VoiceXML - Technologie der Wahl für telefonbasierte Sprachdialogportale?

Roland Stuckardt*

Juli 2001

1 Ausgangspunkt und Zielsetzung

Im Zeitalter der ständig wachsenden Mobilitätsanforderungen kommt dem flexiblen, dezentralen Zugriff auf Datenbestände aller Art eine immer größere Bedeutung zu. Steht ein Zugang via Internet nicht zur Verfügung, so bietet sich als Alternative die Verwendung eines Mobiltelefons an. Auf der Grundlage des WAP-Protokolls können elementare grafische Zugriffsschnittstellen geschaffen werden; deren Möglichkeiten sind jedoch begrenzt:

- Im Vergleich zu stationären Computerterminals ist die Displaygröße i.d.R. gering; entsprchend aufwändig verläuft das Browsing.
- Die gegenwärtige Technologie verfügt über eine geringe Bandbreite.
- die Navigation über Tasten wird vom Benutzer als umständlich empfunden.
- Es gibt Einsatzkontexte, die eine tastaturbasierte Interaktion a priori ausschließen.

Als Alternative bieten sich *gesprochensprachige Schnittstellen* an, in denen der Benutzer einen *Mensch-Maschine-Dialog* mit einem telefonbasierten Sprachportal führt. Die Grundlage derartiger Anwendungen bietet Hardware- bzw. Software-Technologie zu

- Computer-Telefonie-Integration,
- Spracherkennung,
- Sprachsynthese.

Mit diesen technologischen Basiskomponenten alleine ist es jedoch noch nicht getan: In Abhängigkeit von den spezifischen Erfordernissen der jeweiligen Anwendung sind geeignete Vorgaben zu spezifizieren, die den Computer in die Lage versetzen, den Dialog mit seinem menschlichen Gegenüber *in problemadäquater Weise* zu führen. Wichtige Anforderungen sind:

- *Natürlichkeit*: Ausgestaltung der sprachlichen Interaktion in einer Weise, die den Erwartungen des Anwenders hinsichtlich des jeweiligen Anwendungsfalls entsprechen;
- *Flexibilität*: Anpassung an die Eigenarten des jeweiligen Nutzers (Novize oder geübter Anwender etc.);

*D-60433 Frankfurt am Main, E-Mail: roland@stuckardt.de

- *Robustheit*: geeignetes Handling von Missverständnissen, unvollständigem Benutzer-Input sowie Unzulänglichkeiten der maschinellen Sprachverarbeitung (insbesondere Fehler in der Spracherkennung) etc.

Formale Spezifikationen des maschinellen Dialogverhaltens werden als *Dialogmodelle* bezeichnet. Hinsichtlich der generischen Wiederverwendbarkeit[1] der Dialogsoftware ist es sinnvoll, derartige Beschreibungen in einem standardisierten Formalismus, einer *Dialogmodellierungssprache* abzufassen, die sich somit in erster Näherung als eine "Programmiersprache" für eine *generische Dialogmaschine* auffassen lässt. Folglich stellt sich die Frage, wie eine geeignete Dialogmodellierungssprache aussehen könnte.
In Bezug auf *webbasierte* Sprachportale wurde vom W3C die XML-basierte Dialogmodellierungssprache *VoiceXML* als Standardisierungsvorschlag erarbeitet ([7]). Im vorliegenden Dokument sollen zunächst *Reichweite und Grenzen der Sprache VoiceXML evaluiert* werden. Auf der Grundlage der Evaluation sollen *strategischen Empfehlungen* für Unternehmen abgeleitet werden, die sich als Anwendungsentwickler auf dem Innovationsmarkt der telefonbasierten Sprachportale betätigen wollen. Die zentralen Fragen lauten:

1. Welches sind die zentralen Probleme der Entwicklung telefonbasierter Sprachportale?
2. Inwieweit löst VoiceXML diese Probleme?
3. Inwiefern lohnt es sich somit, (z.B. zwecks Herausbildung eines Alleinstellungsmerkmals) auf die Technologie VoiceXML zu setzen?
4. Welche Alternativen existieren? In welchen anderen Bereichen sollte man ggf. Kernkompetenzen herausbilden?

2 Evaluation der Technologie VoiceXML

Mit VoiceXML, Version 1.00 liegt die Spezifikation eines XML-basierten Standards zur Dialogmodellierung vor ([7]). Es soll nun zunächst ein Überblick der Rahmendaten dieser Technologie gegeben werden. Die Dialogmodellierungs-Sprache VoiceXML

- wurde unter Maßgabe der Primärzielsetzung einer sprachdialogischen Annotation von *Web-Ressourcen* entworfen,
- verfügt über ein *umfassendes Inventar an Sprachkonstrukten* für alle Bereiche der Dialog- sowie Grammatik-/Lexikon-Modellierung,
- stellt *sowohl deklarative als auch prozedurale*[2] *Modellierungskonzepte* zur Verfügung,
- unterstützt *Subdialoge, Verifikations- und Retry-Sequenzen*,
- basiert auf einem programmiersprachenähnlichen *Variablen- und Bedingungs-Konzept*,
- ist hinsichtlich der zu unterstützenden Grammatikformalismen[3] noch nicht festgelegt - dies soll nicht per präskriptiver Standardisierung, sondern dynamisch durch den Markt geregelt werden -,

[1] Stichpunkt: Verringerung des "*Time to Market*" in Folgeprojekten
[2] d.h. zustandsübergangsorientierte, auf Endlichen Automaten basierende Modellierungskonzepte
[3] z.B. der Standard *JSGF* (Java Speech Grammar Format Specification) von Sun Microsystems (vgl. [5])

- spezifiziert formal einen sog. *Form Interpretation Algorithm*, der VoiceXML-Dialogmodelle ausführt (sog. *Dialogscheduling*).

Wenn somit VoiceXML eine derart mächtige Sprache ist, die aus internationalen Standardisierungsanstrengungen hervorgegangen ist und von zahlreichen *Global Players* unterstützt wird, so erscheint es auf den ersten Blick aus fachlichen und ökonomischen Gründen unumgänglich, auf diese Technologie zu setzen. Andererseits bestehen eine ganze Reihe von Einschränkungen, die man in einem kritischen Prozess der Abwägung berücksichtigen sollte:

- Die Adäquatheit einer Sprache bemisst sich nicht nach der *Anzahl* der zur Verfügung gestellten Konstrukte, sondern nach deren *Geeignetheit* zur Lösung des jeweils relevanten Anwendungsproblems. Im betrachteten Szenario besteht das Ziel in der Entwicklung telefonbasierter Sprachportale mit Datenbankzugriff.

- In VoiceXML lassen sich eine ganze Reihe von Sprachkonstrukten identifizieren, die auf die Problemstellung der Annotation von *Web*-Ressourcen zugeschnitten sind. In anderen Anwendungskontexten - etwa der hier betrachteten sprachbasierten Datenbankschnittstellen - könnten sich diese Konstrukte als *Overhead* erweisen, der zu hohen Implementierungskosten führt und überdies die Wartbarkeit und Wiederverwertbarkeit erschwert.

- Letzteres gilt im Speziellen für das nichtdeklarative zustandsübergangsbasierte Konzept zur Spezifikation von Dialogtransitionen, das in Bezug auf klassische Dialoganwendungen als unhandhabbar eingestuft wird[4].

- Im Rahmen der - wünschenswerten und segensreichen - Standardisierungsbemühungen wird *top-down* vorgegangen; zwar ist die Rahmensyntax bereits definiert, jedoch werden derzeit für zentrale Komponenten der Modellierungssprache keine verbindlichen Vorgaben gemacht. Insbesondere ist zum gegenwärtigen Zeitpunkt nicht festgelegt, welche Audiodatei-, Grammatik- und Inhaltserschließungsregel-Formate schlussendlich unterstützt werden sollen; dies sollen "*market pressures*" entscheiden ([7], S. 35f.).

- Selbst wenn man in Bezug auf Grammatiken davon ausgeht, dass etwa das *Java Speech Grammar Format* (JSGF) zu den unterstützten Formaten zählen wird, so wird dabei außer Betracht gelassen, dass JSGF selbst nur hinsichtlich einer Rahmensyntax standardisiert ist, dass jedoch andererseits offen gelassen wird, wie etwa die der Spezifikation der Konzeptextraktionsregeln dienende Attribuierungs-Sprache der semantischen *Tags* auszugestalten ist.[5] Hersteller von Basistechnologie für Sprachdialoglösungen (Spracherkennung, Konzept-Spotting) treffen daher zum gegenwärtigen Zeitpunkt *propriätere Festlegungen*[6]

- Ferner gelten *Performanz-Gesichtspunkte*: Die zweckmäßigerweise als funktionale Einheit zu betrachtende Kombination von VoiceXML-Dialogskript und Erkennergrammatik führt nicht mit allen Spracherkennern zu gleichermaßen akzeptablen Ergebnissen.

Die beiden zuletzt genannten Punkte sind von zentraler Relevanz. Sie lassen sich folgendermaßen paraphrasieren:

[4] Für eine weitergehende Diskussion dieses Problems vergleiche [1, 4, 6].

[5] siehe z.B. [4]

[6] vgl. z.B. die TEMIC-Technologie von Telefunken microelectronic, in der die semantischen Tags als *Eigenschaft-Wert-Strukturen* ausgestaltet sind ([4]).

> VoiceXML ist keine herkömmliche, voll durchstandardisierte Programmiersprache. Zum gegenwärtigen Zeitpunkt kann *nicht* davon ausgegangen werden, dass - sichergestellt durch Standardisierung - ein für eine bestimmte Umgebung konstruiertes und optimiertes Dialogskript auf beliebigen anderen dem VoiceXML-Standard genügenden Umgebungen
>
> 1. überhaupt lauffähig ist,
> 2. insofern lauffähig, zu einer akzeptablen Performanz führt.

Im Rahmen der Abwägung für und wider VoiceXML sind ferner Implementierungs- und Kosten-Nutzen-Aspekte zu berücksichtigen. Die Realisierung eines VoiceXML-kompatiblen Dialogschedulers stellt ein umfassendes Software-Projekt dar. Zwar existieren Werkzeuge zur einfachen Generierung von Dokumenten*parsern* für VoiceXML-Skripte (Vorteil der XML-Verwendung), jedoch ist die Implementierung eines semantischen Interpreters für die zahlreichen Sprachkonstrukte aufwändig und damit teuer.

3 Strategische Empfehlungen

Es sollen nun die Ergebnisse der VoiceXML-Bestandsaufnahme aus der strategischen Perspektive von Unternehmen beleuchtet werden, die sprachdialogbasierte, über Telefon zugreifbare Schnittstellen für Datenbanken entwickeln. Inwieweit stellt VoiceXML den Standard der Wahl zur Dialogmodellierung dar?

3.1 Fokus der VoiceXML-Standardisierung

Anhand der Aufstellung in Abschnitt 2 wird ersichtlich, dass die Sprache VoiceXML auf die Annotation von Web-Ressourcen zugeschnitten ist. In Bezug auf dieses Anwendungsfeld ist der verfolgte Weg der Top-Down-Standardisierung durchaus sinnvoll. Web-Ressourcen werden von einer Vielzahl unterschiedlicher Browser zugegriffen, die jeweils über eine spezifische Voice-Basistechnologie, bestehend aus Spracherkenner, Sprachsynthese und Dialogmanager verfügen. Während jedoch in Bezug auf das Dialogscheduling eine Standardisierung keine größeren Probleme aufwirft[7] und auch an die Sprachsynthesekomponente nur relativ elementare, technisch machbare Anforderungen gestellt werden, erweist sich die *Spracherkennung* als die kritische Software-Komponente: Eine Standardisierung des Erkennungsalgorithmus per se erscheint derzeit unmöglich, da das Problem noch nicht hinreichend gut / allgemeingültig (sprecherunabhängig, sprachübergreifend) gelöst ist. Da somit derzeit noch keine allseits anerkannte State-of-the-Art-Technologie zur Verfügung steht, kann lediglich ein Top-Down-Weg verfolgt werden, indem zunächst einmal *Format*-Standards für die von Erkenner und Inhaltserschließer zu verarbeitenden Grammatiken und semantischen Regeln erarbeitet werden. Wie in Abschnitt 2 ausgeführt, ist nun jedoch nicht damit zu rechnen, dass jede Grammatik auf jedem Erkenner zu guten Erkennungsleistungen führt.

3.2 VoiceXML für telefonbasierte Sprachdialogportale?

In Bezug auf sprachdialogbasierte, über Telefon zugreifbare Schnittstellen zu Datenbanken ergeben sich grundlegend unterschiedliche Anforderungen: Die *Sprachdialogtechnologie*

[7] Der Dialogmanager hat den *Form Interpretation Algorithm* der Technologie VoiceXML zu implementieren.

ist in diesem Falle *serverseitig lokalisiert*. Standardisierungsbedingungen, die in Szenarien clientseitiger Sprachdialogtechnologie, d.h. insbesondere im Falle von voice-fähigen Web-Browsern bestehen, sind hier irrelevant.[8]
Wenn nun aber die Sprachdialogtechnologie serverseitig läuft, so ist es prinzipiell unnötig, auf einen Standard wie VoiceXML zurückzugreifen, der darauf zugeschnitten ist, mit einer Vielzahl unterschiedlicher Spracherkennungs- und Dialogmodule zusammenzuarbeiten. Da hier die Rahmenparameter der jeweils intendierten Anwendung - insbesondere

- Sprache (Deutsch, Englisch, ...),
- Größe,
- Variabilität,
- phonetische Charakteristika (Ähnlichkeiten)

des erkennungsrelevanten Vokabulars - a priori feststehen, ist es in der Regel sinnvoll, einen *bestimmten* Spracherkenner zu wählen, der in Bezug auf diese *spezifischen* Rahmenparameter optimale Ergebnisse erzielt. Somit erweist es sich als nachrangig, Dialogmodell, Grammatik / Sprachmodell und Inhaltserschließungsregeln in standardisierten Sprachen zu formulieren; ausschlaggebend ist zunächst alleine die Lauffähigkeit auf der mit Blick auf die Anwendungs-Rahmenparameter ausgewählten Sprachtechnologie der Wahl.[9] Mit Blick auf die Komplexität der Aufgabe, überhaupt eine hinreichende, den Erfordernissen der jeweiligen Anwendung genügende Qualität in der Spracherkennung zu erzielen, sollte vielmehr der Fokus zunächst auf die *Performanz* im Anwendungskontext gelegt werden.
Darüberhinaus verfügt VoiceXML über eine Reihe von Sprachkonstrukten, die im Kontext der Annotation von Web-Ressourcen ihre Berechtigung haben, die jedoch in Bezug auf sprachdialogbasierte Datenbankschnittstellen irrelevant sind. Um dem Standard zu genügen, wäre die Semantik dieser Konstrukte in der Dialogscheduling-Komponente mitzuimplementieren; dieser Zusatzaufwand trägt jedoch nichts zur Lösung der eigentlichen zentralen Probleme bei und sollte deshalb aus wirtschaftlichen Erwägungen vermieden werden.[10]
Aus den obenstehenden Betrachtungen lassen sich eine Reihe von *strategischen Empfehlungen* ableiten.

3.3 Empfehlung: VoiceXML als *Ausgangspunkt*

VoiceXML ist auf den Einsatz in webbasierten Sprachportalen zugeschnitten, in denen die Sprachdialogtechnologie clientseitig lokalisiert ist. Im Rahmen des Designs dieser Sprache haben deshalb Standardisierungserfordernisse Vorrang vor optimalen, auf den jeweiligen

[8] In diesem Zusammenhang führe man sich die originäre Motivation clientseitig laufender Web-Browser vor Augen: Im Falle *grafischer* Schnittstellen sind die Charakteristika der jeweiligen Client-Hardware ausschlaggebend; Aufgabe des clientspezifischen Browsers ist es, die Inhalte clientdisplay-adäquat darzustellen. Im Falle sprachbasierter Inhalte sind die Hardware-Charakteristika der Zielmaschine weitgehend, d.h. modulo der Erfüllung bestimmter Mindestanforderungen irrelevant; jedoch erlaubt es die Architektur des Internet nicht, die Sprachdialogtechnologie konsequenterweise auf dem Server anzusiedeln.

[9] Eine Standardisierung bringt natürlich weitergehende Vorteile mit sich, die durchaus relevant sind, z.B. geringere Einarbeitungszeiten für Sprachtechnologie-Ingenieure, bessere Wiederverwertbarkeit im Falle eines Wechsels der Basistechnologie etc.

[10] Anders sähe es aus, wenn man auf einen in Kombination mit den Sprachtechnologiekomponenten der Wahl wiederverwertbaren Voice-XML-basierten Dialogscheduler zurückgreifen könnte und somit kein Overhead entstände.

Anwendungskontext abgestimmten Lösungen. In Bezug auf serverseitig lokalisierte sprachdialogbasierte Datenbankschnittstellen würde diese Festlegung auf den kleinsten gemeinsamen Nenner i.d.R. zu suboptimalen Lösungen führen. Dies erweist sich als inakzeptabel, denn auf der Basis des State-of-the-Art der Sprachdialogtechnologie ist es ohnehin bereits schwierig genug, den eingangs identifizierten Anwendungsanforderungen von Natürlichkeit, Flexibilität und Robustheit der maschinellen Dialogperformanz Genüge zu leisten.
Es wird daher empfohlen, auf eine Technologie zurückzugreifen, die

- hinsichtlich der spezifischen Erfordernisse der Anwendung optimiert ist - vgl. Abschnitt 3.2, Stichpunkt Rahmenparameter -,
- auf den Overhead webportalbezogener Sprachkonstrukte verzichtet.

Insbesondere ermöglicht dies die Implementierung eines *maßgeschneiderten*, unter Komplexitätsgesichtspunkten handhabba(re)ren Dialogschedulers. Das intelligente Dialogmanagement kann mit zusätzlichen Features versehen werden, die unter Maßgabe der spezifischen Anwendungszielsetzung sinnvoll erscheinen. Auch der Wiederverwertbarkeitszielsetzung kann in sachadäquater, problemfokussierter Weise Rechnung getragen werden: Indem der Dialogscheduler auf Dialogmodellen arbeitet, die in einer auf die Problemstellung sprachbasierter Datenbankschnittstellen zugeschnittenen Dialogbeschreibungssprache formuliert sind, wird eine *Generizität auf der relevanten Ebene der Anwendungsklasse* erreicht, die sich von der (im Webkontext erzwungenen) Form der technologischen Generizität auf der Ebene unterschiedlicher Sprachtechnologie-Basismodule grundsätzlich unterscheidet. Der Fokus der Implementierungsarbeiten kann somit auf der eigentlichen Anwendungsproblemlösung liegen. Zudem kann auch bei dieser Vorgehensweise auf eine Reihe von Vorarbeiten oder bereits verfügbare Technologie zurückgegriffen werden (vgl. etwa [1, 4, 6]).[11] Zumindest jedoch sollte VoiceXML als *Ausgangspunkt* proprietärer Entwicklungen dienen: Unbeschadet des aus der Weborientierung resultierenden Overheads ergeben sich in Sachen Dialogmodellierung größere Überschneidungen; bestimmte Sprachkonstrukte können deshalb als Vorbild für eine abgespeckte Modellierungssprache dienen. Die weitere Entwicklung des Standards VoiceXML sollte deshalb im Auge behalten werden.

3.4 Empfehlung: VoiceXML formal unterstützen

Mit Blick auf den Bekanntheitsgrad der Technologie und die zu erwartende positive Außenwirkung sollten Sprachtechnologiefirmen den Standard VoiceXML dennoch *aus marketingstrategischen Gründen formal "unterstützen"*. Dies könnte sich für solche Unternehmen als besonders bedeutsam erweisen, die ihre Aktivitäten zwar gegenwärtig auf telefonbasierte Sprachdialoglösungen fokussieren, jedoch mittel- bis langfristig in den Markt für webbasierte Sprachportale einsteigen möchten. Zwar ist aufgrund der sprachtechnologie-generischen Ausrichtung von VoiceXML damit zu rechnen, dass entsprechende Anwendungen in den nächsten Jahren notwendigerweise eine vergleichsweise elementare Funktionalität aufweisen (vgl. o., Stichpunkt "gemeinsamer Nenner"); aufgrund der hohen Anzahl von Nutzern fungiert das Web jedoch als wichtiger Motor für die Verbreitung spachdialogbasierter Portale, indem es die ergänzende Funktionalität des sprachbasierten Browsings einem breiten

[11] Die Forschungsgruppe TEMIC der Telefunken microelectronic GmbH Ulm verfolgt offensichtlich eine vergleichbare Strategie (vgl. [4]); es wird deshalb eine proprietäre, planbasiert-deklarative Lösung samt integrierter Entwicklungsumgebung für Dialogskripte, Grammatiken und Lexika entwickelt, die vermutlich mittelfristig (ab 2002?) als Komplettlösung zur Entwicklung telefonbasierter Sprachdialogsystemen vermarktet werden soll.

Anwenderkreis zur Verfügung stellt. Im Anwendungskontext webbasierter Sprachportale dürfte kein Weg am Standard VoiceXML vorbeiführen.

3.5 Empfehlung: Technologieintegrations-Kompetenz aufbauen

Wenn nun einerseits davon abgeraten wird, in Bezug auf serverbasierte Sprachdialogtechnologie auf den Standard VoiceXML zu setzen, und andererseits besonders für kleinere und mittelständische Unternehmen empfohlen wird, verfügbare Sprachdialogtechnologie soweit wie möglich wiederzuverwenden, so stellt sich die Frage, auf welchen Gebieten Kernkompetenzen aufgebaut werden sollten, um z.B. ein Alleinstellungsmerkmal herauszubilden.
Ein erster Bereich, in dem es sinnvoll erscheint, Expertise aufzubauen, ist die *Technologieintegration*. Auf dem Markt sind eine Vielzahl sprachtechnologischer Basismodule zu Computer-Telefonie-Integration, Spracherkennung, Dialogmanagement und Sprachsynthese verfügbar, die sich hinsichtlich ihrer Charakteristika z.T. grundlegend voneinander unterscheiden. Nicht jeder Spracherkenner führt in jedem Anwendungskontext zu optimalen Umgebungen, und nicht jede Lösung zur Computer-Telefonie-Integration ist kompatibel mit den Anforderungen des Kunden oder dessen bereits bestehender und zu unterstützender Hardware. Zu wissen

- welche leistungsfähigen Basismodule es gibt,
- welche Charakteristika diese haben,
- inwieweit die Basismodule technisch miteinander kombinierbar sind,
- in welches Anwendungs-Setting diese Basismodule passen

verschafft einen entscheidenden Kompetenzvorsprung gegenüber solchen Anbietern, die auf *bestimmte* Module festgelegt sind oder die einen hohen Aufwand in der Entwicklung einer *starren, integrierten*, alle Bereiche abdeckenden Standardlösung betreiben, die jedoch den spezifischen Erfordernissen im Einzelfall u.U. nicht gerecht wird. Dessen ungeachtet sollte man Kompetenzen auch betreffend die integrierten Lösungen aufbauen, die sich - sofern den jeweiligen Anforderungen genügend - i.d.R. einfacher und damit kostengünstiger konfigurieren lassen.

3.6 Empfehlung: Dialogmanagement-Kompetenz aufbauen

Ergebnis der obigen Bestandsaufnahme war die Feststellung, dass VoiceXML als Dialogmodellierungssprache den Erfordernissen der serverbasierten Sprachdialogtechnologie nur in Teilen gerecht wird (vgl. Abschnitt 3.3). In Bezug auf das Dialogmanagement im engeren Sinne, d.h. losgelöst von der Problematik der Spezifikation von Grammatiken/Sprachmodellen für die Spracherkennung, ergab sich daraus die Empfehlung, VoiceXML als Ausgangspunkt einer proprietären Entwicklung heranzuziehen.
Während für Spracherkennung und Synthese sowie Computer-Telefonie-Integration eine relativ große Anzahl von Modulen am Markt verfügbar sind, ist die Auswahl betreffend die Dialogmanagement-Technologie begrenzt. Elementare zustandsübergangsbasierte Lösungen genügen den Ansprüchen komplexer Dialoganwendungen nicht; komplexere deklarativ-/planbasierte Modellierungsansätze sind jedoch oftmals nicht über den Status von Forschungsprojekten hinausgekommen.[12] Wenn also überhaupt in die Entwicklung proprietärer

[12] Z.B. bietet Philips eine Technologie (SpeechMania) an, die auf einem zustandsübergangsbasierten Dialogmodell beruht; an den Philips-Forschungslaboratorien in Aachen wird jedoch bereits seit Mitte der

Technologien investiert werden soll, so empfiehlt es sich für kleine oder mittelständische Unternehmen, den Fokus auf die Dialogtechnologie im engeren Sinne zu legen. Dies setzt voraus, dass die entsprechenden computerlinguistischen Kompetenzen zur Verfügung stehen, die jedoch auf dem Arbeitsmarkt nur in sehr begrenztem Umfang angeboten werden. In jedem Falle sollte die Geschäftsführung zumindest über die notwendige Beurteilungskompetenz verfügen.

3.7 Empfehlung: Kompetenz im Engineering sprachdialogbasierter *Anwendungen* aufbauen

Erfahrungen aus einschlägigen Projekten zeigen, dass ein alleiniger Fokus auf die *technischen* Aspekte von Basiskomponenten und Komponentenintegration zu kurz greift. Aus der Perspektive des *Nutzers* bestehen Anforderungen an die *Nützlichkeit* - sprich: Problemlösungskompetenz - sprachdialogbasierter Anwendungen; eine alleinige Performanzoptimierung der System*komponenten* in formal-technischen Evaluationsdisziplinen stellt jedoch nicht sicher, dass das *Gesamt*system diesen anwendungspragmatischen Bedingungen genügt. Die Problemlösungskompetenz eines Sprachdialogsystems ist mehr als die technische Performanz der Komponentenmodule, m.a.W. es reicht *nicht* aus, den Spracherkenner mit bester Performanz in der Anwendungssprache und den linguistisch/pragmatisch elaboriertesten Dialogmanager miteinander zu kombinieren. Natürlich haben die Teilsysteme bestimmten Basisanforderungen zu genügen; eine weitere gewichtige Determinante der Nutzerakzeptanz ist jedoch die *Ergonomie* des Systems in toto: Der Benutzer nimmt eben die Sprachdialoganwendung *nicht* unter technologischen Gesichtspunkten wahr, sie oder er möchte einfach ein bestimmtes Problem lösen.
Mit anderen Worten: Auf der alleinigen Grundlage technischer Kompetenz lassen sich keine aus der - einzig maßgeblichen - Sicht des Nutzers performante Sprachdialogsysteme bauen; (mindestens) ebenso wichtig ist die Berücksichtigung sog. *weicher Faktoren*, die sich nicht einfach auf formalmathematische Optimierungsprobleme reduzieren lassen. Somit sind *zwei* Kompetenzbereiche abzudecken:

- das Handwerkvon Technologiewahl und Integration (vgl. Abschnitt 3.5) *und*
- die *Kunst* des Designs *nützlicher* sprachdialogischer Anwendungssysteme.

Gerade in technologieorientierten kleinen und mittelständischen Unternehmen wird oftmals der Fahler begangen, den Schwerpunkt zu sehr auf die Technologiebeherrschung zu setzen. Dass dies alleine nicht hinreichend ist, nützliche Software zu entwickeln, die *Anwendungsprobleme* löst, läßt sich anhand der strategischen Fehler belegen, die in einer ganzen Reihe von schlußendlich gescheiterten KI-Startupunternehmen gemacht wurden.
Ein Blick in die einschlägige Literatur von Künstlicher Intelligenz, Computerlinguistik und Natural Language Engineering offenbart eine vergleichbare Schwerpunktsetzung. Erst in jüngerer Zeit wurde damit begonnen, die Relevanz "weicher" Faktoren systematisch zu untersuchen. An erster Stelle ist hier das von der EU geförderte DISC-Projekt zu nennen, in dessen Rahmen an einer Methodik zur systematisch-fundierten Entwicklung sprachdialogbasierter Anwendungssysteme gearbeitet wird.[13] Zum gegenwärtigen Zeitpunkt liegen bereits eine ganze Reihe von Erkenntnissen aus dem DISC-Projekt vor, u.a. eine Aufstellung relevanter "*Human Factors*", Instrumente für eine methodische Vorgehensweise wie

neunziger Jahre an einer wesentlich ausgefeilteren, in der Dissertation [1] von Harald Aust partiell dokumentierten deklarativen Technologie für Dialogscheduling und Modellierung (HDDL) gearbeitet.

[13] vgl. [3] sowie die umfassende Ressourcensammlung des DISC-Projekts ([2])

Fragebögen zur Datenerhebung, Software-Werkzeuge etc.
In diesem Bereich Kompetenzen aufzubauen, dürfte noch um einiges schwieriger sein als die Beherrschung der technischen Seite der Sprachdialog-Anwendungsentwicklung. Andererseits bietet sich gerade hier - unter gleichzeitiger Vermeidung eines verbreiteten Strategiefehlers einschlägiger (mittlerweile gescheiterter) KI-Startups - eine echte Chance zur Herausbildung eines *Alleinstellungsmerkmals* auch für kleine oder mittelständische Unternehmen. Wer als erster nicht nur über technische Kompetenz in der Integration sprachdialogischer Basismodule verfügt, sondern darüberhinaus die Kunst des Designs sprachdialogischer Anwendungssysteme beherrscht, für den sollte der Weg frei sein zur mittelfristigen Erringung der *Marktführerschaft*.

Literatur

[1] Harald Aust. *Sprachverstehen und Dialogmodellierung in natürlichsprachlichen Informationssystemen.* Dissertation, RWTH Aaachen, Fachgruppe Informatik, Reihe Aachener Informatik-Berichte 98-8.

[2] *Internetseiten des DISC-Projekts*, ohne Datum. `http://www.disc2.dk/`

[3] Laila Dybkjaer, Niels Ole Bernsen. *Usability Issues in Spoken Dialogue Systems.* In: *Natural Language Engineering. Special Issue on Spoken Language Dialogue System Engineering.* Vol. 6, Parts 3 & 4, September 2000.

[4] Marus Hennecke, Gerhard Hanrieder. *Easy Configuration of Natural Language Understanding Systems.* In: *Proceedings of VOTS*, 2000.

[5] *Java Speech Grammar Format Specification Version 1.0*. Sun Microsystems Inc., 26. Oktober 1998.

[6] Roland Stuckardt. *Entwurf einer deklarativen Dialogmodellierungsprache und eines Scheduling-Algorithmus*. Technologieskizze, Knowbotic Systems GmbH & Co. KG, 30. August 2000.

[7] *Voice eXtensible Markup Language VoiceXML, Version 1.00*. 7. März 2000. Verfügbar unter `http://www.voicexml.org/specs/VoiceXML-100.pdf`.