

# SEMANTISCHE ANWENDUNGSMODELLIERUNG UND NUTZERMODELLE: ANFORDERUNGEN AN EINEN SPRACHLICH ZU BEDIENENDEN PERSÖNLICHEN ASSISTENTEN IM DYNAMISCHEN SYSTEMUMFELD

*Sven Reichel<sup>1</sup>, André Berton<sup>1</sup>, Ute Ehrlich<sup>1</sup> und Michael Weber<sup>2</sup>*

*<sup>1</sup> Speech Dialogue Systems, Daimler AG, Ulm*

*<sup>2</sup> Institut für Medieninformatik, Universität Ulm*

*sven.reichel@daimler.com*

**Kurzfassung:** Infotainmentsysteme im Auto bieten dem Fahrer viele Funktionen, die inzwischen weit über die boardeigenen Anwendungen hinaus gehen. Internetzugang ist vorhanden und bringt, neben vielen Anwendungen, Anforderungen an eine konsistente Sprachbedienung des Infotainmentsystems. Dieser Beitrag formuliert Anforderungen an ein zukünftiges Infotainmentsystem anhand von fünf Anwendungsfällen und motiviert als Abstraktionsschicht zwischen Mensch und Anwendungen einen persönlichen Assistenten. Dieser beinhaltet Weltwissen, Interaktionshistorie, Nutzermodell und Situationsverständnis um eine aufgabenbasierte Interaktion zu ermöglichen.

## 1 Einleitung

Durch die zunehmende Funktionsvielfalt von Softwaresystemen nehmen die Anforderungen an eine einfach und intuitiv zu bedienende Mensch-Maschine-Schnittstelle zu. Ein prominentes Beispiel ist hierbei das Smartphone, welches durch optionale Anwendungen (Apps genannt) für die verschiedensten Aufgaben genutzt werden kann. Somit gleicht auf der Softwareseite kein Gerät dem anderen (im Folgenden wird dies als dynamisches System bezeichnet). Dieser Trend zeichnet sich auch in Infotainment-Anwendungen im Auto ab (zum Beispiel R-LINK<sup>1</sup> von Renault oder Mercedes-Benz Apps<sup>2</sup>). Im Gegensatz zum Smartphonenuutzer benötigt der Fahrer allerdings seine visuelle Aufmerksamkeit auf der Straße und die Hände am Lenkrad. Dies hat zur Folge, dass die sprachliche Modalität in Infotainment-Anwendungen einen sehr hohen Stellenwert hat, da sie im Vergleich zur haptischen Bedienung die Fahrerablenkung senkt [11]. Für den fest definierten Funktionsumfang von heutigen Infotainment-Anwendungen ist die Sprachbedienung im Auto daher etabliert. Doch durch diverse Verbindungen zu externen Geräten, dem Internet, App-Stores und anderen Fahrzeugen wird die Infotainment-Anwendung immer mehr zu einem dynamischen System, welches vollständig und einheitlich per Sprache bedient werden sollte. Dies geht weit über die Funktionalität aktueller Systeme hinaus und setzt neue Anforderungen.

Selbst bei aktuellen automobilen HMIs fällt es den Nutzern schwer, alle verfügbaren Funktionen mit ihren Sprachkommandos zu lernen. Im Optimalfall würde das HMI sich wie ein perfekter Beifahrer verhalten und natürlich mit dem Fahrer interagieren. In der Mensch-Mensch-Kommunikation können komplizierte Sachverhalte sehr einfach über natürliche Sprache ausgedrückt werden. Hierbei beachten Menschen ihren Kommunikationspartner, die Situation und Erfahrungswerte [10]. Damit lässt sich auch aus einer kurzen sprachlichen Äußerung oftmals eine komplexe Anweisung ableiten. Weiterhin erschweren abstrakte Begriffe und Synonyme

---

<sup>1</sup><http://bit.ly/147qgJO>

<sup>2</sup>[http://apps.mercedes-benz.com/apps/de\\_DE](http://apps.mercedes-benz.com/apps/de_DE)

die richtige Interpretation einer Nutzeräußerung. Menschen interagieren nicht anwendungsorientiert, sondern aufgabenbasiert. Beispielsweise würde ein Nutzer eher „öffne das Fenster“ als „betätige den Schalter links neben dir“ sagen. In diesem Beispiel wäre die Aufgabe das Fenster zu öffnen. Dies ist über die Bedienschnittstelle Schalter möglich, welche - im übertragenden Sinne - mit der zugehörigen Schaltungselektronik, die Anwendung darstellt.

Dynamische Systeme sind modular aufgebaut und können zur Laufzeit vom Nutzer verändert werden. Sie können z.B. aus internen Anwendungen bestehen und vom Anwender über Apps erweitert werden. Dies hat zur Folge, dass ein System nicht von einer einzigen Entwicklergruppe realisiert wird, sondern von vielen verschiedenen. Eine Konsistenz zwischen Apps ist somit nur schwierig zu erreichen. Wenn der Nutzer im Kontext der Anwendung interagiert, ist dies nicht weiter schlimm, aber wenn er aufgabenbasiert mit mehreren interagiert, ist ein einheitliches Bedienkonzept von entscheidender Bedeutung. Zum Beispiel muss ein mentales Modell von einer internen Musikanwendung auf eine Internetradio-App übertragbar sein. Dafür ist es notwendig, die Aufgaben der Apps zu identifizieren und semantisch gleiche Aufgaben mit derselben Interaktionsform anzusprechen.

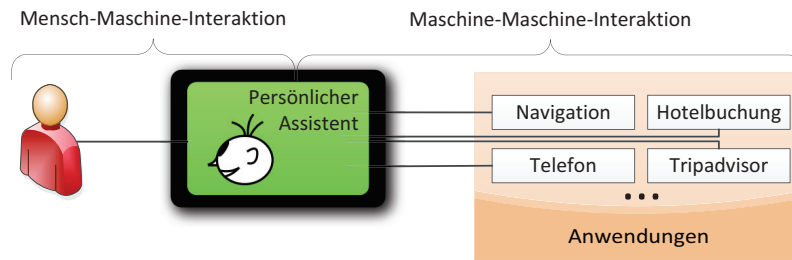
Ein dynamisches System, das ein einheitliches, aufgabenorientiertes Bedienkonzept ähnlich einem perfekten Beifahrer anbieten will, muss diverse Anforderungen erfüllen. Der zweite Abschnitt motiviert einen persönlichen Assistenten (PA) als einzige und einheitliche Benutzerschnittstelle. Im anschließenden Kapitel zeigen wir fünf Anwendungsfälle in der automobilen Domäne, mit welchen wir die abstrakte Softwarearchitektur eines PAs konkretisieren. Im fünften Abschnitt werden Anforderungen aus den Anwendungsfällen und der abstrakten Architektur abgeleitet. In Abschnitt 6 setzen wir bestehende Arbeiten in Relation zu den Anforderungen und ziehen anschließend ein Fazit, in welchen Bereichen zusätzlicher Forschungsbedarf besteht.

## **2 Das System als Kommunikationspartner**

Strategy Analytics prophezeit App Stores im Infotainmentsystem des Autos nur einen zeitlich begrenzten Hype [20]. Autohersteller sollten den Fokus auf die Funktionen der Apps legen und deren Anbindung über bekannte Benutzerschnittstellen maskieren. Dies motiviert unseren aufgabenorientierten Ansatz mit einem perfekten Beifahrer insofern, dass wir eine zentrale Interaktionskomponente zwischen internen Anwendungen, Apps und Nutzer einführen (siehe Abbildung 1). Diese Komponente, als persönlicher Assistent (PA) bezeichnet, bietet einerseits eine Kommunikation mit dem Nutzer (Mensch-Maschine-Schnittstelle) und andererseits Verbindungen zu internen Anwendungen sowie Apps (Maschine-Maschine-Schnittstelle). Im Folgenden wird nur auf den semantischen Datenaustausch zwischen PA und internen Anwendungen sowie Apps eingegangen und technische Belange ignoriert. Aus diesem Grund ist eine Unterscheidung zwischen internen Anwendungen und dynamischen Apps nicht notwendig. Der PA vermittelt somit zwischen Nutzer und dynamischem System. Er muss zum einen die Intention des Nutzers verstehen und zum anderen diese Intention auf die möglichen Funktionen, die das dynamische System anbietet, abbilden. Im dynamischen System sind die verfügbaren Funktionen nicht statisch, wodurch der PA auch als Service-Provider die Semantik hinter den verfügbaren Funktionen verstehen muss. Im Folgenden werden die notwendigen Anforderungen an einen PA anhand verschiedener Anwendungsfälle im automobilen Umfeld definiert.

## **3 Anwendungsfälle und deren Klassifikation**

In der automobilen Domäne interagiert der Fahrer mit dem Infotainmentsystem, um ein bestimmtes Ziel zu erreichen. Dies kann zum Beispiel der Wechsel des Radiosenders oder das Verfassen einer Nachricht an einen Bekannten sein. Dialoge zwischen Fahrer und System sind



**Abbildung 1** - Abstrakte Architektur eines Persönlichen Assistenten (PA)

somit von einem Ziel geprägt und meist über wenige Interaktionen zu erreichen. Das heißt, der Nutzer stellt eine bestimmte Aufgabe an das System - wie zum Beispiel Informationen zu beschaffen oder eine Aktion auszuführen - und das System reagiert dementsprechend. Je nach Aufgabe sind oftmals weitere Faktoren (auch Parameter genannt) notwendig, wie zum Beispiel der Name des Radiosenders oder der Adressat der Nachricht. Diese können entweder direkt aus der Nutzeräußerung extrahiert werden oder müssen im Zweifelsfall aus weiteren Quellen bezogen, beziehungsweise beim Nutzer nachgefragt werden. Interaktionen können somit abstrakt als Aufgaben, die der Nutzer an das System stellt, und zugehörige Parameter definiert werden. In diesem Abschnitt klassifizieren wir verschiedene Parameter anhand von Beispielaufgaben in der automobilen Domäne. Die Aufgaben sind jeweils unterstrichen und die Parameter markiert dargestellt.

### 3.1 Anwendungsfall 1: Weltwissen

Menschen haben ein je nach Erfahrung und Bildung unterschiedliches Allgemeinwissen. Dieses Wissen über Dinge und Sachverhalte in der Welt versteht man unter dem Begriff Weltwissen<sup>3</sup>. In der Interaktion mit einer Maschine kann ein Nutzer auch dieses Wissen verwenden. Somit muss die jeweilige Maschine oder Anwendung auch Parameter verstehen, die außerhalb der Anwendungsdomäne liegen. Bei einer Anweisung zum Buchen eines Hotels könnte der Nutzer zum Beispiel folgende Äußerung verwenden:

*Buche ein Hotel von Karfreitag bis Ostermontag*

Wenn nun die Anwendung lediglich ein Datum erwartet und keinerlei Informationen über Ostern enthält, kann die Nutzeranfrage nicht bearbeitet werden. Somit muss eine Übersetzung des Parameters in die Anwendungsdomäne stattfinden.

### 3.2 Anwendungsfall 2: Nutzermodell

Neben Fakten aus dem Weltwissen besitzt ein Mensch auch persönliches Wissen. Dies ist personengebunden und kann einerseits nur dem Individuum bekannt sein, oder andererseits auch Gesprächspartnern zugänglich gemacht werden. Ein Gesprächspartner in der Mensch-Mensch-Kommunikation hat somit, je nach Vorwissen, ein Abbild des persönlichen Wissens seines Kommunikationspartners. Bei einem klassischen Softwaresystem wird der Nutzer hierauf weniger zurückgreifen. Im Falle eines PA wird es allerdings zunehmend wichtiger solche Referenzen auflösen zu können. Die Datenbasis für Wissen über einen Nutzer nennen wir Nutzermodell. In einer Telefonanwendung könnte eine Nutzeräußerung zum Beispiel folgendermaßen lauten:

*Rufe meinen Onkel an*

<sup>3</sup><http://kitt.cl.uzh.ch/compli/glossar/Weltwissen.html>

Die Funktion *Anrufen* einer Telefonanwendung erwartet normalerweise als Parameter eine Nummer, die anzurufen ist, oder einen Namen aus dem Adressbuch. In diesem Beispiel muss eine Übersetzung des Parameters aus dem Nutzermodell in die Anwendungsdomäne erfolgen.

### 3.3 Anwendungsfall 3: Interaktionshistorie

Erfahrungen und Erlebnisse prägen Menschen sehr stark und spielen in der menschlichen Kommunikation eine entscheidende Rolle. Oftmals werden keine expliziten Parameter genannt, sondern es erfolgt eine implizite Nennung eines vorhergegangenen Ereignisses. Unter einem Ereignis können einerseits verschiedene Parameter, andererseits aber auch komplette Dialogabläufe zusammengefasst werden. Gerade in der automobilen Domäne sind möglichst kurze Interaktionen wichtig, um die Fahrerablenkung so gering wie möglich zu halten. In einer Hotelbuchungsanwendung könnte eine Nutzeräußerung beispielsweise folgendermaßen sein:

*Buche das Hotel vom letzten Jahr*

Die Aufgabe *Buche das Hotel* erfordert vom Nutzer die Eingabe vieler verschiedener Parameter wie beispielsweise Ort, Reisedatum, Hotelname und Zimmertyp. Liegt diese Information allerdings in einem gespeicherten Ereignis bereits vor, muss der Nutzer nur die Unterschiede kenntlich machen, was im genannten Beispiel lediglich das Reisedatum wäre.

### 3.4 Anwendungsfall 4: aktuelle Situation

Ein Mensch nimmt seine Umwelt über verschiedene Sinne wahr und nutzt sie aktiv in der Interaktion. Somit wird auch in der gesprochenen Sprache auf Dinge referenziert, die der Nutzer visuell aufnimmt. Dies impliziert, dass ein System auch Kenntnis von der Umgebung haben muss, um Nutzeräußerungen zu verstehen. Zum Beispiel könnte eine Äußerung in einer Kartenanwendung folgendermaßen lauten:

*Was ist das für ein Gebäude vor mir?*

Die Anwendung kann diese Frage nur beantworten, wenn sie Position und Blickrichtung des Nutzers kennt und versteht. Ist sie darauf ausgelegt, Information über ein Gebäude unter einer bestimmten Adresse zu liefern, muss eine Übersetzung des Parameters *vor mir* erfolgen.

### 3.5 Anwendungsfall 5: verschiedene Anwendungsdomänen

Neben globalen Informationen wie Weltwissen, Nutzermodellen, Ereignissen und Situationen ist auch der Parametertausch zwischen Anwendungen wichtig. Bei der aufgabenorientierten Interaktion abstrahieren wir von den Anwendungen hin zu Aufgaben. Somit ist dem Nutzer unter Umständen nicht bekannt, mit welcher Anwendung er aktuell interagiert und er kann zwischen verschiedenen Anwendungen unbewusst wechseln. Für eine reibungslose Interaktion müssen Parameter somit zwischen Anwendungen ausgetauscht und richtig interpretiert werden. Ein häufiges Beispiel ist, während eines Hotelbuchungsprozesses Informationen von einem unabhängigen Bewertungsportal einzuholen:

System: Sie haben das Hilton Hotel in Berlin ausgewählt, wollen Sie das buchen?  
*Was für Bewertungen hat Tripadvisor zu diesem Hotel?*

Da die Auswahl des Hotels mit einer Hotelanwendung stattfand, hat die Tripadvisor Anwendung keine Kenntnis, zu welchem Hotel der Nutzer eine Bewertung wünscht. Um eine erneute Eingabe durch den Nutzer zu vermeiden, sollte die Hotelanwendung diese Information bereitstellen.

## 4 Abstrakte Architektur

Im vorherigen Abschnitt haben wir verschiedene Anwendungsfälle gezeigt und Parameter für Funktionen klassifiziert. Aktuelle Apps auf Smartphones setzen bereits viele Funktionen inklusive Parametertausch um, wobei dies hauptsächlich über fest definierte APIs oder Integration des Wissens in den Anwendungen geschieht. Dies resultiert in einer unübersichtlichen Kommunikation der Anwendungen untereinander und redundanter Datenhaltung. Zusätzlich besteht in einem dynamischen System Unsicherheit über die aktuell verfügbaren Anwendungen, somit können feste Verbindungen im Zweifelsfall nicht aufgebaut werden. Im Folgenden betrachten wir die Verbindungen zwischen verschiedenen Modulen lediglich auf semantischer Ebene und unabhängig von technischen Aspekten.

Wie in Abschnitt 2 beschrieben, steht der PA als zentraler Interaktionspartner zwischen Nutzer und Anwendung. Für eine konsistente Interaktion mit verschiedenen Anwendungen ist es wichtig, dass Referenzen des Nutzers in die Wissensbasen jeweils auf identische Parameter abgebildet werden. Aus diesem Grund sollte die Verwaltung und der Zugriff auf die Wissensbasen im PA stattfinden (siehe Abbildung 2).

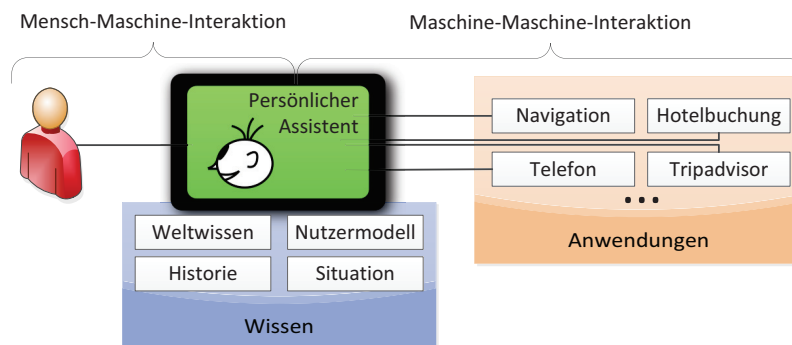


Abbildung 2 - Abstrakte Architektur eines persönlichen Assistenten mit Wissensbasen

## 5 Anforderungen

Um Nutzeräußerungen, wie in Abschnitt 3 beschrieben, verstehen zu können, muss der PA gewisse Anforderungen erfüllen. In diesem Abschnitt stellen wir die wichtigsten vor. Die Betrachtung erfolgt auf abstrakter Ebene, und konkrete technische Umsetzungen sind Teil zukünftiger Arbeiten. In erster Linie wurden folgende funktionale Anforderungen identifiziert:

- Der PA muss Aufgaben vom Nutzer verstehen und Anwendungsfunktionen zuordnen.
- Der PA muss Parameter vom Nutzer abhängig vom Weltwissen, Nutzermodell, Situation und Historie verstehen und in die Anwendungsdomäne konvertieren.
- Der PA muss Wissensbasen über Weltwissen, Nutzermodell, Interaktionshistorie und Situation verwalten.
- Der PA muss Anwendungen verwalten und einen Datenaustausch ermöglichen.
- Der PA muss die Bedeutung hinter Anwendungsfunktionen verstehen.
- Zwischen PA und Nutzer sollte eine natürliche Dialogführung stattfinden.
- Der PA sollte Hilfestellung über die aktuell verfügbaren Funktionen bieten.
- Der PA sollte ausgeführte Aktionen erklären können.



Zentrale Punkte sind folglich das Verstehen des Nutzers und das Abbilden der Nutzerintention auf die verfügbaren Funktionen der Anwendungen. Für eine erfolgreiche Abbildung muss der PA auch die Bedeutung hinter den Funktionen verstehen, die eine jede Anwendung zur Verfügung stellt. Als Analogie betrachten wir zuerst die Verarbeitung von Sprache beim Menschen, wie von Hilberg [4] beschrieben. Die Bedeutung eines Wortes wird dabei als Symbol betrachtet, welches im Inneren des Gehirns einen Repräsentanten (z.B. ein Neuron) hat. Nach Hilberg ist nun die Position eines Neurons und seine Verbindungen entscheidend und nicht die Bedeutung an sich. Übertragen auf ein Softwaresystem motiviert dies den Einsatz von semantischen Netzen, in welchen Objekte und deren Verbindungen repräsentiert, sowie Interferenzen gezogen werden können [17].

Semantische Wissens- und Anwendungsmodelle erlauben zwar einerseits vielfältige Modellierungsmöglichkeiten und das Ziehen komplexer Schlussfolgerungen, aber bereiten andererseits auch Probleme. Ein entscheidender Punkt eines semantischen Netzes ist die Eindeutigkeit der Objekte [17]. Bei einem PA handelt es sich um ein dynamisches System (open multi-agent system), welches auf eine Vielzahl verschiedener Wissensbasen zurückgreifen muss. Somit muss ein Konsens zwischen den verschiedenen Systemmodulen über gleiche Objekte stattfinden [14, 12]. Ein klassisches Problem von Mensch-Maschine-Schnittstellen ist, dass zwischen System- und Nutzerontologie eine Diskrepanz bestehen kann: „A person who wants to find information is unlikely to have the same working ontology as the system uses for its classification“[15]. Durch den Einsatz eines semantischen Nutzermodells, in dem die Nutzerontologie repräsentiert ist, lässt sich diese Diskrepanz (durch den Abgleich der Wissensbasen untereinander) minimieren.

## 6 Bestehende wissenschaftliche Arbeiten

Durch die Entwicklung auf dem Smartphone-Markt erregen persönliche Assistenten derzeit Aufsehen. Viele verschiedenen Anbieter konkurrieren um den perfekten PA (zum Beispiel Apple mit Siri<sup>4</sup> oder Nuance mit Nina<sup>5</sup>). Diese Assistenten sind fest programmiert und können nur mit definierten Apps interagieren. Im wissenschaftlichen Kontext sind PAs allerdings schon lange ein Forschungsthema, wie zum Beispiel im SmartKom Projekt [18]. Der PA in SmartKom konnte mit mehreren Anwendungen multimodal interagieren und besaß eine globale Ontologie. In der automobilen Domäne wurden auch Assistenten mit natürlichsprachlicher Interaktion untersucht [3], wobei die semantische Interpretation lediglich auf Nutzerseite stattfand. Ebenfalls auf Nutzerseite passen Czibula et al. [2] die Aufgabenreihenfolge mit neuronalen Netzen an und Chen et al. [1] erleichtern wiederholt auftretende Aufgaben mit einem case-based reasoning Ansatz. Kravari et al. [9] hingegen entwickelten ein Multi-Agenten-System um komplexe Tasks über verschiedene Agenten hinweg zu kombinieren. Der Nutzer wird allerdings nur als Rolle vordefiniert, und Kontextinformationen werden nicht beachtet.

Laut Johnson et al. [6] geht die Entwicklung von Benutzerschnittstellen vom Nutzer aus: „start by designing how people think about their tasks: the concepts users will think about, and how those concepts fit together into a structure“. In der weiteren Entwicklungsphase werden die Konzepte auf Funktionen abgebildet, die meist für Menschen verständlich beschrieben sind (oder zumindest verständlich für den Entwickler). Dies kann mit einer Spezifikation in Form eines Lastenhefts sein oder aber in der modellbasierten Entwicklung mit diversen Modellen, wie zum Beispiel ConcurTaskTrees [16]. Bei der späteren Umsetzung fließt diese Semantik normalerweise nicht in die Anwendung selbst. Dabei bieten semantisch modellierte Anwendungen großes Potential in der Mensch-Maschine-Interaktion, wie Karnagel et al. [7] anhand einer sta-

---

<sup>4</sup><http://www.apple.com/ios/siri/>

<sup>5</sup><http://bit.ly/NX5RCJ>

tisch spezifizierten Home-Entertainment-Steuerung gezeigt haben. Weitreichendere Ideen zur semantischen Anwendungsmodellierung werden im SemanticWeb [15] und Semantic Web of Things [8] betrachtet, wobei diese Ontologien nicht nutzerspezifisch sind.

Laut Wendemuth und Biundo [21] sind kognitive technische Systeme, Companion Systeme, die sich individuell an den Nutzer anpassen, die Systeme der Zukunft. Im SFB/TRR 62 wird ein adaptives System entwickelt, welches den Nutzer und seine Umgebung in den Fokus stellt. Eine Verbindung zu verschiedenen Anwendungen, also ein dynamisches System, ist nicht Teil des Projekts. COMPANIONS (EU-Projekt FP 6<sup>th</sup>) [22] ist ein weiteres sprachbedienbares Companion System, das in erster Linie den Inhalt von eingehenden Dialogschritten extrahiert und in eine Wissensdatenbank einpflegt, um daraus neue Fakten zu generieren. Wie SFB/TRR62 ist dies auch kein dynamisches System.

Wie in diesem Abschnitt gezeigt, beschäftigen sich wenige Ansätze mit der semantischen Interpretation von individuellen Nutzereingaben und gleichzeitig aufgabenorientierter Interaktion mit Hilfe semantischer Anwendungsspezifikation in dynamischen Systemen. Zwei Arbeiten, die dies im Ansatz berücksichtigen, sind [19] und [5]. Das TADEUS Projekt [19] brachte modellgetriebene Anwendungsentwicklung mit aufgabenbasierter, nutzerorientierter Interaktion zusammen. Die Nutzer waren in diesem Projekt allerdings durch feste Rollen abstrahiert. Jimanez-Molina et al. [5] hingegen betrachten den Kontext des Nutzers in einer ubiquitären Umgebung und entwickelten einen Selektionsalgorithmus für semantisch beschriebene Aufgaben. Dieser Ansatz deckt zum großen Teil die Anforderungen an einen PA ab, solange die Anwendungen nicht dynamisch definiert sind.

## 7 Fazit

Die zunehmende Funktionsvielfalt in dynamischen Infotainmentsystemen im Auto stellt diverse Anforderungen an ein Sprachdialogsystem, um eine konsistente Bedienung zu gewährleisten. Wir stellten in diesem Paper einen PA als Schnittstelle zwischen Mensch und Anwendungen vor, um aufgabenbasierte Interaktion zu ermöglichen. Anhand von fünf Anwendungsfällen wurde demonstriert, dass globales Weltwissen, Interaktionshistorie, Nutzermodell und Situationsverständnis wichtig für die Abbildung von Nutzersemantik auf Anwendungssemantik sind. Anschließend definierten wir Anforderungen an einen PA. Diverse Arbeiten decken Teile dieser Anforderungen ab, allerdings betrachten wenige Ansätze die Abbildung der Semantik zwischen Nutzer- und Anwendungsdomäne bei aufgabenorientierter Interaktion. In zukünftiger Forschung betrachten wir semantische Anwendungsmodelle und wie, mit deren Hilfe, Nutzeräußerungen auf Anwendungsfunktionen abgebildet werden können.

Neben dem Verständnis einer Nutzeräußerung kann auch die Interaktion des PAs Richtung Mensch adaptiert werden. Laut McGuinness et al. [13] ist es für einen PA entscheidend, seine Aufgaben erklären zu können und angepasste Hilfe anzubieten. Wenn der PA seine Aufgaben versteht und Voraussetzungen für deren Ausführung kennt, könnte er diese für den Nutzer aufbereiten. Die Erklärung einer Telefonfunktion könnte zum Beispiel lauten: „Zum wählen brauche ich die Telefonnummer oder einen Namen aus Deinem Adressbuch“. In wie weit dies mit semantischen Anwendungsmodellen ermöglicht wird, ist Teil weiterer Forschung.

## Literatur

- [1] CHEN, K.-J. und J.-P. BARTHÈS: *MemoPA: Intelligent Personal Assistant Agents with a Case Memory Mechanism*. In: *Proceedings of the 3rd International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, S. 1357–1367, Berlin, Heidelberg, 2007. Springer.

- [2] CZIBULA, G., A.-M. GURAN, I. G. CZIBULA und G. S. COJOCAR: *IPA - An Intelligent Personal Assistant Agent for Task Performance Support*. In: *Intelligent Computer Communication and Processing*, S. 31–34, Cluj-Napoca, 2009.
- [3] DAUSEND, M. und U. EHRLICH: *A Prototype for Future Spoken Dialog Systems Using an Embodied Conversational Agent*. In: ANDRÉ, E., L. DYBKJÆR, W. MINKER, H. NEUMANN, R. PIERACCINI und M. WEBER (Hrsg.): *Perception in Multimodal Dialogue Systems*, S. 268–271. Springer, Berlin, Heidelberg, 2008.
- [4] HILBERG, W.: *Die rätselhafte Fähigkeit des Gehirns, Sprache ohne Hilfe von neuronalen Codes zu verarbeiten*. In: *Elektronische Sprachsignalverarbeitung*, Cottbus, 2012.
- [5] JIMENEZ-MOLINA, A., J.-S. KIM, H.-M. KOO, B.-S. KANG und I.-Y. KO: *A Semantically-Based Task Model and Selection Mechanism in Ubiquitous Computing Environments*. In: VELÁSQUEZ, J., S. RÍOS, R. HOWLETT und L. JAIN (Hrsg.): *Knowledge-Based and Intelligent Information and Engineering Systems*, S. 829–837. Springer, Berlin, Heidelberg, 2009.
- [6] JOHNSON, J. und A. HENDERSON: *Conceptual Models: Core to Good Design*. Morgan&Claypool, 2011.
- [7] KARNAGEL, K., R. RÖMER, S. ROGGE und J. LINDEMANN: *Semantische Modellierung von System-Benutzer-Interaktionen am Beispiel einer Home-Entertainment Steuerung*. In: *Elektronische Sprachsignalverarbeitung*, Cottbus, 2012.
- [8] KATASONOV, A., O. KAYKOVA, O. KHRIYENKO, S. NIKITIN und V. TERZIYAN: *Smart Semantic Middleware for the Internet of Things*. In: *Proceedings of the 5-th International Conference on Informatics in Control, Automation and Robotics*, S. 11–15, 2008.
- [9] KRAVARI, K., N. BASSILIADES und H. BOLEY: *Cross-Community Interoperation Between Knowledge-Based Multi-Agent Systems: A Study on EMERALD and Rule Responder*. *Expert Systems with Applications*, S. 9571–9587, 2012.
- [10] LITTLEJOHN, S. und K. FOSS: *Theories of Human Communication*. Wadsworth series in communication studies. Cengage Learning, 2008.
- [11] MACIEJ, J. und M. VOLLRATH: *Comparison of manual vs. speech-based interaction with in-vehicle information systems*. *Accident Analysis&Prevention*, S. 924–930, 2009.
- [12] MAZUEL, L. und N. SABOURET: *A Communication Protocol for Semantic Heterogeneity with Incomplete Ontology Alignment*. In: *Autonomous Agents and Multiagent Systems*, S. 1187–1188, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.
- [13] MCGUINNESS, D. L., A. GLASS, M. WOLVERTON und P. P. DA SILVA: *Explaining Task Processing in Cognitive Assistants That Learn*. In: *Florida Artificial Intelligence Research Society Conference*, Key West, 2007.
- [14] MORGE, M. und J.-C. ROUTIER: *Debating over heterogeneous descriptions*. *Applied Ontology*, 2:333–349, 2007.
- [15] PASSIN, T. B.: *Explorer's Guide to the Semantic Web*. Manning, Greenwich, 2004.
- [16] PATERNÒ, F.: *ConcurTaskTrees: An Engineered Approach to Model-based Design of Interactive Systems*. ISTI-CNR, Pisa, 2001.
- [17] REICHENBERGER, K.: *Kompendium semantische Netze*. X.media.press. Springer, Heidelberg, 2010.
- [18] REITHINGER, N., J. ALEXANDERSSON, T. BECKER, A. BLOCHER, R. ENGEL, M. LÖCKELT, J. MÜLLER, N. PFLEGER, P. POLLER, M. STREIT und V. TSCHERNOMAS: *SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications*. In: *Multimodal interfaces*, S. 101–108, New York, 2003.
- [19] STARY, C.: *TADEUS: Seamless Development of Task-Based and User-Oriented Interfaces*. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, S. 509–525, 2000.
- [20] STRATEGY ANALYTICS: *App Stor in the Car - It's Not 'App-ening*. <http://bit.ly/SI63MT>, Oktober 2012. [Online - accessed 09.01.2013].
- [21] WENDEMUTH, A. und S. BIUNDO: *A Companion Technology for Cognitive Technical Systems*. In: ESPOSITO, A., A. VINCIARELLI, R. HOFFMAN und V. C. MÜLLER (Hrsg.): *Proceedings of the EUCogII-SSPNET-COST2102 International Conference (2011)*, LNCS Proceedings on Cognitive Behavioural Systems, Dresden, 2012. Springer.
- [22] WILKS, Y., R. CATIZONE, S. WORGAN, A. DINGLI, R. MOORE, D. FIELD und W. CHENG: *A Prototype for a Conversational Companion for Reminiscing About Images*. *Computer Speech and Language*, S. 140–157, 2011.