Kontextsensitives Fehlermanagement bei multimodaler Interaktion mit Infotainment- und Kommunikations-einrichtungen im Fahrzeug

Context-Sensitive Error Management during Multimodal Interaction with Car Infotainment and Communication applications

G. McGlaun, **M. Lang**, **G. Rigoll**, Technische Universität München;
F. Althoff, BMW Forschung und Technik, München

Kurzfassung

In diesem Beitrag wird ein multimodales System zur Bedienung von Infotainment- und Kommunikationsapplikationen im Kfz vorgestellt, welches mittels einer Fuzzy-Logik bzw. eines neuronalen Netzes aus Eingaben und Kontextwissen system- und nutzerseitige Fehlermuster bzw. –potenziale detektiert und darauf mit einem kontextadäquaten fehlerauflösenden Dialog reagiert. In einem Usability-Test wurde das entwickelte System unter Laborbedingungen hinsichtlich Leistungsfähigkeit und Nutzerakzeptanz evaluiert.

Abstract

In this contribution, we present a multimodal system for operating car infotainment and communication applications. Exploiting inputs and contextual knowledge with a fuzzy-logic and a neural net, the system is capable of detecting user- as well as system-specific error patterns or potentials, and reacts with a context-adequate error-resolving dialog. In a usability test, we evaluated the system under lab conditions regarding efficiency and user acceptance.

1. Einleitung

In modernen Kfz-Einrichtungen ist ein enormer Zuwachs an Technologien, wie Internet, drahtloser Kommunikation sowie Komfortapplikationen und Infotainment-Systemen zu verzeichnen. Damit inhärent ist aber auch eine stark gesteigerte Komplexität des Bedienumfangs. In der Domäne Fahrzeug treten daher oft Situationen auf, in denen es zu Fehlern bei

Quelle: USEWARE 2004, VDI Verlag Düsseldorf

der Mensch-Maschine-Interaktion mit verschiedenen Applikationen kommt. Während der Benutzer Bedienvorgänge etwa an einem Home-PC überwiegend konzentriert ausführen kann, ist in der Automobilumgebung meist eine gewisse Basis-Workload des Fahrers vorhanden. Der Offset an mentaler Belastung entsteht aufgrund der Erledigung *primärer* (Navigation, Fahrzeugstabilisierung etc.) und *sekundärer Aufgaben* (Blinken, Gangwechsel etc.) und wird ggf. durch *Umgebungseinflüsse*, wie etwa die Unterhaltung mit einem Insassen, verstärkt. Interagiert der Fahrer in solch einer Belastungsphase z.B. mit im Fahrzeug befindlichen Infotainment- oder Kommunikationssystemen (*tertiäre Aufgabe*), so sind häufig Unaufmerksamkeit, Stress bzw. Irritation die Folgen einer aus der Superposition oben genannter Aufgaben resultierenden hohen Belastung des Fahrers, welche sich in einem erhöhten Fehlerpotenzial bzw. in Fehlbedienungen dieser Tertiärsysteme manifestieren. Daher werden in der Kfz-Domäne zunehmend *multimodale Interfaces* (MIs) eingesetzt, die neben klassisch *taktilen* Sensoren (Buttons, Drehdrücksteller etc.) auch *sprachliche* bzw. *gestische* Eingabeparadigmen beinhalten[1]. Gegenüber monomodalen Systemen ermöglichen MIs neben signifikant verkürzten Lernphasen eine individuelle und intuitive Form der Interaktion, da sie den natürlichen menschlichen Kommunikationsgewohnheiten gerecht werden[2]. Falls die gewählte Modalität gestört ist, kann dem Fahrer eine alternative Eingabeform als Fallback in Abhängigkeit von der gegebenen Funktionalität bzw. vom situativen Kontext nahegelegt werden; wenn sich etwa die Erkennungsleistung eines Spracherkenners durch starken Umgebungslärm drastisch verschlechtert, bietet das System bei stehendem Kfz dem Benutzer alternativ z.B. taktile Eingabe an. Oviatt et al. führen aus, dass in dedizierten Szenarios bis zu 86% aller taskkritischen Fehler vermeidbar sind, wenn eine alternative Eingabemodalität vorhanden ist[2]. Bei synchroner Eingabe in unterschiedlichen Modalitäten können *wechselseitige Ambiguitäten*[3] eliminiert werden. Um den Nutzer vom Fahren möglichst wenig abzulenken sowie den Bedienkomfort zu erhöhen, ist ein *kontextsensitives Fehlermanagement* notwendig, welches potenzielle Fehlerquellen bei der Bedienung oben genannter Applikationen im Vorfeld identifiziert und entsprechende Präventionen (z.B. Warnhinweise) einleitet (*A priori-Fehlermanagement*). Wenn Fehler aufgetreten sind, sollen diese rasch, effektiv und für den Nutzer möglichst transparent behoben werden, um eine weitere Ablenkung bzw. Fehlerresonanzen zu verhindern (*A posteriori-Fehlermanagement*).

2. Themenverwandte Arbeiten

Die Fehlerrobustheit bei der Interaktion im Kfz wurde bereits in einigen Vorarbeiten untersucht. Ziel im Projekt EMBASSI[4] war die Schaffung eines aus modularen Dialog- und Assistenzkomponenten bestehenden ganzheitlichen Systems, das den Nutzer durch Bereitstel-

len multimodaler Schnittstellen bei der Bedienung einer großen Bandbreite von Alltagstechnologien möglichst optimal unterstützt. In der Kfz-Domäne wurden u.a. die Themen Unterhaltungswunsch und positionsabhängige Umwelterfassung (z.B. Staumeldungen) adressiert.
Eine Forschungsgruppe am Lehrstuhl für Technische Informatik der RWTH Aachen[5] befasst sich unter anderem mit der intuitiven Gestaltung multimodaler Kfz-Bordsysteme, wobei insbesondere auf die Nutzeradaption in unterschiedlichen Bediensituationen fokussiert wird.

3. Theoretischer Hintergrund

Allgemein spricht man von einem *Fehler in der Mensch-Maschine-Interaktion*, wenn das gewünschte Ziel des Benutzers nicht erreicht wird und dafür kein Zufall verantwortlich ist. Grundsätzlich lassen sich *benutzer-* und *systemseitige* Fehler unterscheiden.
Die bei der Modellierung der nutzerseitigen Fehlerklassen zugrundeliegende Taxonomie geht auf die Veröffentlichungen von J. Reason[6] und J. Rasmussen[7] zurück. Bei der Untersuchung der Fehlerursache können drei wiederkehrende Level zur Fehlertypisierung identifiziert werden: Fehler auf dem *Skill-Based Level* (z.B. das Verfehlen oder Abrutschen von einem Button), dem *Rule-Based Level* (z.B. Einsetzen eines sonst gültigen Sprachbefehls in einem Modus, in dem er nicht erlaubt ist) und dem *Knowledge-Based Level* (z.B. Verwendung eines dem System unbekannten Sprachbefehls).
In den Fehlermodellen werden auch systemseitige Fehler adressiert. Hierzu zählt die Gruppe der *Erkennungsfehler*, d.h. Nicht- oder Fehlerkennung einer korrekten Eingabe des Benutzers oder eine inkorrekte systemseitige Aktivierung des Spracherkenners (weil z.B. der Nutzer das Schlüsselwort, auf das der Spracherkenner triggert, beiläufig in einem Gespräch mit dem Beifahrer verwendet). Des Weiteren können *Verarbeitungsfehler* (Timingprobleme oder widersprüchliche Erkennungsergebnisse verschiedener Einzelerkenner etc.) sowie *technische Fehler* (Overflow, Ausfall von Systemkomponenten etc.) auftreten. Eine ausführliche Charakterisierung system- und nutzerseitiger Fehler sowie weitere Beispiele finden sich in[8].
Wurden System- oder Benutzerfehler bzw. -fehlerpotenziale erkannt, so versucht das System, diese mittels eines Dialogs aufzulösen. Mögliche anzuwendende Strategien (Warnung, Aufforderung zur wiederholten Befehlseingabe, Aufforderung zum Modalitätenwechsel, Auswahl aus Alternativen, Tutorial, Forcing Function, Gagging, Self-Correct etc.) unterscheiden sich anhand folgender Charakteristika: Initiierung der Fehlermeldung, Stärke der Kontextabhängigkeit, Grad der Berücksichtigung individueller Merkmale des Benutzers (z.B. dessen Lernstand bzgl. des Systems), Mächtigkeit der Strategie (d.h. Ausführlichkeit der Wissensvermittlung und der Umfang der Intervention) sowie Inklusion des Nutzers (d.h. Grad der Interaktivität beim Korrekturvorgang sowie Umfang des Systemfeedbacks). Die Wahl der

Dialogstrategie hängt wesentlich von kontextuellen Einflussparametern ab. Dazu zählen der persönliche Zustand (z.B. emotionales Verhalten) des Fahrers, die situativen Gegebenheiten (z.B. Parkplatz, Stadtverkehr, Autobahn) sowie der Systemkontext (gewählte Eingabemodalität, Status bzw. Modus der Applikation).

Wenn die Strategie gewählt ist, muss auf Applikationsseite die Information dem Nutzer individuell und kontextadäquat dargeboten werden. Das Systemfeedback im Kfz wird wesentlich durch die Art der Kombination und die individuelle Konfiguration der jeweiligen Feedbackdevices bestimmt und kann *optisch* (Ausgabegerät (HUD, Instrumentenkombi, zentrales Display), Ausführlichkeit der Darstellung, Formatierung (Font, Hervorhebung), Verhältnis Symbolik zu Text etc.), *akustisch* (Variation über die Dialoglänge, Sprachausgabe vs. tonales Feedback, Intonation etc.) bzw. *haptisch* (Art des Vibrationsmusters (Intensität, Dauer), Feedbackelemente (aktives Gaspedal, Stellteile, Lenkrad, etc.)) erfolgen.

4. Systembeschreibung

Die fehlererkennenden bzw. –behandelnden Module (*Fehler-*, *Kontext-*, *Dialog-* und *Feedback-Sentinels*, vgl. Bild 1) sind Teil einer multimodalen Systemarchitektur, in der die strikt *paramodal* repräsentierten Informationen der Einzelerkenner, basierend auf einer *Late-Semantic-Fusion*, nach einer entsprechenden Vorverarbeitung unter Berücksichtigung von *Kontextparametern* und *applikationsspezifischem Wissen* im *Intentionsdekoder* kombiniert werden. Für eine ausführliche Darstellung des Fusionsprozesses selbst sei auf[9] verwiesen. Das Ergebnis der multimodalen Fusion sind mögliche Kombinationen aus Befehlen und Parametern, welche in Form von Hypothesen vorliegen. Aus letzteren werden nun die für das Fehlermanagement benötigten Merkmale extrahiert, deren Relevanz durch Auswertung von Vorversuchen[10] bestimmt worden ist.

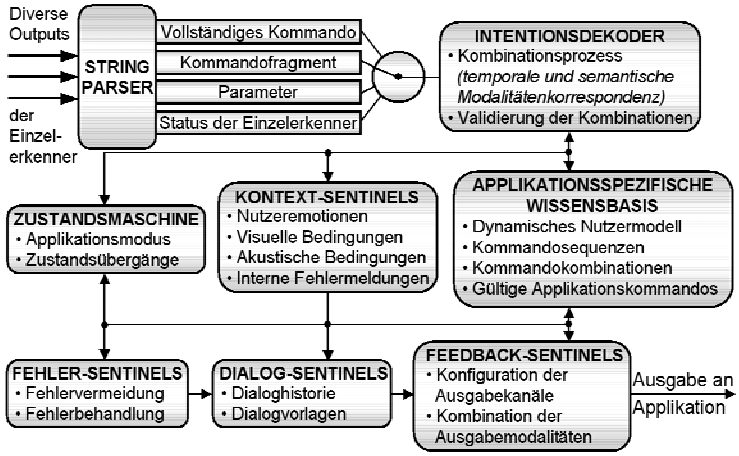

Bild 1: Einzelkomponenten des multimodalen Gesamtsystems

Grundsätzlich können *Befehlsmerkmale* (Interaktion, konkurrierende Interaktionen, korrigierende Interaktionen, redundante Interaktionen, Befehlswiederholung, Modalitätenwechsel, Zeitrelationen zwischen Befehlen) und *kontextuelle Merkmale*, wie Spezifika des Benutzers (Systemerfahrung, emotionale Merkmale etc.), Umwelteinflüsse (z.B. Fahrzeuggeschwindigkeit, Beifahrer) sowie Systemparameter (z.B. Lautstärke des Radios, Konfidenzmaße der Einzelerkenner) unterschieden werden. Die Verifikation eines Fehlerpotenzials bzw. die Detektion eines vorhandenen Fehlermusters erfolgt mittels dynamischer *Fehler-Sentinels* (error sentinels, ES). Hierbei handelt es sich um einen Pool hochspezialisierter Einzelagenten, wobei jeder ES speziell auf die Überprüfung eines individuellen Fehlermusters ausgelegt ist. Dabei können einzelne ES als Instanz einer Klasse generischer Sentinels betrachtet werden, wodurch das System auch auf andere Domänen portierbar und in seinem Umfang einfach skalierbar ist. Zur Validierung eines individuellen Fehlermusters dient ein Ansatz aus dem *Fuzzy-Controlling*. Die Methode wurde gewählt, weil sie bei relativ geringem Trainingsaufwand eine schnelle Realisierungsbasis im Sinne eines Proof of Concept darstellt und sich aufgrund der Unschärfe der formulierten Bedingungen als deutlich flexibler im Vergleich zu einem hart entscheidenden Expertensystem erweist. Die ES wurden dabei als Regler nach Mamdani implementiert. Die Regelkorpora umfassen je nach ES 22 bis 103 Regeln, wobei eine Regel zwischen drei und acht Messgrößen (Komponenten des Merkmalsvektors) enthält. Pro Messgröße existieren je zwei bis sieben Fuzzy-Wertebereiche, wobei die Zugehörigkeitsfunktionen für die einzelnen Fuzzy-Mengen als Dreiecksfunktionen realisiert wurden. Die Defuzzifizierung erfolgt mittels der Center-of-Gravity (COG)-Methode, da sie i.a. ein glattes Regelverhalten erzeugt. In Bild 2 ist ein vereinfachtes Beispiel dargestellt, welches die prinzipielle Funktionsweise des Algorithmus aufzeigt.

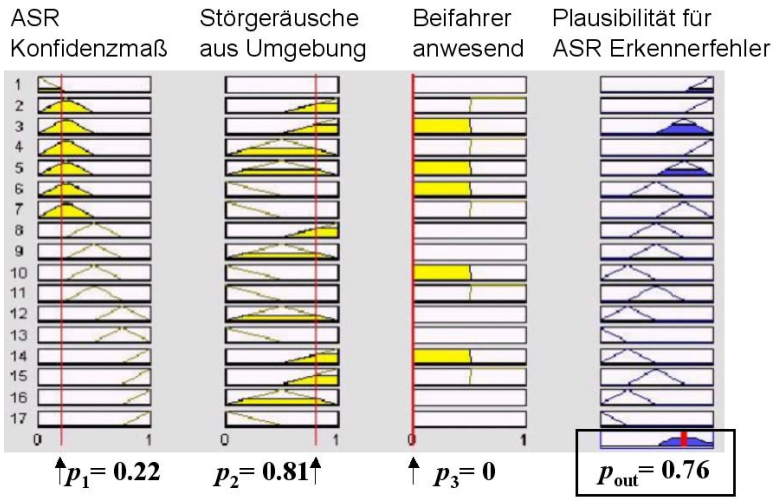

Bild 2: Ermittlung der Plausibilität eines systemseitigen Fehlers am Beispiel Spracherkennung

Hierbei bezeichne die Ausgabevariable p_{out} die Plausibilität für einen Erkennungsfehler des automatischen Spracherkenners (ASR). Der Fuzzy-Regelkorpus umfasst insgesamt 17 individuelle Regeln r_i, i = 1,2,…,17, wobei jede drei Messgrößen enthält (alle Werte sind normalisiert): das Konfidenzmaß des ASR (p_1), die Intensität der Störgeräusche aus der Umgebung (p_2) und das Bool-Flag „Beifahrer anwesend“ (p_3). p_1 besitzt vier Fuzzy-Regelblöcke, p_2 insgesamt drei und p_3 zwei Regelblöcke. Wenn der ES etwa die Werte p_1 = 0,22, p_2 = 0,81 und p_3 = 0 erhält, greifen die Fuzzy-Regeln r_1, r_3 und r_5. Durch Superposition der entsprechenden Membership-Funktionen der Ausgabevariablen erhält man p_{out} = 0,76. Dieser Wert kann als Bewertungsmaßzahl im Sinne einer Plausibilität, dass ein systemseitiger Fehler vorliegt, aufgefasst werden.

Jeder ES wertet anhand seines spezifischen Kriterienkataloges die entsprechenden Merkmale aus und ermittelt daraufhin für den repräsentierten Fehlertyp eine entsprechende Wahrscheinlichkeit und einen Zeitstempel. Der oder die resultierenden Fehlertypen werden mittels eines dynamischen Schwellwertentscheiders bestimmt. Das System besitzt so das Potenzial, sogar zeitlich bzw. inhaltlich superponierte Fehler oder Fehlerresonanzen zu erkennen. Die Erkennung und Evaluierung dieser komplexen Fehlermuster wurden jedoch im momentanen Entwicklungsstand des Systems noch nicht umgesetzt.

Als Einflussgrößen für die Strategiewahl werden in den *Dialog-Sentinels* (DS) aktuelle Kontextparameter sowie die Fehlertypen als Singletons in Regelsätzen fuzzyfiziert. Insgesamt stehen zwölf verschiedene Korrekturstrategien (siehe 3.) in Form von *Dialog-* bzw. *Aktionstemplates* zur Verfügung. Jeder Strategietyp ist durch einen DS instanziiert. Aufgrund der Eingabegrößen ermittelt jeder DS mittels eines Mamdani-Reglers in analoger Weise wie ein ES ein Plausibilitätsmaß für die von ihm repräsentierte Strategie. Als Ergebnis erhält man am Regelausgang eines DS eine zwischen 0 und 1 liegende Bewertungsmaßzahl für jedes Strategietemplate. Aufgrund der relativen Vergleichbarkeit wird aus der Menge der Strategietemplates das mit der höchsten Bewertungsmaßzahl gewählt.

Die Aufgabe der *Feedback-Sentinels* (FS) ist die situationsadäquate Aufbereitung bzw. Adaption des Strategietemplates. Im umgesetzten System wurde insbesondere über die Parameter *Schriftgröße, Wortwahl, Intonation* und *Dialoglänge* variiert. Hierzu wurden prototypisch vier FS, basierend auf je einem Feed-Forward-KNN (Künstliches Neuronales Netz) mit acht Ein- und vier Ausgängen sowie einem Hidden Layer mit vier Neuronen etabliert. Das KNN war zuvor mit dem *Levenberg-Marquadt-Algorithmus* auf 1024 Datensätzen trainiert worden. Ein Symbolkollektor sammelt die Ausgabewerte des KNN, belegt das ausgewählte Stratiegietemplate und generiert daraus schließlich die entsprechende Aktion bzw. Dialogausgabe.

5. Evaluierung

In einer Vorversuchsreihe[10] wurden mit dem multimodalen Basissystem ohne Fehlermanagementkomponente fünf verschiedene Fehlerszenarien provoziert. In einem Laborszenario mussten 40 Testpersonen in einer Fahrsimulation einem Straßenverlauf folgen und bekamen währenddessen diverse Aufgaben zur Bedienung des multimodalen Systems gestellt. Zielapplikation war ein Infotainment- und Kommunikationssystem mit Standardfunktionalitäten, bestehend aus einem CD / MP3-Player, Radio, Telefonie und einer WAP-Simulation. Als taktile Modalitäten standen dem Nutzer ein 10“-Touchscreen in der Mittelkonsole (der gleichzeitig als visuelle Ausgabeschnittstelle diente) sowie ein in die Armlehne integriertes Array aus acht Buttons und zwei Drehdrückstellern zur Verfügung. Des Weiteren konnten Eingaben auch in natürlicher Sprache (30 verschiedene Befehle mit Keyword-Initializing) sowie per Gestik (15 dynamische Hand- sowie sechs dynamische Kopfgesten, jeweils ohne Initialisierungsparadigma) erfolgen. Sechs Systembefehle (z.B. „ja” und „nein”) konnten in allen Modalitäten eingegeben werden. Zur zielgerichteten Provokation von Fehlerszenarien wurden sämtliche Benutzereingaben bis auf die taktilen Modalitäten vom Versuchsleiter interpretiert (*partieller Wizard-of-Oz-Versuch*). Die vorgenommenen Interaktionsschritte der Versuchsteilnehmer sowie Kontextparameter aus der Fahraufgabe (z.B. Geschwindigkeit) wurden dabei mitgeloggt. Nach Durchführung der Testreihen wurden die aufgetretene Fehlerszenarien in Form von Testdatensätzen transkribiert und katalogisiert. Das entwickelte Fehlermanagement-System wurde dann mit einem Teil der Datensätze trainiert und in fünf Testszenarien mit jeweils 15 Testdatensätzen offline evaluiert. Trainings- und Testdatensätze waren strikt disjunkt. Im Test wurden Fehlertypen bzw. die Dialogstrategien mit Raten zwischen 86,7% und 94,3% bzw. 91,0% und 96,1% korrekt klassifiziert bzw. gewählt. Die Strategietemplates wurden durch das KNN in 92,7% aller Fälle adäquat belegt.

Die oben beschriebene Versuchsreihe wurde schließlich mit den gleichen Testpersonen wiederholt, wobei nun die Fehlermanagementkomponente in das multimodale System integriert war. Die erzeugten Fehlerszenarien waren bis auf die Reihenfolge mit denen des Vorversuchs identisch. 95,0% der Testpersonen bewerteten das entwickelte System als hilfreich und effizient. Drei von 40 Versuchspersonen kritisierten in bestimmten Situationen (Stressszenarien, wie z.B. im Hochgeschwindigkeitsbereich) unangemessen zu lange Dialoge. Zwei Versuchspersonen fühlten sich durch das System in bestimmten Situationen bevormundet. Die Bedienzeit (Zeit zwischen Aufgabenstellung und Erreichen des Aufgabenziels) war in der Versuchreihe mit Fehlermanagement in vier von fünf Fehlerszenarien signifikant ($p < 0,01$) kürzer. Ein Benchmark-Test zeigte, dass das mit dem in 4. beschriebenen Regelkorpus implementierte System echtzeitfähig ist.

6. Ausblick

Laufende bzw. nachfolgende Arbeiten adressieren eine benutzerzentrierte Akzeptanzuntersuchung des Systems in einem Usability-Test im Feld bzw. mit realen Erkennern. Des Weiteren soll die Eignung statistischer bzw. hybrider Klassifikatoren (Hidden-Markov-Modelle bzw. Neuro-Fuzzy-Systeme) evaluiert werden. Ein besonderer Schwerpunkt liegt schließlich auf der Behandlung komplexerer Fehlermuster (Fehlersuperposition und –fortpflanzung).

7. Literatur

[1] Projekt FERMUS (Fehlerrobuste Multimodale Sprachdialoge); www.fermus.de; 2003

[2] S. Oviatt und R. van Gent: „Error Resolution during Multimodal Human-Computer Interaction“; in: Proceedings of ICSLP 1996, Vol. I, pp. 204-207; Philadelphia, USA; 1996

[3] S. Oviatt: „Taming Speech Recognition Errors Within a Multimodal Interface“; in Communications of the ACM, no. 43 (9), pp. 45-51; 2000

[4] Projekt EMBASSI (Elektronische Multimediale Bedien- und Service Assistenz); www.embassi.de; 2003

[5] S. Akyol, L. Libuda und K.-F. Kraiss: „Multimodale Benutzung adaptiver Kfz-Bordsysteme“; in: Th. Jürgensohn, K.-P. Timpe (ed.): Kraftfahrzeugführung, Springer-Verlag, Berlin; 2001

[6] J. Reason: „Generic Error-Modelling System (GEMS): A Cognitive Framework for Locating Common Human Error Forms“; in: J. Rasmussen, K. Duncan, J. Leplat (ed.): New Technology and Human Error, Kap. 7, pp. 63-83; 1987

[7] J. Rasmussen: „Skills, Rules, Knowledge: Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models“; in: Transactions: Systems, Man & Cybernetics, SMC-13; pp. 257-267; 1983

[8] G. McGlaun, F. Althoff, M. Lang und G. Rigoll: „Development of a Generic Multimodal Framework for Handling Error Patterns during Human-Machine Interaction“; angenommen auf der 8. World Multiconference on Systemics, Cybernetics, and Informatics (SCI 2004), Orlando, Florida, USA; 2004

[9] G. McGlaun, F. Althoff und M. Lang: “A New Approach for Integrating Multimodal Input via Late Semantic Fusion“; GMA Fachtagung USEWARE 2002, Darmstadt, Deutschland; in: VDI-Bericht 1678; pp. 181-185; 2002

[10] F. Althoff, G. McGlaun, M. Lang und G. Rigoll: „Evaluating Multimodal Interaction Patterns in Various Application Scenarios“; 5. Int. Workshop on Gesture and Sign Language Based Human-Computer Interaction 2003, Genua, Italien; in: Post-Proceedings: Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin; 2004