

Natürlichsprachliche Dialoge in Raumsteuerungssystemen

Markus Berg
Nils Weber
Christoph Eigenstetter
Antje Düsterhöft

Hochschule Wismar
University of Applied Sciences:
Technology, Business and Design

Im Landesforschungsschwerpunkt *Mobile Assistenzsysteme* steht das Projekt *Maike* für die Entwicklung mobiler Assistenzsysteme für intelligente kooperierende Räume und Ensembles. Ziel der Arbeiten ist die Verbesserung der Interaktion mit sprachgesteuerten Anwendungen. Im Vordergrund steht dabei aus Sicht der Hochschule Wismar die Verarbeitung natürlichsprachlicher Äußerungen und die Berücksichtigung komplexer Nutzeraktivitäten. Weiterhin sollen solche Systeme befähigt werden, Befehle auf ihre Plausibilität zu überprüfen und gegebenenfalls optimierte Lösungen abzuleiten. Der folgende Text beschreibt verschiedene in diesem Zusammenhang auftretenden Probleme und stellt mögliche Lösungsansätze vor.

1 Einführung

Das *Natural Language Laboratory* der *Hochschule Wismar* forscht auf dem Gebiet der automatisierten Verarbeitung von natürlicher Sprache. Schwerpunkte stellen dabei, neben der Integration von Spracherkennungs- und Sprachsynthesystemen, die semantische Interpretation von sprachlichen Äußerungen sowie die Modellierung von natürlichsprachlichen Dialogen dar. Dies erfordert neben der Informatik die Einbeziehung benachbarter Disziplinen wie der Psychologie oder der Computerlinguistik. Auch Elemente der Bildverarbeitung, Automatisierungstechnik und theoretischen Informatik fließen in die Forschung mit ein und verdeutlichen den Charakter dieses interdisziplinären, multimedialen Forschungsgebietes, das einen Teil der Mensch-Maschine-Interaktion darstellt.

Die Sprache gilt als wichtigstes Kommunikationsmittel des Menschen. Daher liegt es nahe, sie auch zur Steuerung von Geräten zu verwenden. Gerade ältere und behinderte Menschen werden hierdurch in die Lage versetzt, Alltagssituation zu bewältigen. Doch auch zur Vereinfachung von Bedienungsabläufen ist der Einsatz von Sprache sinnvoll. Dabei ist zu beachten, dass die Sprache eine dynamische Natur hat und einer ständigen Evolution unterliegt. Weiterhin ist die Sprache ein höchst ambivalentes Kommunikationsmedium, das erst durch Kontextinformationen eindeutig wird. Die meisten aktuellen Systeme basieren lediglich auf einem „Command and Control“-Interface, das nur die Verwendung von einzelnen Befehlswörtern erlaubt. Dies stellt einen sehr unkomfortablen Weg der Interaktion dar, da der Anwender zunächst einen Befehlsatz auswendig lernen muss. Daher ist in Zusammenarbeit mit der Basis GmbH aus Wismar¹ ein Prototyp zur Steuerung eines Raumes über die natürliche Sprache entstanden, der die im vorangegangenen Projekt *eOhr* [1] [2] aufgezeigten Probleme adressiert und löst.

¹<http://dz.basis-wismar.de>

2 Verwandte Arbeiten

Auf dem Gebiet der Raumsteuerung gibt es verschiedene bereits abgeschlossene und noch andauernde Projekte. Die meisten stützen sich jedoch auf eine Fernbedienung per Touchscreen oder auf selbstlernende Systeme, die anhand des Nutzerverhaltens Vorschläge unterbreiten. Auf dem Gebiet der Sprachverarbeitung gibt es vergleichsweise wenige Ansätze.

Die in Saarbrücken ansässige Firma *SemVox* entwickelt eine *Ontology-based Dialogue Platform* [3], mit der sich verschiedenste sprachgesteuerte Applikationen erstellen lassen. Eine Sprachsynthese ist vorhanden und wird vorrangig zur Bestätigung der Befehle genutzt. Im Falle von Zweideutigkeiten, wird jedoch ein Nachfragemanagement eingeschaltet. Über eine eigens entwickelte XML-basierte Sprache lassen sich Regeln für das Verhalten des Dialoges und für die Steuerung der Spracherkennung erstellen. Mit Hilfe des ODP wurden verschiedene Beispielapplikationen entwickelt. Sowohl auf der IFA 2008 als auch auf der CeBit 2009 wurden Systeme zum Steuern von Musiksammlungen vorgestellt. Es gibt jedoch auch ein Raumsteuerungssystem mit dem Namen *i2home*², das mit Hilfe eines PDA gesteuert wird und die Spracheingaben an einen Server übermittelt.

Das *European Media Laboratory* hat eine Sprachsteuerung für Mediengeräte entwickelt, die beispielsweise bei dem Befehl „*Ich will das Erste Programm sehen!*“ den Fernseher auf dem entsprechenden Kanal einschaltet [4]. Hierzu wird ein PC (Home Server) an die EIB-Busstruktur angeschlossen. Über Fernbedienungselemente, die mit dem HomeServer verbunden sind (PDA, Mikrofon) können die Geräte gesteuert werden. Die Spracherkennung kann dabei wahlweise auf dem PDA oder auf dem HomeServer erfolgen.

Im *Institut für Multimedialechnik* wurde ein Demonstrator [5] entwickelt, der eine kontextsensitive Raumsteuerung ermöglicht. Hierzu wird der Raum in verschiedene Zonen eingeteilt, wodurch der Befehl „Licht an“ – je nach Position – zum Einschalten einer anderen Lichtquelle führt. Auch bei dieser Lösung existiert keine Dialogkomponente.

Das elektronische Ohr [6] der Firma *Basis GmbH* basiert auf dem Spracherkennungssystem *SpeaKing* und verwendet als Hardwareansteuerungssystem Komponenten der Firma *AMX*. Ein Keyword-Spotting-Verfahren sorgt für die Extraktion des Schaltobjektes und des Schaltbefehls. Eine nähere Beschreibungsmöglichkeit, eine Sprachsynthese sowie ein Dialogmanagement existieren nicht. Somit sind nur einfache Befehle wie „Licht aus“ oder „Ventilator an“ möglich.

3 Dialogmanagement

Wie eingangs bereits erwähnt, ist die natürliche Sprache sehr variant. Oftmals sind Äußerungen nicht eindeutig und in ihrer Bedeutung mehrschichtig. So zielt die Frage „*Kannst du bitte das Licht anmachen?*“ nicht auf die Potenz ab, sondern stellt vielmehr eine indirekte Aufforderung dar und ist somit der pragmatischen Ebene zuzuordnen. Auch emotionale Angaben wie „*mir ist zu kalt*“ stellen auf den ersten Blick keinen direkten Befehl dar, sollen aber dennoch zum Einschalten der Heizung führen. Hierbei ist ebenfalls zu berücksichtigen, dass der Sinn des Adjektives invertiert wird und statt Kälte Wärme erzeugt werden soll.

Der wechselseitige Austausch von Informationen wird Dialog (altgriechisch *dialégesthai*: sich unterhalten) genannt. Zum optimalen Verständnis wird dabei eine korrekte Verwendung von Syntax und Semantik vorausgesetzt. Es ist jedoch erstaunlich, wie stark die Fähigkeit des Menschen entwickelt ist, Fehler (unbewusst) zu korrigieren oder umgangssprachliche Äußerungen richtig einzuordnen, um hierdurch die korrekte Interpretation einer Aussage zu ermöglichen (Sprachperformanz).

²<http://www.i2home.org>

Die Gestaltung des Dialogs ist eng mit der Flexibilität und den potentiellen Fähigkeiten des Systems verbunden. Hat beispielsweise ein Nutzer zeitnah eine Lampe eingeschaltet, bildet die Realisierung einer Anaphernresolution die Voraussetzung dafür, dass diese über den Befehl „*Mach sie wieder aus*“ wieder ausgeschaltet werden kann.

In der Sprachverarbeitung unterscheidet man zwischen den folgenden Dialogformen:

- *systemgeführt* – Das System gibt die Fragen vor, auf die der Nutzer antworten muss. Er hat keine Möglichkeit den Dialog zu lenken, wird jedoch meist erfolgreich an das Ziel geführt. Dies ist jedoch untypisch für eine zwischenmenschliche Kommunikation und daher oft nicht zufriedenstellend.
- *gemischte Initiative* (mixed initiative) – Bei dieser Form der Kommunikation versucht das System den Nutzer zum Ziel zu führen, wobei der Weg dorthin vom Nutzer beeinflusst werden kann.
- *usergeführt* – Beim usergeführten Dialog ist das Ziel unbekannt. Das System reagiert nur und hilft dem Nutzer nicht dabei, ein bestimmtes Ziel zu erreichen. Diese Dialogform ist nicht für Beratungs- und Buchungssysteme geeignet und dient eher der Unterhaltung.

Bei einem Raumsteuerungssystem wird implizit eine domänenspezifische, offene Frage gestellt: „*Was möchten Sie schalten?*“. Wenn das System die Antwort des Nutzers nicht interpretieren kann beziehungsweise diese unvollständig ist, erfolgt eine Nachfrage durch die der Dialogablauf beeinflusst wird. Daher handelt es sich bei diesem Programm um eine gemischte Initiative. Systemgeführte Systeme hingegen erlauben keine offene Frage, sondern stellen eine Folge von konkreten Fragen, was für den Nutzer der Domäne Raumsteuerung eher hinderlich wäre.

Um eine komfortable Bedienung zu ermöglichen, ist es wichtig, stets Informationen über den Gesprächsablauf zu besitzen. Daher werden die einzelnen Dialogschritte in einer Historie gespeichert. Hierdurch ist das Auflösen von Referenzen zwischen verschiedenen Befehlen möglich. Besonderer Fokus liegt dabei auch auf der Unterscheidung zwischen einem neuen Dialogschritt und der Ergänzung eines vorherigen Befehls. Zu untersuchen ist hierbei, wie lange Informationen in der Historie vorgehalten werden müssen bzw. auf wie viele bereits getätigte Befehle sich der Benutzer bezieht.

Wenn ein Befehl nicht eindeutig ist, muss das Dialogmanagement eine Nachfrage auslösen, um den Wunsch des Nutzers genauer spezifizieren zu können. Dabei kann es vorkommen, dass sich der Nutzer nicht kooperativ verhält und statt auf die Frage des Systems zu reagieren einen neuen Befehl gibt. Dies darf nicht zu Fehlerzuständen führen, da es sich hierbei um gängige Phänomene in der alltäglichen zwischenmenschlichen Kommunikation handelt.

4 Implementierung

Zur Umsetzung dieser Ziele ist eine Architektur [7] erforderlich, die eine effektive Verarbeitung der sprachlichen Eingaben ermöglicht. Die Spracherkennung wird durch den Microsoft Recognizer durchgeführt, der über die Microsoft Speech API 5.3 angesprochen wird und SRGS-Grammatiken verarbeitet. Die Sprachsynthese kann durch jede SAPI 5.3 konforme Stimme realisiert werden. Die SAPI-Schnittstelle ermöglicht dem Kunden, jeden kompatiblen Spracherkennung/-synthese zu verwenden. Hierdurch ist dieses Programm unabhängig von einem einzelnen Hersteller. Die Hardware wird über ein AMX-Mediensteuerungssystem kontrolliert, das eine ActiveX-API anbietet. Das Domänen- und Sprachmodell wird in einer OWL-basierten Ontologie gespeichert und über die Jena API abgefragt. Da diese Technologien auf verschiedenen Programmiersprachen basieren, werden XML-Webservices zur Kommunikation eingesetzt.

Das Weltmodell beinhaltet Informationen über die verwendeten Geräte, die korrespondierenden Schaltaktionen und mögliche Wertebereiche. So lässt sich eine Lampe beispielsweise auf einen Wert zwischen 0% und 100% einschalten. Weiterhin wird in der Ontologie das Sprachmodell gespeichert, was die automatisierte Generierung einer Grammatik erlaubt. So ist die Integration neuer Synonyme an zentraler Stelle möglich. Großer Wert wird auf die Natürlichsprachlichkeit gelegt. So soll der Benutzer nicht bestimmte Befehle auswendig lernen müssen, sondern hat die freie Wahl in welcher Form er einen Befehl formuliert.

Nach der Erkennung des vom Nutzer geäußerten Befehls, werden die für die weitere Bearbeitung relevanten Informationen aus der Aussage extrahiert und in einen Frame überführt. Aus diesem Frame wird eine Anfrage an die Ontologie generiert. Je nach Eindeutigkeit des Ergebnisses wird entweder direkt ein Schaltbefehl für die Hardware erzeugt oder eine Nachfrage generiert und synthetisiert.

4.1 Architektur

Das System besteht aus vier Modulen: Spracherkennung, Sprachverstehen, Hardwaresteuerung und Sprachsynthese, wie in Abbildung 1 zu sehen ist. Nachdem die Sprache erkannt wurde, wird das Ergebnis zum *Natural Language Understanding Modul* weitergereicht, welches die natürliche Sprache in eine abstrakte, maschinenverarbeitbare Form überführt. Hierzu kommt das Frame-Konzept zum Einsatz. Jeder Dialogschritt wird durch eine Framestruktur repräsentiert, die Schlüsselinformationen wie *Gerät* und *Aktion* zusammenfasst. Diese Frames werden in einer Historie gespeichert, um Referenzen zwischen aufeinanderfolgenden Dialogschritten auflösen zu können.

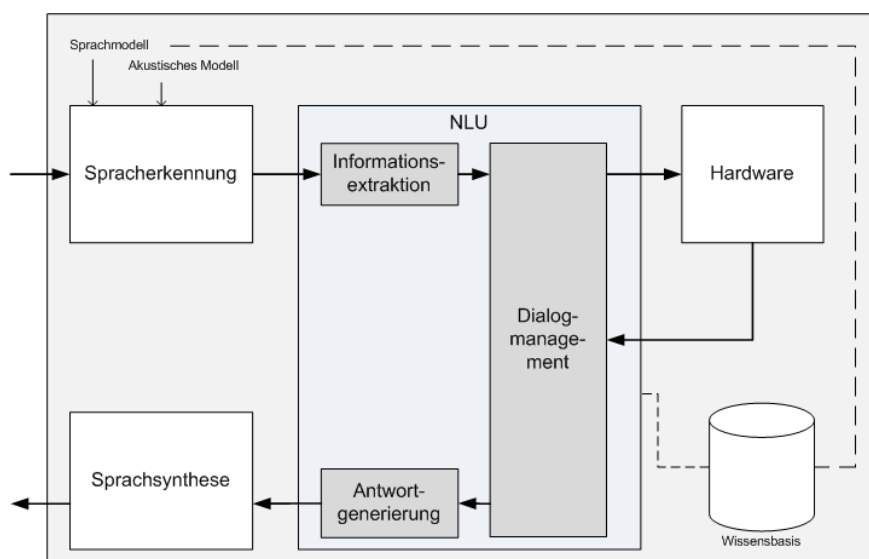


Abbildung 1: Architektur

Das Dialogmodul trifft die Entscheidung, ob es sich bei einer Nutzeräußerung um eine Antwort, einen Befehl, der sich auf einen anderen Befehl bezieht oder um einen komplett neuen Befehl handelt. Dies wird durch den lokalen Dialogkontext realisiert. Nachdem der Benutzer einen Befehl gegeben hat und die relevanten Informationen aus diesem extrahiert worden sind, wird der entsprechende Frame im Dialogkontext gespeichert. Anschließend wird eine Suchanfrage generiert und an die Wissensbasis, die das Sprach- und das Domänenmodell enthält, gesendet. Das Domänenmodell beschreibt das Inventar, Beziehungen zwischen den Geräten und mögliche Befehle. Es erlaubt die dynamische Generierung einer Grammatik mit Hilfe des Sprachmodells. Wenn das Ergebnis eindeutig ist, wird das entsprechende Gerät durch das Hardwaremodul geschaltet. Anderenfalls wird eine Nachfrage erzeugt und synthetisiert, um dem User eine Spezifizierung

seines Befehls zu ermöglichen. Die Antwort des Nutzers wird nun wieder in einen Frame überführt. Dieser Frame enthält ausschließlich die Ergänzung bzw. Spezifizierung, nicht jedoch die Informationen, die der Nutzer im letzten Satz gegeben hat. Um nun zu verstehen, was der Nutzer beabsichtigt, wird der Ergebnisframe aus dem aktuellen Frame und dem Kontextframe inferiert. Wenn der Befehl nun eindeutig ist, wird der Kontext zurückgesetzt und der geschlussfolgerte Ergebnisframe wird der History für eine spätere Anaphernresolution hinzugefügt. Anschließend wird ein Hardwarebefehl generiert und durch das Hardwaremodul umgesetzt. Wenn der Befehl immer noch nicht eindeutig sein sollte, erfolgt eine weitere Nachfrage. Falls eine Nutzeraussage keinen Sinn ergeben sollte, schlägt der Plausibilitätstest fehl und eine Fehlermeldung wird ausgegeben.

Da verschiedene Hardware-Implementierungen (z.B. AMX oder EIB) existieren, ist das Hardwaremodul als Plugin realisiert. Dies erlaubt die einfache Anpassung an die hardwaretechnische Ausstattung des Nutzers. Weiterhin ist eine grafische Simulation integriert, die es ermöglicht das Sprachverarbeitungssystem auch ohne reale Hardware präsentieren zu können.

4.2 Ontologie

Um natürlichsprachliche Befehle interpretieren zu können, benötigt die Applikation eine umfassende Datenbasis, die Informationen über die Domäne enthält. Hierzu gehören neben den verwendeten Geräten und Schaltbefehlen, vor allem Informationen über die semantischen Beziehungen aller Objekte. Auch das Vokabular wird an dieser Stelle gespeichert.

Ontologien sind dazu in der Lage, Konzepte mit der Hilfe von Assoziationen miteinander zu verbinden und stellen daher einen bequemen Weg dar, Wissen für natürlichsprachliche Anwendungen zu modellieren. Ursprünglich ist die Ontologie ein Begriff aus der Philosophie, der die Lehre des Seins darstellt. In der Informatik versteht man unter einer Ontologie ein formales System aus Konzepten und Relationen. Im Gegensatz zu Datenbanken, stellen Ontologien eine Möglichkeit zur formalen Beschreibung semantischer Beziehungen zur Verfügung. Durch diese Regeln ist es möglich, Informationen aus bereits vorhandenen Daten zu schlussfolgern und das Modell hierdurch anzureichern. Des Weiteren können Inkonsistenzen in der Datenbasis verhindert werden.

Es existieren zahlreiche Sprachen und Standards zur maschinenverarbeitbaren Beschreibung von Ontologien. Die bekanntesten sind das *Resource Description Framework* (RDF) woraus sich die *Web Ontology Language* (OWL) abgeleitet hat, F-Logic sowie die *Semantic Web Rule Language* (SWRL). Die gebräuchlichste Anfragesprache ist *SPARQL*.

Die vorgestellte OWL-Ontologie umfasst ein begrenztes Wissen, um einen automatisierten Raum zu steuern, wie im Ausschnitt in Abbildung 2 zu sehen ist. Das bedeutet, die Ontologie muss das Schalten von verschiedenen Geräten (z.B. Lampen, Jalousien, Heizung, Unterhaltungselektronik wie DVD-Player oder PC) unterstützen. Jedes Gerät ist eine Instanz des Konzeptes *Objekt*, die mit mehreren Schaltaktionen in Beziehung gesetzt werden kann. Dabei werden drei verschiedene Schaltkategorien unterschieden:

- binär
- prozentual
- media

Alle diese Kategorien sind Subklassen des Konzeptes *Aktion*. Jedes Gerät gehört zu einer oder mehreren Schaltkategorien. Binäre Geräte haben genau einen Zustand: an oder aus. Prozentuale Geräte können einen Wert zwischen 0 und 100 annehmen wohingegen Mediengeräte spezielle Aktionen wie „vor“ oder „zurück“ besitzen. Jede Schaltklasse hat weitere Subklassen die die eigentlichen Schaltbefehle wie „an“, „aus“, „höher“, „tiefer“ oder „nächster Titel“ bestimmen.

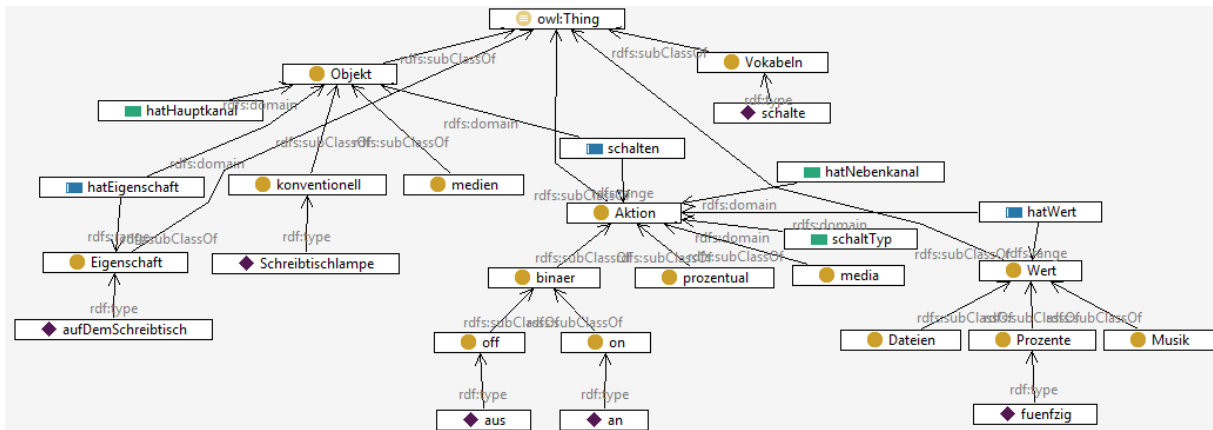


Abbildung 2: Ausschnitt der Ontologie

Des Weiteren hält die Ontologie das gesamte Vokabular der Anwendung bereit. Die Grammatik für den Spracherkennner wird dynamisch beim Systemstart generiert. Daher besitzen alle Instanzen das datatype-property *rdfs:label*, das den natürlichsprachlichen Ausdruck enthält. Über diesen Mechanismus können auch Synonyme definiert werden.

4.3 Anaphernresolution

In der natürlichsprachlichen Kommunikation werden Äußerungen erzeugt, die semantisch interpretiert werden müssen. Dies geschieht durch ein Natural Language Understanding Modul, das dazu in der Lage ist, eine Äußerung in eine abstrakte, maschinenverarbeitbare Form zu überführen. Ein spezieller Anwendungsfall ist die Auflösung von Referenzen (z.B. „sie“ oder „es“) aufeinanderfolgender Dialogschritte.

Diese Auflösung erfolgt dabei als Präprozessorstufe während ein Satz in einen Frame überführt wird. Gängige Ansätze führen hierzu eine syntaktische Analyse durch, um Informationen über die Beziehung zwischen den Wörtern ermitteln zu können. Dies erfordert detailliertes grammatikalisches Wissen in Form einer lexikalischen Datenbank. Mit der Hilfe eines POS-Taggers können Beziehungen erkannt und Referenzen aufgelöst werden. Dieser Ansatz ist sehr komplex, benötigt viel Rechenleistung und erfordert ein hohes Maß an lexikalischen Informationen.

In Befehlsszenarien gibt es drei große Vorteile, die die Nutzung eines heuristischen Ansatzes ermöglichen:

- Die referenzierte Subjekt-Kategorie ist a priori bekannt. Es muss sich um ein schaltbares Gerät handeln, das im Domänenmodell vorhanden ist.
- In Prosatexten existieren Sätze mit mehreren Referenzen wie z.B. „Sie gab ihm das Buch“. In Steuerungsszenarien ist meist nur eine Referenz pro Satz vorhanden, wie z.B. „Schalte sie wieder aus“.
- Gerät und Aktion müssen kompatibel sein: Eine Heizung kann nicht mit der Aktion *dunkler* in Bezug gesetzt werden.

Jeder Frame wird nun für eine bestimmte Zeit oder abhängig von der Größe eines FIFO-Speichers in einer Historie gespeichert. Eine Zeitfunktion könnte dabei das menschliche Kurzzeitgedächtnis repräsentieren und den zuletzt erfolgten Einträgen eine höhere Wahrscheinlichkeit zuordnen. In dieser Implementierung wird eine statische Historiengröße mit einer Länge von vier Einträgen genutzt, da die Nutzer sich meist nicht auf mehr als vier Geräte in der Vergangenheit beziehen. Wenn der Präprozessor eine Anapher erkennt, wird die Aktion aus dem Befehl extrahiert und geprüft, ob ein Gerät aus der Historie mit dieser Aktion kompatibel ist und sich noch nicht im

gewünschten Zustand befindet. Wenn dieser Vorgang erfolgreich ist, kann der Frame um das entsprechende Gerät ergänzt werden. Wenn mehr als eine Möglichkeit besteht, weil mehrere Historieneinträge passen, wird der aktuellste Eintrag genutzt.

5 Ausblick

Der Anspruch eines intuitiv bedienbaren Systems erfordert eine Erweiterung des Systems und die Integration relevanter Kontextinformationen in den Entscheidungsprozess. Hierbei kann zwischen statischem Umgebungs- und dynamischem Situationskontext unterschieden werden.

Als statischer Kontext werden Eigenschaften von Geräten und deren Wechselwirkungen aufgefasst. So werden in der zugrunde liegenden Ontologie Beziehungen definiert, die Geräte mit konträrer Wirkung identifizieren. Weiterhin können hier durch den Nutzer Prioritäten oder aber abstrahierte Befehle festgelegt werden, welche verschiedene Schaltvorgänge zusammenfassen.

Zum dynamischen Kontext können alle Parameter gezählt werden, die sich direkt auf die aktuelle Situation beziehen. Hierbei spielen die Lichtverhältnisse, Temperatur, die Bewegung des Nutzers im Raum, vergangene Systeminteraktionen oder ähnliche Faktoren eine entscheidende Rolle. Viele dieser Informationen können in Form von Sensorwerten in die Entscheidung einbezogen werden. Aber auch der Verlauf des Dialoges kann wichtige Hinweise für die selbstständige Ergänzung unvollständiger Frames liefern. So ist es bei häufig wiederkehrenden Missverständnissen denkbar, aus der Dialoghistorie Vorschläge für wahrscheinliche Nutzerziele anzubieten.

In Zusammenarbeit mit der Universität Rostock wird an der Weiterentwicklung der multimedialen Fähigkeiten des Systems gearbeitet. Im Vordergrund stehen hier eine Intentionsanalyse und das Zusammenspiel von grafischen Eingaben, Nutzerverhalten und Sprache. So soll ermöglicht werden, dass das System situationsabhängig verschiedene Kommunikationskanäle nutzt und die Module sich jeweils im Bedarfsfall mit Information versorgen und unvollständige Schaltbefehle ergänzen.

Literatur

- [1] Ole Grelck: „Elektronisches Ohr“ – Entwicklung einer kombinierten Hard- und Softwarelösung zur Kopplung eines Sprachsteuerungssystems mit einem Mediensteuerungssystem. Hochschule Wismar – University of Technology, Business and Design, 2006. Diplomarbeit.
- [2] Nicole Fernholz: „Elektronisches Ohr“ – Entwicklung und Realisierung eines Meta-Sprachkonzeptes zur Steuerung von Medienanlagen. Hochschule Wismar – University of Technology, Business and Design, 2006. Diplomarbeit.
- [3] Steigner, Jochen et al.: ODP. Ontology-based Dialogue Platform.
http://www.semvox.de/whitepapers/odp_whitepaper-de.pdf,
- [4] Dr. Saueressig, Peter: CeBIT 2009: Sprachsteuerung für Mediengeräte - einfach, schnell, für alle!
http://www.em1-development.de/deutsch/presse/archiv.php?we_objectID=351&pid=0WEhjdAfgkh,
- [5] Stein, M.: Natürlichsprachliche kontextgebundene Kommunikation in der Gebäudeautomation. Technical Report, University of Technology, Business and Design, Institut für Multimedialechnik, 2007
- [6] BASIS Computer- & Systemintegration GmbH: Projekt: „Elektronisches Ohr“.
<http://dz.basis-wismar.de/downloads/Projektinfo.pdf>,
- [7] Markus Berg: Entwicklung eines Architekturkonzeptes für ein natürlichsprachliches, dialogbasiertes Raumsteuerungssystem. Hochschule Wismar – University of Technology, Business and Design, 2009. Master Thesis.