Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht - WiSe 2009/10

Kristoffer Witt

Spracherkennung im Kontext Living Lab Hamburg
Szenario und Architektur

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

[overlay]

Inhaltsverzeichnis

1 Szenario

Freitag, später Nachmittag, irgendwo in der Hamburger Innenstadt. Endlich Feierabend! Nach einem anstrengendem Meeting freue ich mich auf einen entspannten Abend. Ich entschließe mich, dem Hungergefühl das mich die letzten zwei Stunden begleitet nachzugeben. „Chinesisch? Junk-Food? Pizza!“ denke ich und ehe ich merke wie mir geschieht habe ich auch schon mein Smartphone gezückt und die Software meiner Wohnung, des Living-Lab-Hamburgs, gestartet.
In der einen Hand das Telefon, mit der anderen meine Notebooktasche fest umklammert stehe ich in der Fußgängerzone. „Guten Abend! Was kann ich für Sie tun?“ erscheint in freundlichen Lettern auf dem Bildschirm, während sich unbemerkt die Verbindung zum Server aufbaut. „Bitte zeig mir die Pizzerien in der Nähe.“ spreche ich in das Headsetmikrofon. Nach kurzer Ladezeit wechselt die Ansicht und der Bildschirm zeigt mir eine Übersichtskarte meines Standorts. Darauf mit großen Buchstaben gekennzeichnet die Italienischen Restaurants und Pizzabringdienste der Umgebung. Ich überfliege kurz die Liste im unteren Teil des Bildschirms und sage dann: „Route von hier zu Marke C“. Die Ansicht wechselt in den Navigationsmodus, ein zweigeteilter Bildschirm, rechts eine Liste mit der chronologisch geordneten Wegbeschreibung, rechts eine Karte mit einem blauen Pfad der mich Quer durch das Hamburger Zentrum führt. Fünf Autominuten später erreiche ich den Italiener meines Vertrauens. Während ich auf mein Essen warte, plane ich noch schnell den Rest des Abends. „Welche Fußballspiele werden heute gezeigt?“, frage ich mein Handy unter den verwunderten Blicken des Restaurantinhabers. Als die Spiele auf dem Bildschirm erscheinen denke ich noch was ein Glück, dass gerade Fußball-WM ist. Eine melodiöse Tonfolge verrät zeigt mir das eintreffen einer Nachricht an. Gespannt klicke ich auf den hüpfenden Briefumschlag. „Moin, heute Fußballgucken bei Dir? :)“. Das trifft sich ja super: schnell „Allen antworten“ angewählt, und einen Zweizeiler diktiert, hätte ich das über die On-Screen-Tastatur eingeben müssen, wäre mir wohl die gute Laune direkt wieder vergangen. Der Abend wird vielleicht nicht entspannt, aber bestimmt unterhaltsam. Mit meinem herrlich duftenden Pizzakarton unter dem Arm und in der Gewissheit, dass mich zu hause sobald ich die Tür öffne ein perfekt temperiertes, warm ausgeleuchtetes Zimmer und meine Lieblingsmusik erwartet, verlasse ich das Restaurant und mache ich auf den Weg nach Hause.
Als die Haustür hinter mir zuschlägt, erinnere ich mich daran, wie ich früher durch die Wohnung laufen musste um Musik anzumachen, die Heizung aufzudrehen und die Lampen anzuschalten. Wie schön ist es, dass meine Wohnung das alles auf Befehl ganz alleine macht, sonst wäre die Pizza jetzt kalt.

2 Einleitung

Sprache als Eingabemodalität für die Steuerung von Systemen bietet sich insbesondere dort an, wo andere Modalitäten nicht verfügbar sind. Sei es wie im beschriebenen Szenario mitten auf der Straße, wenn die Hände zum Tragen genutzt werden, wenn ein Fahrzeug geführt werden soll oder auch in der Küche wenn Speisen zubereitet werden. Ein weiterer Ansatzpunkt ist Sprache für die Eingabe von Texten. Personen die mit mobilen Endgeräten arbeiten haben zum Teil Probleme bei der Nutzung der verfügbaren Tastatur (zum Beispiel aufgrund des fehlenden taktilen Feedbacks (Touchscreen-Tastatur), oder zu kleiner Tasten). In diesen Fällen bietet es sich an, den gewünschten Text zu diktieren.
In dieser Arbeit werden aus dem beschriebenen Szenario Komponenten abstrahiert um Sprache als Eingabemodalität im Kontext „Living-Lab Hamburg“ nutzbar zu machen.
Im ersten Teil wird eine technische Realisation des Szearios beschrieben. Die darausfolgenden Schlüsse werden im zweiten Teil in eine abstrakte Architektur überführt. Den Abschluss bildet einen Ausblick auf die Einsatzmöglichkeit der Architektur im Living-Lab Hamburg.

3 Technische Realisation des Szenarios

3.1 Eingesetzte Technologien

Im Folgenden werden kurz die benutzten Technologien erläutert.

Microsoft Speech API Die Microsoft Speech API bietet out of the box Zugriff auf einen über DOT.NET ansprechbaren Spracherkenner. Sie ermöglicht das dynamische laden von Grammatiken sowie die Assoziation von Schlüsselwörtern mit Semantischen Attributen (Schlüssel- und Wertepaare).

Google Maps,Google Search API Mit diesen beiden APIs ist ein Zugriff auf das Kartenmaterial von Google-Maps und den Suchbestand von Google möglich. Die Maps API ermöglicht automatische Routenplanung, Geocodierung und das hervorheben von Points of Interest. Die Search API ermöglicht es, asynchron den Google Datenbestand abzufragen um zum Beispiel lokale Suchen durchführen zu können.

jquery JQuery ist eine JavaScript-Bibliothek die den Zugriff auf das DOM der Browsers simplifiziert. Weiterhin bietet sie Funktionalität für den asynchrone Abruf von Webseiten per AJAX-Technologie.

3.2 Exemplarischer Ablauf

Dieser Abschnitt beschreibt exemplarisch den Ablauf einer Abfrage aus dem Szenario. Er dient dazu, die Komponenten des Prototypen zu erläutern.

Anfrage „Suche italienische Restaurants“

Precondition: Kontext gesetzt Grammatik geladen Wissenbasis gefüllt Spracherkenner aufnahmebereit Benutzer hat Webseite aufgerufen

- Spracherkenner erkennt Phrase aus Grammatik: „Suche[command:search] (DICTATION[parameter])“ (in eckigen Klammern sind die Semantik-Attribute angegeben, key:value)
- Aus geladenem Kontext wird abgeleitet, das es sich um eine Anfrage für eine spatial lokale Suche handelt.
- Die Phrase wird an den entsprechenden Interpreter weitergeleitet (GoogleMapsInterpreter)
- Der Interpreter erzeugt die XML-Kommando-Datei und speichert sie im Austauschordner.
- Per AJAX-Request wird die XML-Kommando-Datei gepollt und auf neuen Inhalt untersucht.
- Der Such-Befehl wird erkannt, die Abfrage an per Aufruf der Google-API abgesetzt. (Zu beachten aufgrund fehlender spatialer Informationen bezieht sich die Suchabfrage auf den aktuell gewählten Kartenausschnitt, Sinnvoll da dies der Abschnitt ist der dem Benutzer angezeigt wird)
- Die gefundenen Objekte werden dargestellt.
- Über den HTTP-Rückkanal erfolgt eine Meldung an die Software über die gefundenen Lokalitäten.
- Die Lokalitäten werden in die Wissensbasis eingetragen und mit dynamischen Selektions-Begriffen verknüpft. (Marke A, Marke B usw.)
- Benutzer äußert Phrase: „Route[command:route] von hier[parameter_from] nach Marke B[parameter_to]“
- Der Interpreter befragt die Wissenbasis bezüglich der Auflösung der Begriffe „hier“ und „Marke B“ in Attribute des Typs Location. (Der Befehlt „route“ akzeptiert als Parameter entweder Freitexteingaben (nicht in der Wissenbasis vorhanden), oder in der Wissensbasis vorhandene Elemente die das Attribut „is a location“ bzw. „has coordinates“ unterstützen)

- Die Wissensbasis kann „hier“ nicht vollständig auflösen, der Begiff ist aber als overrideable markiert, das bedeutet er wird dem Interpreter weitergeleitet und erst später vollständig aufgelöst“
- „Marke B“ ist bekannt und die Koordinaten der Lokalität werden weitergegeben.
- Der Interpreter erzeugt die XML-Kommando-Datei und speichert sie im Austauschordner.
- Der Route Befehl wird, nach ersetzen des Begriffs „hier“ mit den aktuellen Koordinaten (Kartenmitte) ausgeführt und eine Route berechnet und dargestellt.
- (Nicht implementiert) Die Routendaten werden per HTTP-Rückkanal an die Software übertragen und in die Wissensbasis eingefügt um Navigations Kommandos wie „Nächster Wegpunkt“ oder ähnliche Befehle zu ermöglichen.

Abbildung 1: Bestandteile des Prototyps

3.3 Implementation

Der Benutzer interagiert über sein Eingabegerät (getestet wurde dies mit einem Notebook) mit einer HTML-Seite. Über GPS-Signale wird die Position des Benutzer bestimmt und bei Bedarf an den Anwendungsserver übermittelt. Weiterhin besteht eine direkte Audioverbindung zum Server, im Kontext des Living Labs durch Richt- bzw. Omnidirektionale Mikrophone, im mobilen Kontext beispielsweise über eine Freisprecheinrichtung. Über diese Audioverbindung äußert der Nutzer die zu erkennenden Sprachbefehle, denkbar ist auch eine synthetisierte Antwort. Die Sprachbefehle werden vom Anwendungsserver (SAPI) interpretiert, dies entspricht dem in Minker u. a. (2005) beschriebenen Netzwerkerkennungs-Ansatz. Anhand des Kontextes, abgeleitet aus den Benutzeraktionen oder implizit über Sensoren, wird bestimmt welche Wortschatz (Grammatik) erkannt werden soll. Jeder Grammatik ist einer oder mehrere Interpreter zu geordnet. Ein Interpreter verarbeitet die erkannte Sprachsequenz anhand vordefinierter Semantischer Schlüssel-Wert-Paare. Anhand dieser Paare wird entschieden, welche Aktion durchgeführt werden soll. Diese Aktion wird in eine XML-Datei kodiert und über einen Austausch-Ordner dem Web-Server zugängig gemacht. Der Webserver polled diese Datei und erkennt anhand von Sequenznummern ob ein neuer zu bearbeitender Befehl vorhanden ist. Je nach Befehlsart erfolgt ein Feedback über einen Ajax-HTTP-Kanal zum Anwendungsserver.

Semantik

Für die semantische Analyse wurde eine Datenbankgrundstruktur geschaffen. Diese ermöglicht es Objekte mit Eigenschaften abzubilden und anhand dieser zu durchsuchen. Ebenso ist es möglich Kontexte zu schaffen und Entitäten zu erstellen, deren Attribute durch den aktuellen Kontext festgelegt wurden. Ein Beispiel ist die virtuelle Entität „ich“, je nachdem wer gerade die Anwendung bedient zeigt diese auf die jeweilige Benutzerentität. Vergleichbar ist dieser Ansatz mit dem des Dynamischen Lexikons(siehe (Wahlster, 2006), S.123 ff.).

3.4 Ergebnis

Ein Großteil der Funktionalität die im Szenario beschrieben wurde konnte anhand der vorgestellten Komponenten umgesetzt werden. Bisher nicht getestet ist die Audioübertragung vom mobilen Endgerät zum Spracherkenner.

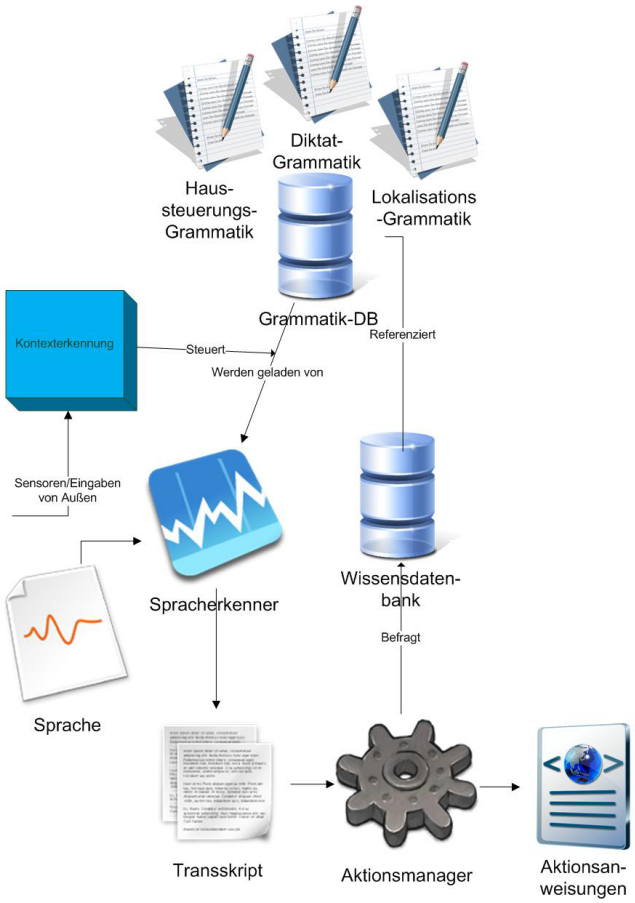

Abbildung 2: Komponenten des Anwendungsservers

4 Architektur

Aus dem Prototypen der das beschriebene Szenario umsetzt, können Anforderungen und Komponenten abgeleitet werden um eine generische Architektur zu schaffen. Die Bestandteile dieses Ansatzes werden im folgenden Abschnitt beschrieben.

4.1 Szenario-Komponenten

4.2 Anwendungsserver

Der Anwendungsserver bildet die Basis für die Umsetzung der Szenarien. Er beinhaltet die Module für Spracherkennung, Daten- und Wissensmanagement sowie die Steuerung der Aktionen.

Wissensdatenbank Die Wissensdatenbank ist die digitale Abbildung des semantischen Wissens über die Welt und Ihre Bestandteile. Sie ist eine stark vereinfachte relationale Abstraktion der real Welt, beschränkt auf die Module die für die Steuerung des Living Labs notwendig sind.

Aktionsmanager Der Aktionsmanager verwaltet die möglichen ausführbaren Prozesse. Er adaptiert die semantischen Informationen der ankommenden Transskripte auf ausführbare Kommandos und delegiert diese an die Haustechnik bzw. in diesem Fall an einen Webserver. Diese Adaption beinhaltet ein auflösen etwaiger Entitäten in die benötigten Detailinformationen durch die Wissensdatenbank.

Spracherkenner Der Spracherkenner übersetzt das digitale Audiosignal in Textdaten. Dies geschieht anhand der vorher geladenen Grammatiken, also vorgefertigten Regeln und Sätzen. Diese Sätze werden dynamisch aus der Grammatik-Datenbank geladen. Im Gegensatz zur vollständig statischen Vorgenerierung der zu erkennenden Sätze, ermöglicht es dieser Ansatz auf Änderungen des Benutzer zu reagieren (zum Beispiel etwaige Landmarken die auf einer Karte dargestellt werden per Name zu referenzieren).

Grammatik-Datenbank Die zu erkennenden Phrasen und Begriffe sind in dieser Datenbank enthalten. Einzelne Worte können hierbei semantische Schlüsselwörter, wie zum Beispiel „parameter“ und Semantikwerte also eindeutigen Referenzen auf Entitäten der Wissenbasis zugeordnet werden. Diese Zuordnung ermöglicht dem Aktionsmanager eine Abbildung auf verfügbare Kommandos und ein vereinfachtes auflösen der Parameter in die benötigten Attribute.

Kontexterkennung In der Kontexterkennung wird festgelegt, welche Grammatiken der Spracherkenner laden soll. Um robuste und eindeutige Resultate zu erhalten, ist es nötig den Erkennungsbereich der Spracherkenner möglichst gering zu halten. Kontexterkennung ist ein äußers komplexes Themengebiet in sich. Vereinfachend werden für diese Arbeit daher explizite Kontextanforderungen genutzt. Gegliedert in Software aktionen: starten eines bestimmten Programmteils, betätigen einer Schaltfläche. Und Sensorik/Hardware-Aktionen: auslösen eines Kontaktes an einer Schublade oder einem Haushaltsgerät, feststellen der Position des Benutzer anhand eines Indoor-Positioning-Systems. Anhand

diese Ereignisse ist es möglich einen Kontext festzulegen und etwaig passende Grammatiken zu laden.

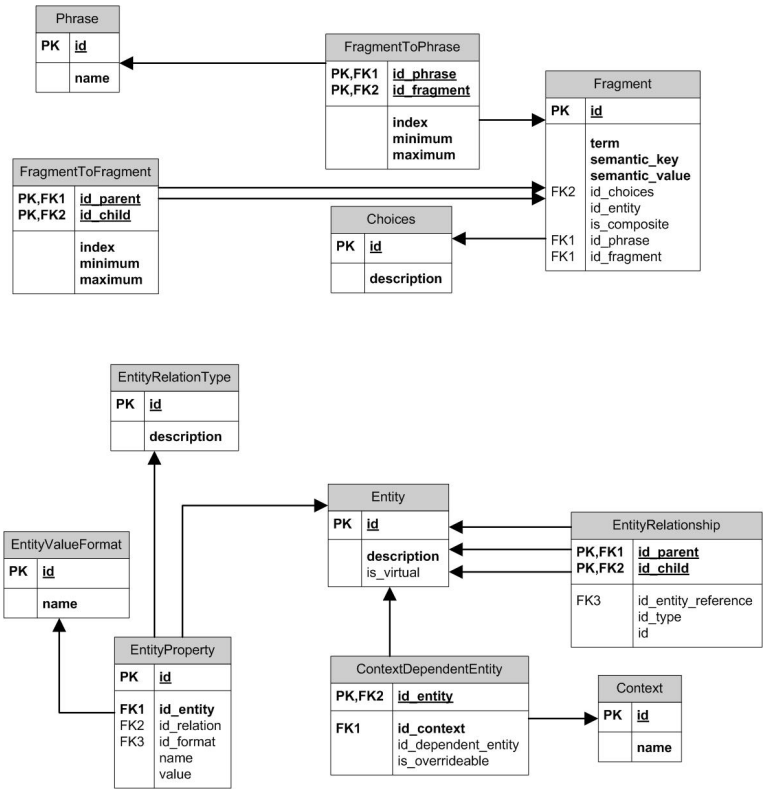

Abbildung 3: Datenbankaufbau

5 Zusammenfassung und Ausblick

Die generische Struktur der abgeleiteten Komponenten ermöglicht eine schnelle Umsetzung von verschiedensten Szenarios. Um dies möglichst einfach zu gestalten wäre es denkbar, eine Oberfläche für die Erstellung der Sprach-Grammatiken zu entwerfen.
Des weiteren wäre eine Entwicklung einer generischen Semantikanalyse denkbar. Aktuell liefert der Spracherkenner Resultate anhand der vorgeladenen Grammatiken. Weicht der Benutzer von diesen Strukturen ab wird die Anfrage zwar über die Diktat-Grammatik erkannt aber nicht weiter ausgewertet. Dieser erkannte Freitext sollte verarbeitet und seine semantischen Inhalte in Kommandos übersetzt werden.
Um nachzuweisen das sich die erdachte Struktur auch für andere Szenarien eignet, wurde mit

Umsetzung eines Küchenszenarios begonnen. In diesem Szenario geht es um die Zubereitung von Speisen, angefangen vom Suchen des Rezeptes, über den Abgleich der vorhandenen und benötigten Zutaten bis zur schrittweisen Zubereitung der Speise. Bisher implementiert ist die Möglichkeit nach Rezepten anhand von Begriffen zu suchen sowie das auslesen und zurückgeben der Zutaten. Denkbar wäre diese Anwendung über ein Display in der Küche auszugeben um bei der Zubereitung von Speisen zu assistieren.

Literatur

[De Mori u. a. 2008] DE MORI, R. ; BECHET, F. ; HAKKANI-TUR, D. ; MCTEAR, M. ; RICCARDI, G. ; TUR, G.: Spoken language understanding. In: *Signal Processing Magazine, IEEE* 25 (2008), May, Nr. 3, S. 50–58. – ISSN 1053-5888

[Grimaldi und Cummins 2008] GRIMALDI, Marco ; CUMMINS, Fred: Speaker Identification Using Instantaneous Frequencies. In: *IEEE Transactions on Audio, Speech and Language Processing* 16 (2008), Nr. 6, S. 1097–1111. – URL http://dblp.uni-trier.de/db/journals/taslp/taslp16.html#GrimaldiC08

[Hannes Mögele 2006] HANNES MÖGELE, Florian S.: LREC06:SmartWeb UMTS Speech Data Collection, The SmartWeb Handheld Corpus. (2006), May

[Minker u. a. 2005] MINKER, Wolfgang (Hrsg.) ; BÜHLER, Dirk (Hrsg.) ; DYBKJÆR, Laila (Hrsg.): *Text, Speech and Language Technology*. Bd. 28: *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*. Dordrecht : Springer, 2005. – ISBN 978-1-4020-3073-4

[Ringlstetter u. a. 2007] RINGLSTETTER, Christoph ; SCHULZ, Klaus U. ; MIHOV, Stoyan: Adaptive text correction with Web-crawled domain-dependent dictionaries. In: *ACM Trans. Speech Lang. Process.* 4 (2007), Nr. 4, S. 9. – ISSN 1550-4875

[Wahlster 2006] WAHLSTER, Wolfgang (Hrsg.): *SmartKom: Foundations of Multimodal Dialogue Systems*. Berlin : Springer, 2006. – ISBN 3-540-23732-1