachlich
      <visual:area id="map" slot="location_to"/>           ← grafisch
    </item>

    <item>
      <speech:item> nach da </speech:item>                 ← sprachlich
      <visual:area id="map" slot="location_to"/>           ← grafisch
      <speech:item> von hier aus </speech:item>            ← sprachlich
      <visual:area id="map" slot="location_from"/>         ← grafisch
    </item>

  </one-of>
</grammar>
```

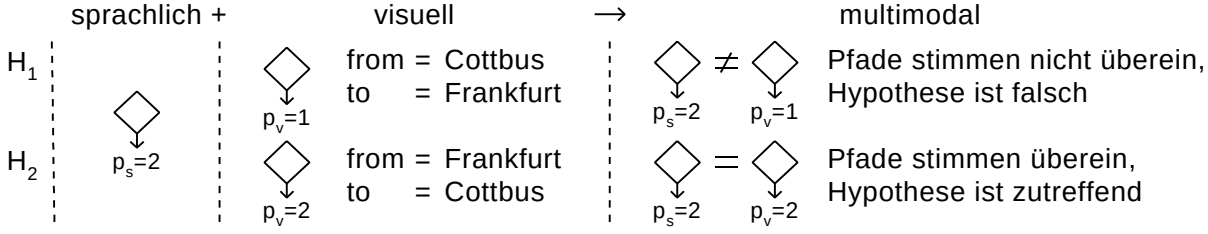
3.2 Zusammenführung einer multimodalen Eingabe

Zur Durchführung der multimodalen Eingabe muss die multimodale Grammatik in ihre unimodalen Bestandteile separiert werden, die dann einzeln bei den jeweiligen Erkennern aktiviert werden:



Um den in der multimodalen Grammatik eingeschlagenen Eingabepfad (P_1 oder P_2) zu verfolgen, wird in jeder unimodalen Grammatik an jedem Entscheidungsknoten eine synthetische Markierung eingefügt, die es dem Input Broker erlaubt, die unimodalen Eingabepfade parallel in der multimodalen Grammatik zu verfolgen. Die beiden Alternativzweige der unimodal-grafischen Grammatik sind in diesem Beispiel ambivalent: Der grafische Erkenner kann nach zwei Klicks auf die Karte allein nicht entscheiden, welcher der beiden Zweige eingeschlagen wurde; er muss also beide Möglichkeiten als Alternativhypothesen zurückliefern und es dem Input Broker überlassen, die korrekte daraus auszuwählen.

Der Benutzer unseres Routenplaners habe Folgendes eingegeben: „Nach da“ [tippt Cottbus an] „von hier aus“ [tippt Frankfurt an]. Die beiden Erkenner liefern folgende Hypothesen:



Anhand des vom Spracherkenner zurückgelieferten Eingabepfads kann der Input Broker erkennen, dass die erste vom visuellen Erkenner zurückgelieferte Hypothese (H_1) nicht korrekt ist. Bei der zweiten Hypothese (H_2) stimmen die Pfade überein, und somit steht das Ergebnis fest: Startort ist Frankfurt, Zielort Cottbus.

3.3 Limitierungen und mögliche Erweiterungen

Das beschriebene Verfahren erfordert von jedem Erkenner, selbständig feststellen zu können, wann seine Teileingabe beendet ist. Spracherkenner lösen das (auch in unimodalen Systemen) typischerweise, indem sie die Spracheingabe nach einem gewissen Zeitraum der Stille als beendet betrachten. Das gleiche Verfahren lässt sich auch auf grafische Eingaben anwenden. Ein dynamischeres Verhalten auf Grundlage der in der multimodalen Grammatik definierten möglichen Gesamteingaben erfordert von den einzelnen Erkennern, eventuell noch komplettierbare Hypothesen als Zwischenergebnisse zurückzumelden. Falls schon für das Zwischenergebnis eine vollständige Übereinstimmung in der multimodalen Grammatik gefunden werden kann, wird die Eingabe beendet; ansonsten wird sie fortgesetzt.

Die aktuelle Implementierung leitet den Audiodatenstrom des Benutzers am Input Broker vorbei direkt an die sprachverarbeitende Eingabemodalität; daher können nicht gleichzeitig mehrere Eingabemodalitäten verwendet werden, die den Audiostrom verwenden (z.B. ein Sprach- und ein Tastentonerkenner). Diese Limitierung zu umgehen würde erfordern, dass der Audiodatenstrom von Input Broker selbst empfangen und dann parallel an mehrere Erkenner in mehreren einzelnen Datenströmen weitergeleitet wird.

4 Multimediale Ausgaben

Die berechnete Routenbeschreibung unseres Routenplaners soll dem Benutzer vorgelesen und gleichzeitig visuell dargestellt werden. Während dem Benutzer die relevanten Wegpunkte beschrieben werden, werden sie auf der dargestellten Karte grafisch hervorgehoben.

4.1 Multimediale Prompts

Im Gegensatz zu Grammatiken können Promptdefinitionen in VoiceXML nicht von extern referenziert werden und müssen sich daher syntaktisch nach den Vorgaben des VoiceXML-Standards orientieren, der die Verwendung von SSML [7] vorschreibt.

Der SSML-Standard lässt allerdings an mehreren Punkten gezielt Spezifikationslücken, um Herstellern von Sprachsynthesesoftware einen Freiraum für eigene Erweiterungen zu lassen. Eine dieser Lücken wird hier verwendet, um Teile eines Prompts bestimmten Ausgabemedien zuzuordnen: Die Stimmauswahl für die Sprachsynthese wird mit einer Ausgabemedienauswahl überladen, die vom Output Broker interpretiert wird. Die multimediale Promptdefinition für die berechnete Route von Frankfurt nach Cottbus sieht somit wie folgt aus:

```
<voice name="medium:visual"> highlight id="A5" </voice>     ← grafisch
Fahren Sie von Frankfurt aus auf die A5,                    ← sprachlich

<voice name="medium:visual"> highlight id="A4" </voice>     ← grafisch
dann weiter auf die A4 in Richtung Berlin,                  ← sprachlich

<voice name="medium:visual"> highlight id="A13" </voice>    ← grafisch
dann weiter auf die A13 in Richtung Cottbus.                ← sprachlich
```

Die sprachlichen Ausgaben in diesem Beispiel werden *synchron* (also hintereinander und an einem Stück) abgespielt, während die eingebetteten visuellen Ausgabebefehle *asynchron* zu den entsprechenden Zeitpunkten während der synchronen Sprachausgabe angestoßen werden. Alle Ausgabeströme werden spätestens am Ende eines explizit definierten Absatzes bzw. am Ende des gesamten Prompts synchronisiert.

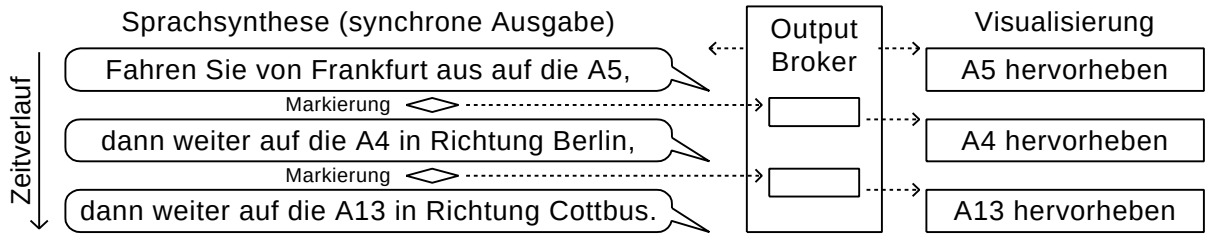
Falls explizit Absätze in SSML-Syntax innerhalb eines Prompts verwendet werden, dann legt das zu Beginn eines Absatzes aktive Ausgabemedium fest, nach welcher Teilausgabe synchronisiert wird. Im folgenden Beispiel wird eine aus mehreren Teilen bestehende visuelle Animation abgespielt und zu Beginn jedes Teils und ganz am Ende jeweils ein sprachlicher Hinweis ausgegeben:

```
<voice name="medium:visual">
  <paragraph>
    <voice name="medium:speech"> Erster Teil. </voice>      ← sprachlich
    animate id="part1"                                      ← grafisch
    <voice name="medium:speech"> Zweiter Teil. </voice>     ← sprachlich
    animate id="part2"                                      ← grafisch
    <voice name="medium:speech"> Das war alles. </voice>    ← sprachlich
  </paragraph>
</voice>
```

4.2 Durchführung der Ausgabe

Jeder Absatz der auszugebenden SSML-Daten wird bei der Ausgabe durch den Output Broker als separate Einheit betrachtet, für die eines der verfügbaren Ausgabemedien synchron abgespielt wird und den zeitlichen Verlauf vorgibt, während die eingebetteten Teilausgaben der übrigen Medien zeitlich synchronisiert zu den jeweils passenden Zeitpunkten angestoßen werden, ohne die synchrone Ausgabe zu beeinflussen. Da davon ausgegangen wird, dass jedes Medium sinnvoll nur einen einzigen Ausgabevorgang zu jeder Zeit durchführen kann, werden auch asynchrone Ausgaben in eine medienspezifische Warteschlange gestellt, falls sie sich überlappen würden, weil die synchrone Ausgabe schneller voranschreitet als die asynchrone.

SSML (und somit jede SSML-konforme Sprachsynthesesoftware) bietet die Möglichkeit, *Markierungen* in einen Ausgabestrom einzubetten, von deren Erreichen während der Ausgabe die aufrufende Software in Echtzeit informiert wird. Der Output Broker nutzt das, um synthetische Markierungen in den für das synchrone Ausgabemedium bestimmten Ausgabestrom einzubetten, die ihm als Wegpunkte für das Anstoßen der asynchronen Ausgaben dienen.



4.3 Limitierungen

Der beschriebene Mechanismus zur zeitlichen Synchronisierung multimedialer Ausgaben stößt bei komplexeren Aufgabenstellungen schnell an Grenzen, die sich im Rahmen von SSML nicht sinnvoll überwinden lassen. Ein speziell für multimediale Ausgaben entworfener Standard ist SMIL [8], dessen Syntax sich allerdings nicht in VoiceXML-2.1-Dokumente einbetten lässt. Diese Limitierung ließe sich allerdings umgehen, indem in SMIL formulierte Prompts über ein triviales Output Broker-Treibermodul als externe Ressourcen referenziert werden. Standardsoftware zum Abspielen von SMIL-Dokumenten ist vielfältig verfügbar. [9]

Der Audiodatenstrom der Sprachsynthesesoftware wird in der aktuellen Implementierung am Output Broker vorbeigeleitet; daher kann nur ein einziges audioerzeugendes Ausgabemedium

gleichzeitig verwendet werden. Um mehrere verwenden zu können, müsste der Output Broker mehrere Audioströme mischen und als einzelnen Datenstrom weitergeben können.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde der Ansatz verfolgt, die Entwicklung multimodaler und multimedialer Dialoge auf konventionellen Sprachplattformen durch eine Broker-Software zu ermöglichen, die sich innerhalb der Sprachplattform in die MRCP-Kommunikation zwischen Dialogmanager, Spracherkenner- und Sprachsynthesesoftware einbindet. Die Kommunikation mit dem Endgerät kann weiterhin über standardisierte Protokolle stattfinden; auf dem Endgerät kann auf proprietäre Software verzichtet werden. Die Dialogentwicklung findet weiterhin in VoiceXML statt. Bereits vorhandene Dialoge können unverändert ausgeführt und sukzessive um multimodale und multimediale Elemente erweitert werden.

Multimodale Eingaben werden über multimodale Grammatiken definiert, die in einer stark an SRGS angelehnten Syntax formuliert werden. Es wurde ein Verfahren beschrieben, um multimodale Grammatiken in unimodale Teilgrammatiken zu separieren und die Einzelergebnisse mehrerer unimodaler Erkenner basierend auf der multimodalen Grammatik zu einer zusammenhängenden Gesamteingabe zu verknüpfen.

Multimediale Ausgaben können im Rahmen des VoiceXML-Standards wie normale Eingabeprompts in SSML formuliert werden. Dabei ist es auch möglich, den Ablauf der Ausgaben verschiedener Medien zeitlich zu synchronisieren. Komplexere multimediale Ausgaben könnten, entsprechende Darstellungssoftware auf dem Endgerät vorausgesetzt, durch Referenzierung externer SMIL-Dokumente ermöglicht werden.

Referenzen

[1] XHTML+Voice Profile 1.0, 2001, http://w3.org/TR/xhtml+voice
[2] Boi, G., Kasper, K. et al.: Eine mobile Interaktionsplattform für multimodale Interaktion. In: Klaus Fellbaum (Hrsg.), Tagungsband der 15. Konferenz Elektronische Sprachverarbeitung, Cottbus, w.e.b. Universitätsverlag, 2004, pp. 261-267
[3] VoiceXML 2.1, 2007, http://w3.org/TR/voicexml21/
[4] Media Resource Control Protocol (MRCP), 2006, http://tools.ietf.org/html/rfc4463
[5] Session Initiation Protocol (SIP), 2002, http://tools.ietf.org/html/rfc3261
[6] Speech Recognition Grammar Specification (SRGS), http://w3.org/TR/speech-grammar/
[7] Speech Synthesis Markup Language (SSML), http://w3.org/TR/speech-synthesis/
[8] Synchronized Multimedia Integration Language (SMIL), http://www.w3.org/TR/SMIL/
[9] W3C Synchronized Multimedia: Players, http://www.w3.org/AudioVideo/#SMIL