[ZILL82] Zillig, W. (1982): Bewerten: Sprechakttypen der bewertenden Rede. Tübingen: Niemeyer. [MORI*: 377]

[ZISS85] Zissos, A. Y. and I. H. Witten (1985): User Modeling for a Computer Coach: A Case Study. International Journal of Man-Machine Studies 23, 729–750. [KASS*: 391)]

Subject Index

acceptability SPAR*: 347, 349ff., 356f.
acquisition of user models WAHL*: 10–14; CHIN*: 85f., 86–89, 91–94; QUIL*: 129; MCCO*: 235, 254; I4: 337, 339; FINI*: 416
– system-driven vs. user driven MORI*: 375
– system-driven MORI*: 374, 377, 379
– user-driven MORI*: 374, 383, 385
ACT* KASS*: 404
action-achieving dialog, *see* dialog, action-achieving
active path CARB*: 138, 140
actor model, *see* agent model
adaptive parsing, *see* parsing, adaptive
addition change type MORI*: 373
advice WAHL*: 5, 27; I2: 70f.; CHIN*: 74; QUIL*: 108, 126; JAME*: 267, 294; I4: 338, KASS*: 400, 407
agent WAHL*: 6f., 20; KOBS*: 54, 56, 60; CHIN*: 75, 104; I4: 336; SPAR*: 343f.
– definition by Sparck Jones I4: 337
– model WAHL*: 7, 20; COHE*: 330f.; I4: 337; MORI*: 366; KASS*: 406
– modeled KOBS*: 54, 58
– modeling KOBS*: 54, 58; I4: 337
– role SPAR*: 343f., 346–354, 356, 360f.
analogy MCCO*: 241
answer CARB*: 158
– expression CHIN*: 103
anticipation feedback loop WAHL*: 22f., 25f.; I3: 198; JAME*: 274; KASS*: 407
– global WAHL*: 24
– local WAHL*: 22
applicability conditions CARB*: 136
apprentice-expert dialog, *see* dialog, apprentice-expert
appropriate query assumption CARB*: 143, 151, 153
AQUA WAHL*: 27; I2: 71; CHIN*: 101; QUIL*: 128
ARGOT WAHL*: 12, 19, 32; KOBS*: 53, 57, 59f., 63–66; I2: 72
assumptions about the user WAHL*: 5f., 9, 11–15, 20, 27f., 32; I2: 70–73; CHIN*: 94, COHE*: 313; I4: 336ff.; KASS*: 408
ATMS RICH*: 47
attitude change JAME*: 287, 292
attribute salience MCCO*: 243–249, 251, 253
attribution of beliefs and goals KOBS*: 53f.

background COHE*: 313–317, 319–328, 331ff.
backward chaining FINI*: 428
Bayes' theorem RICH*: 47
Bayesian inference JAME*: 293, 303
belief
– discrepancy MCCO*: 235
– report KOBS*: 55, 57
– representation WAHL*: 7, 15ff., KOBS*: 53, 58, 65, 67; QUIL*: 111–115; KASS*: 406
– revision WAHL*: 28, 33; JAME*: 285, 287; COHE*: 325–328
– types I1: 3; KOBS*: 53f., 56ff., 60–65, 67; CHIN*: 82
– verification QUIL*: 121, 131
beliefs WAHL*: 4f., 7–13, 16f., 26f., 29–33; KOBS*: 52ff., 57–65; I2: 70ff.; QUIL*: 109–112, 126, 129; CARB*: 136, 144ff., 151ff., 157; I3: 197; PARI*: 201, 206, 212; MCCO*: 236; I4: 339; SPAR*: 356, 362; MORI*: 376; KASS*: 391, 407, 409; FINI*: 415
– basic KOBS*: 56f., 59, 61
– collective WAHL*: 4, 29
– complex WAHL*: 32; KOBS*: 54, 56f., 60–63; JAME*: 277, 293
– contradictory WAHL*: 7; I2: 72; QUIL*: 115–121
– domain-specific QUIL*: 115, 123f.
– embedded, *see* beliefs, complex
– infinite-reflexive WAHL*: 7, 11; KOBS*: 54, 61ff.

[ZILL82] Zillig, W. (1982): Bewerten: Sprechakttypen der bewertenden Rede. Tübingen: Niemeyer. [MORI*: 377]

[ZISS85] Zissos, A. Y. and I. H. Witten (1985): User Modeling for a Computer Coach: A Case Study. International Journal of Man-Machine Studies 23, 729–750. [KASS*: 391)]

Subject Index

acceptability SPAR*: 347, 349ff., 356f.
acquisition of user models WAHL*: 10–14; CHIN*: 83f., 86–89, 91–94; QUIL*: 129; MCCO*: 235, 254; I4: 337, 339; FINI*: 416
– system-driven vs. user driven MORI*: 375
– system-driven MORI*: 374, 377, 379
– user-driven MORI*: 374, 383, 385
ACT* KASS*: 404
action-achieving dialog, *see* dialog, action-achieving
active path CARB*: 138, 140
actor model, *see* agent model
adaptive parsing, *see* parsing, adaptive
addition change type MORI*: 373
advice WAHL*: 5, 27; I2: 70f.; CHIN*: 74; QUIL*: 108, 126; JAME*: 267, 294; I4: 338; KASS*: 400, 407
agent WAHL*: 6f., 20; KOBS*: 54, 56, 60; CHIN*: 75, 104; I4: 336; SPAR*: 343f.
– definition by Sparck Jones I4: 337
– model WAHL*: 7, 20; COHE*: 330f.; I4: 337; MORI*: 366; KASS*: 406
– modeled KOBS*: 54, 58
– modeling KOBS*: 54, 58; I4: 337
– role SPAR*: 343f., 346–354, 356, 360f.
analogy MCCO*: 241
answer CARB*: 158
– expression CHIN*: 103
anticipation feedback loop WAHL*: 22f., 25f.; I3: 198; JAME*: 274; KASS*: 407
– global WAHL*: 24
– local WAHL*: 22
applicability conditions CARB*: 136
apprentice-expert dialog, *see* dialog, apprentice-expert
appropriate query assumption CARB*: 143, 151, 153
AQUA WAHL*: 27; I2: 71; CHIN*: 101; QUIL*: 128
ARGOT WAHL*: 12, 19, 32; KOBS*: 53, 57, 59f., 63–66; I2: 72
assumptions about the user WAHL*: 5f., 9, 11–15, 20, 27f., 32; I2: 70–73; CHIN*: 94, COHE*: 313; I4: 336ff.; KASS*: 408
ATMS RICH*: 47
attitude change JAME*: 287, 292
attribute salience MCCO*: 243–249, 251, 253
attribution of beliefs and goals KOBS*: 53f.

background COHE*: 313–317, 319–328, 331ff.
backward chaining FINI*: 428
Bayes' theorem RICH*: 47
Bayesian inference JAME*: 293, 303
belief
– discrepancy MCCO*: 235
– report KOBS*: 55, 57
– representation WAHL*: 7, 15ff.; KOBS*: 53, 58, 65, 67; QUIL*: 111–115; KASS*: 406
– revision WAHL*: 28, 33; JAME*: 285, 287; COHE*: 325–328
– types I1: 3; KOBS*: 53f., 56ff., 60–65, 67; CHIN*: 82
– verification QUIL*: 121, 131
beliefs WAHL*: 4f., 7–13, 16f., 26f., 29–33; KOBS*: 52ff., 57–65; I2: 70ff.; QUIL*: 109–112, 126, 129; CARB*: 136, 144ff., 151ff., 157; I3: 197; PARI*: 201, 206, 212; MCCO*: 236; I4: 339; SPAR*: 356, 362; MORI*: 376; KASS*: 391, 407, 409; FINI*: 415
– basic KOBS*: 56f., 59, 61
– collective WAHL*: 4, 29
– complex WAHL*: 32; KOBS*: 54, 56f., 60–63; JAME*: 277, 293
– contradictory WAHL*: 7; I2: 72; QUIL*: 115–121
– domain-specific QUIL*: 115, 123f.
– embedded, *see* beliefs, complex
– infinite-reflexive WAHL*: 7, 11; KOBS*: 54, 61ff.

beliefs
– inconsistent I4: 339
– meta- KOBS*: 61; I2: 71
– mutual WAHL*: 7, 11, 18, 31; KOBS*: 54, 61–64; CARB*: 152; JAME*: 277, 293f.
– saturated KOBS*: 56–59
– unsaturated KOBS*: 58
beneficiary SPAR*: 348
BGP-MS WAHL*: 18, 33; I4: 339
bias JAME*: 269, 309
– negative JAME*: 270, 291f.
black box KASS*: 388, 395
BORIS QUIL*: 112
bug library KASS*: 392, 400, 403f.
BUGGY KASS*: 386, 394f.
bugs KASS*: 392, 395, 403ff.

canonical user, *see* user, canonical
canonical user model, *see* user model, canonical
capabilities FINI*: 415
categories CHIN*: 79
certainty measures RICH*: 43, 46ff.
CGD COHE*: 313–317, 322f., 325f., 331f.
CHAMP I2: 73; LEHM*: 170
change types MORI*: 372–375
classification of stereotypes, *see* stereotypes, classification
closed world assumption CHIN*: 97f.; QUIL*: 121; CARB*: 151; I4: 339; FINI*: 424
CO-OP WAHL*: 19, 26; CHIN*: 100f.; MCCO*: 237f.
coherence MORI*: 382; PARI*: 203, 206
comment, in evaluation-oriented dialog
– definition JAME*: 261
– desirability JAME*: 262, 266f., 270f., 279, 282, 300
– representation in IMP JAME*: 264
communication module KASS*: 388
computer-aided instruction KASS*: 387
conflict resolution RICH*: 42f.
constituency schema I3: 196; PARI*: 205–208, 216–221, 228
consultation WAHL*: 5, 19f., 29; KOBS*: 59; CHIN*: 75, 77f., 83f., 89, 91, 105; QUIL*: 108; COHE*: 314–317; MORI*: 372
contents of user models I1: 3; KOBS*: 52–53; PARI*: 203; I4: 339
context KOBS*: 58, 60; CHIN*: 88, 90; MCCO*: 234, 244
context model CARB*: 134f., 138–146, 148ff., 152f., 155–161; COHE*: 313
– enriched CARB*: 152, 157, 160; COHE*: 313
contextual effects MCCO*: 234f., 242ff.
contextual information MCCO*: 236
contradiction FINI*: 429
– resolution I1: 3; RICH*: 41
conversational
– implicature JAME*: 272, 301
– roles WAHL*: 5, 18; CHIN*: 88; MORI*: 366f., 370, 375
– – in evaluation-oriented dialog JAME*: 258, 278
– setting WAHL*: 4f., 18, 30; I4: 337; MORI*: 367, 370f., 375, 383; MORI*: 367, 375, 383
cooperation SPAR*: 361
cooperative dialog CARB*: 137, 139, 152
cooperative response MCCO*: 235
cooperative system FINI*: 414
cooperativeness WAHL*: 4ff., 8, 18f., 28f.; I2: 70f.; CHIN*: 100; QUIL*: 109; PARI*: 212; MCCO*: 235; JAME*: 256, 261, 272
correction
– schemas MCCO*: 252
– strategies MCCO*: 234, 240
– response CARB*: 161

database WAHL*: 6, 9, 26, 29; PARI*: 203; MCCO*: 233–237; FINI*: 412, 418
– completed FINI*: 424
– deductive FINI*: 417
– distributed KOBS*: 63, 65
– query system FINI*: 418
de dicto KOBS*: 55, 57, 59, 61
de re KOBS*: 55
DEBUGGY KASS*: 392, 394ff.
decision properties JAME*: 294; I4: 336f.; SPAR*: 345f., 348–354, 356ff., 360
deep model KASS*: 395, 409
default
– assumptions WAHL*: 10; RICH*: 35, 45; COHE*: 317–323, 326f., 330–333
– inferences CHIN*: 76ff., 81, 88
– reasoning I1: 3; RICH*: 35, 41, 43, 47; CARB*: 153, 156, 158; I4: 339; FINI*: 411f., 416
– rules FINI*: 420
definitions in a stereotype RICH*: 45; FINI*: 421
Dempster-Shafer model RICH*: 47
description of beliefs and goals KOBS*: 54f., 57–62, 67
destination of modeling information SPAR*: 347
development change type MORI*: 374
deviation LEHM*: 165, 168, 170, 177f.
diagnostic model KASS*: 394, 406

dialog CARB*: 134, 137, 144, 150; MCCO*: 236, 242, 244ff.; SPAR*: 353f., 358, 361
– action-achieving MORI*: 368, 371, 376
– apprentice-expert, *see* dialog, expert-novice
– cooperative CARB*: 137, 139, 152
– contributions WAHL*: 6, 11–14, 19, 26, 32; KOBS*: 63ff.; I2: 70, 72; I3: 198
– expert-novice CARB*: 135, 139, 152
– evaluation-oriented I3: 198, JAME*: 258
– focus MCCO*: 236
– goals CARB*: 155
– information-seeking CARB*: 133–139, 159; MORI*: 367, 370, 375
– markers CARB*: 136, 156
– memory I4: 337; MORI*: 365f., 370
– mixed-initiative WAHL*: 5
– model RICH*: 39; SPAR*: 362; MORI*: 366, 376
– planning I1: 3; WAHL*: 6, 19–24, 31f.; KOBS*: 53, 60–65; I3: 196f.; I4: 336
– strategies PARI*: 200f., 206, 212, 221, 223, 230f.; COHE*: 313
– structure MORI*: 379
– task-oriented WAHL*: 5, 13; JAME*: 269
– update- MORI*: 368
differential modeling KASS*: 390–393, 398ff., 408
difficulty levels CHIN*: 79f., 87, 106
discourse, *see* dialog
disparate models CARB*: 158
disparate plans, *see* plans, disparate
domain
– goals CARB*: 136f.
– highlighting MCCO*: 248
– knowledge PARI*: 200–206, 211ff., 215f., 229, 231; COHE*: 313; FINI*: 415
– model MCCO*: 240, 248f.
– plans CARB*: 136
double stereotypes, *see* stereotypes, double
dynamic SPAR*: 344
– plan inference, *see* plan inference, dynamic
– property SPAR*: 344f., 348–351, 354, 361

educational diagnosis COHE*: 313–320, 322f., 329–333
effectiveness SPAR*: 346, 349ff., 354, 356f.
efficiency SPAR*: 347, 349ff., 356
effort, in evaluation-oriented dialog
– consequences for pragmatic interpretation JAME*: 285, 301f.
– contributing factors JAME*: 262, 279
– representation in IMP JAME*: 264
– rule in selection of comments JAME*: 261f., 267
ellipsis
– intersentential CARB*: 134, 145
– generation WAHL*: 22
– resolution WAHL*: 22
error messages JAME*: 257
evaluation criteria MORI*: 377–380, 382
evaluation standards
– complex beliefs concerning JAME*: 293
– definition JAME*: 261
– evidence concerning JAME*: 278, 293
– examples used by IMP JAME*: 294
– representation in IMP JAME*: 262
– uncertainty in JAME*: 293
evaluator JAME*: 258, 278
evidential reasoning FINI*: 416
expert PARI*: 200–206, 211ff., 223, 229; COHE*: 313
– model KASS*: 388–92, 395, 398–401, 404, 408
– module KASS*: 388
– systems WAHL*: 7; PARI*: 231; MCCO*: 233ff.; COHE*: 313–321, 325f., 329–333; I4: 340; KASS*: 388, 408; FINI*: 418
expertise WAHL*: 5, 15, 27; KOBS*: 52; I2: 70; CHIN*: 74f., 78–85, 90–94, 97, 106; CARB*: 160f.; I3: 196; PARI*: 200–206, 211–223, 228–231; COHE*: 313; I4: 337
– levels CHIN*: 83
– local PARI*: 214
explanation COHE*: 313–318, 320f., 325f., 329–333; FINI*: 415
– for mistaken belief QUIL*: 110, 115
– how to find QUIL*: 120f.
– organization QUIL*: 123
– patterns QUIL*: 115
– in story understanding QUIL*: 111
– verification of QUIL*: 116, 121
expressive power WAHL*: 14, 16ff.; KOBS*: 53, 65

failure as negation FINI*: 420, 427
flexibility
– increments LEHM*: 168
– level LEHM*: 169
focus MCCO*: 236, 244
– of attention CARB*: 137f., 140, 142ff., 146, 150, 155f., 158
– of dialog, *see* dialog focus
– of question JAME*: 278, 295

focused goal
– candidate CARB*: 137f., 140ff.
– current CARB*: 138ff., 147
focused plan
– candidate CARB*: 138–144
– current CARB*: 138–144
focusing CARB*: 139
– constraints CARB*: 139
– heuristics CARB*: 137, 139, 142ff., 150, 153, 156, 158
foreground MORI*: 382
forms LEHM*: 188
frames RICH*: 37
fuzzy logic CHIN*: 83

game playing WAHL*: 7, 8
generalization hierarchy RICH*: 37, 40f., 44, 46; CARB*: 150
generation system PARI*: 201ff., 206, 211f., 213, 221, 226, 230f.; MCCO*: 239
genetic graph KASS*: 401f.
Geometry Tutor KASS*: 393, 403ff., 409
glass box KASS*: 388
goal
– analysis CHIN*: 88, 103, 105
– recognition KOBS*: 64
– representation WAHL*: 17f.; KOBS*: 53, 65, 67; KASS*: 403–406
– types I1: 3; KOBS*: 53f., 59f., 63f., 67
goals WAHL*: 4f., 9–13, 16f., 20, 25ff., 29, 32f.; KOBS*: 53, 57, 59ff., 64f.; I2: 70ff.; CHIN*: 78, 88; QUIL*: 114, 116–120, 126, 129; PARI*: 201, 203, 206, 211, 230; MCCO*: 233, 236, 251; COHE*: 313–318, 320ff., 326ff., 330–333; I4: 339; MORI*: 376f., 382, 385; KASS*: 403f., 407, 409, 414
– basic KOBS*: 57, 59ff.
– complex KOBS*: 54, 57, 63f.
– embedded, *see* goals, complex
– non-primitive CARB*: 136
– saturated KOBS*: 59
– unsaturated KOBS*: 60
grammatical deviance (deviation) LEHM*: 163ff., 167
GREATERP KASS*: 388f., 393, 403ff., 409
GRUNDY WAHL*: 5, 11, 13f., 19f.; RICH*: 36, 41, 48; KOBS*: 53; CHIN*: 78, 80, 83; PARI*: 229; MORI*: 374f., 377; FINI*: 416
GUIDON KASS*: 386, 388f., 391, 394, 398ff., 402
GUMS WAHL*: 28, 33; KOBS*: 65f.; CHIN*: 106; I4: 339; FINI*: 412, 420

habitable interface, *see* interface, habitable
HAM-ANS WAHL*: 5, 19–22, 26, 30; RICH*: 36; KOBS*: 53, 63, 65f.; CHIN*: 78, 101; PARI*: 202; I4: 337f.; MORI*: 364f., 367–379, 383
help systems WAHL*: 7; QUIL*: 108, 126ff.; PARI*: 231; JAME*: 257, 307; FINI*: 411, 415
highlighted user model, *see* user model, highlighted
human-computer interaction WAHL*: 9; JAME*: 255, 307
hypothesize-and-revise paradigm CARB*: 152f.; COHE*: 313

ideal and buggy model KASS*: 404
ideal student, *see* student, ideal
IDEBUGGY KASS*: 394
idiosyncratic language LEHM*: 163, 165, 170, 176f., 183, 185
if-detected daemons CHIN*: 91
image maintenance JAME*: 275f., 280, 283, 285, 298, 301, 304–307, 310
IMP WAHL*: 20, 22, 25; I3: 198; JAME*: 258, 308f., 311
– behavior in employment interview situation JAME*: 258, 267f., 270, 273, 276, 283, 289f., 296f., 301
– plausibility of performance JAME*: 269
impasse KASS*: 405
implicature, *see* conversational implicature
impression
– change JAME*: 262, 264, 280, 282
– definition JAME*: 262
– empirical research JAME*: 269
– expected value JAME*: 264
– graphic representation JAME*: 265, 309
– uncertainty in JAME*: 262, 264
imprecise utterances CARB*: 151
indirect speech acts WAHL*: 6, 19; I2: 71; CHIN*: 85, 88, 103
informant JAME*: 258
information integration theory JAME*: 269
information-seeking dialog, *see* dialog, information-seeking
inheritance RICH*: 40, 43; CHIN*: 79
intelligent computer-aided instruction, *see* intelligent tutoring systems
intelligent tutoring systems WAHL*: 7, 28; KOBS*: 54; CHIN*: 77, 84, 89, 100; MCCO*: 237; I4: 338; KASS*: 386–390, 393, 395, 398, 408, 410; FINI*: 418
intended meaning CARB*: 146
intended recognition CARB*: 157f.

intentions CARB*: 136, 146f., 150, 153
interface
– habitable LEHM*: 164
– natural-language WAHL*: 4ff., 19, 26, 29f.; CHIN*: 75; KASS*: 407
– – learning LEHM*: 163, 165, 167ff., 171ff., 175–184, 187–193
intermediate person MORI*: 367
introspection FINI*: 425
isotopy MCCO*: 251; JAME*: 256; I4: 338; MORI*: 382f., 385
item JAME*: 261
– examples used by IMP JAME*: 263, 296
– representation in IMP JAME*: 263, 268, 281, 296, 300, 305
ITS, *see* intelligent tutoring systems

justification RICH*: 41, 47; MCCO*: 235f., 238–242

KAMP WAHL*: 5; KOBS*: 53, 59, 63–66; MORI*: 375
kernel grammar LEHM*: 164, 170, 172, 184
keyhole recognition CARB*: 157
KL-ONE FINI*: 422
KNOME CHIN*: I2: 70; 74–82, 85–88, 90–94, 97–107; QUIL*: 128; CARB*: 160f.
knowledge WAHL*: 5ff., 10ff.; RICH*: 37, 50; KOBS*: 55, 62ff., 67; I2: 70, 72; CHIN*: 74–78, 82–89, 91, 97–101, 104ff.; I3: 196; PARI*: 200–205, 211f., 214, 217f., 222, 229ff.; MCCO*: 235; COHE*: 313, 318f.; I4: 339; FINI*: 415, 426
– about domain, *see* domain knowledge
– bases FINI*: 412
– conceptual MORI*: 380
– difficulty levels CHIN*: 80f.
– meta- CHIN*: 74, 98f., 100–103
– referential MORI*: 380
– representation FINI*: 413
– semantic LEHM*: 165, 173, 175
– spectrum PARI*: 201f., 204f., 221; COHE*: 313
– syntactic LEHM*: 165, 167f., 173, 175f.
KODIAK CHIN*: 79f., 82, 91, 93, 99, 105

language stabilization LEHM*: 186, 190
LDC WAHL*: 30
learning RICH*: 48
– domain LEHM*: 176
– natural language, *see* interface, natural-language learning
least-deviant-first LEHM*: 165, 167ff., 178f., 182
letters of recommendation JAME*: 261, 276
likelihood ratings CHIN*: 85, 91–94, 97
linguistic particles WAHL*: 13
linguistics KOBS*: 54, 62; PARI*: 206, 224; JAME*: 262, 279f., 285
LISP CHIN*: 105
listener, in evaluation-oriented dialog JAME*: 258
LMS KASS*: 386, 392ff., 396ff., 405, 408f.
levels of expertise, *see* expertise levels
logic
– epistemic WAHL*: 16; KOBS*: 54, 58
– fuzzy, *see* fuzzy logic
– multi-valued FINI*: 425
long-term user model RICH*: 39

magnification, of benefits or costs of making a comment JAME*: 279f., 282
maintenance, of user model FINI*: 416
mal-rules MCCO*: 237; KASS*: 392, 396ff.
MAP CHIN*: 84
maxim
– of quality, revised CARB*: 159
– of relation CARB*: 146, 156
– of relevance CARB*: 134
MBR KOBS*: 59, 66
mental model WAHL*: 5, 9; PARI*: 203, 211f.; JAME*: 255, 307
– functional PARI*: 203
– conditions furthering accuracy JAME*: 310
– of system's dialog motivation JAME*: 256
– of system's user model JAME*: 256
meta-beliefs, *see* beliefs, meta-
meta-interpreter FINI*: 428
meta-knowledge, *see* knowledge, meta-
meta-rules CHIN*: 98
misattribution MCCO*: 236, 240f., 252
misclassification MCCO*: 234, 236, 238, 240
miscommunication CARB*: 152
misconceptions WAHL*: 7, 15, 26, 28; KOBS*: 52; I2: 71f.; CHIN*: 74f., 97f., 100–105; QUIL*: 109ff., 114–122, 125–131; CARB*: 150–153, 161; I3: 196f.; PARI*: 201; MCCO*: 233–238, 240–243, 249–252; COHE*: 313; KASS*: 395, 406f.
misgeneralization KASS*: 397
missing joins CARB*: 145, 147
modal verbs WAHL*: 13
model disparity CARB*: 159
model tracing KASS*: 404, 409
modeling factor SPAR*: 342, 351f.

modeling function SPAR*: 346–351, 354, 357, 360
modification change type MORI*: 373, 377
monolog JAME*: 260
most recently considered subgoal CARB*: 138ff., 142, 144
motivation, of dialog system JAME*: 256
multiple-agent planning systems WAHL*: 7; KOBS*: 54, 63, 65
MUMBLE I3: 197; MCCO*: 236
MYCIN RICH*: 47; COHE*: 329f.; KASS*: 388, 398

NAOS WAHL*: 22, 24
natural language LEHM*: 163, 169f.; SPAR*: 352f.; FINI*: 418
negation as failure I4: 339; FINI*: 424
negotiation CARB*: 152, 158f.
NEOMYCIN KASS*: 388f., 391, 398f.
non-decision properties I4: 336; SPAR*: 345–351, 353–358, 360, 362
norms, social JAME*: 275, 304f., 307
novel plans CARB*: 151
novice, see user, naive

object
– importance MCCO*: 244–247, 251
– misclassification WAHL*: 27
– perspective MCCO*: 233, 235f., 243–249, 251–254
– similarity MCCO*: 236, 239–246, 252ff.
– taxonomy MCCO*: 235f., 248
– in evaluation-oriented dialog JAME*: 258
objective facts about the user WAHL*: 13; KOBS*: 52f.; I2: 72; I4: 336; SPAR*: 344f., 348–351, 354, 362
obstacles in the user's plan WAHL*: 6
open world models CHIN*: 98
ORBIT JAME*: 260
over-answering WAHL*: 19; JAME*: 278, 282f.; MORI*: 370, 379f., 383, 385
overlay technique WAHL*: 7, 15; CHIN*: 77; KASS*: 390–393, 401f., 408

PAGAN CHIN*: 88
parsing LEHM*: 177, 180, 182
– adaptive CARB*: 161; LEHM*: 163ff., 167–173, 175–184, 188–193
– strategy LEHM*: 167f., 178
partition approach WAHL*: 17
path expansion heuristics CARB*: 148ff.
patient role I4: 336f.; SPAR*: 343f., 346–354, 356, 360f.
personality traits WAHL*: 13, 15
perspective RICH*: 36; CARB*: 161; MCCO*: 233
perturbation modeling KASS*: 390, 392f., 395f., 408f.
plan
– body CARB*: 136
– disparity CARB*: 135, 152, 155–160; COHE*: 313
– effects CARB*: 136
– hierarchical CARB*: 135f.
– identification heuristics CARB*: 137
– inference KASS*: 403
– – dynamic CARB*: 134, 137, 144, 150
– recognition FINI*: 414
– representation KASS*: 404f.
planfor CHIN*: 82
planning failures QUIL*: 113, 123, 125ff., 130
plans WAHL*: 5–8, 12, 27, 31; KOBS*: 52ff.; I2: 70ff.; CHIN*: 78, 82, 87ff., 101; QUIL*: 109–121, 127ff.; CARB*: 136, 147–161; PARI*: 201; COHE*: 313; I4: 339; KASS*: 403, 407ff.; FINI*: 414
– disparate CARB*: 152, 154
– invalid CARB*: 151, 153
– organization QUIL*: 122ff., 131
– retrieval QUIL*: 124
politeness JAME*: 256
posited superordinate MCCO*: 239f.
possible-world approach WAHL*: 7, 16
pragmatic interpretation JAME*: 272
– anticipation by speaker JAME*: 272, 274ff., 295f., 299, 304f.
– consequences for impressions of listener JAME*: 273, 276, 285f., 291f., 298
– in human-computer interaction JAME*: 309f.
– relation to conversational implicature JAME*: 272
pragmatically ill-formed queries CARB*: 145f., 148, 161
pragmatically ill-formed utterances CARB*: 134, 145
preconditions CARB*: 136, 142; CARB*: 147f.
predicates PARI*: 206–209, 216, 223ff., 228
predictions I1: 2; CHIN*: 77f., 80, 92, 104
– from user model I1: 2f.; RICH*: 35, 49; KASS*: 396, 402
preferences COHE*: 317f., 320ff., 326–330; FINI*: 415

preparatory expressions JAME*: 280
– dependent on projected ascriptions JAME*: 296
– conditions for use JAME*: 280, 285
– dependent on projected ascriptions JAME*: 295, 300f., 304f.
– pragmatic functions JAME*: 280, 302
presumption failures MCCO*: 237
presuppositions WAHL*: 26; KOBS*: 57f.
previous discourse MCCO*: 234f.
privacy WAHL*: 11, 14, 27; KOBS*: 64
projected ascriptions JAME*: 294ff., 298–305, 307
projected bias JAME*: 274–277, 286ff., 290–294, 298, 302, 307
PROLOG FINI*: 412, 430
process schema I3: 196; PARI*: 205, 208ff., 217–221, 228
property substitution heuristic CARB*: 147f.
protagonist model MORI*: 366
prototypes CHIN*: 79
PROUST KASS*: 393, 403f.
psychological model KASS*: 397, 399, 401–404, 406f., 410

question answering WAHL*: 4, 6, 9, 14, 19, 25; I2: 70; PARI*: 200–203; COHE*: 313; I4: 337
questions JAME*: 282
– attention focusing JAME*: 278, 282
– wh- WAHL*: 11, 13; KOBS*: 57f., 60; COHE*: 316f., 326ff.
– why- COHE*: 316f., 326ff.
– y/n- KOBS*: 57f., 60, 62, 64; JAME*: 282

RABBIT RICH*: 36
real estate agent CHIN*: 78, 83
recommendations WAHL*: 19; KOBS*: 52; MORI*: 379
recovery
– strategies LEHM*: 165, 168, 177f., 179
– actions LEHM*: 168ff., 177
referential knowledge, *see* knowledge, referential
relevance CARB*: 146f., 150, 155
repair theory KASS*: 403, 405, 409
replacement change type MORI*: 373
representation of user models, *see* user models, representation
resource-limited processes WAHL*: 4, 31
response CARB*: 158; MCCO*: 233–241, 249, 251–254
– cooperative QUIL*: 109ff., 126, 130
– schemas MCCO*: 236
– strategies MCCO*: 233, 236, 238–242, 253f.
– tailoring QUIL*: 128, 130
revision change type MORI*: 373f., 376
revision of user models I4: 337, 373f., 376
rhetorical predicates PARI*: 206
rigid designator KOBS*: 58
roles, conversational, *see* conversational roles
ROMPER WAHL*: 27; QUIL*: 126; MCCO*: 236, 242f., 251f.

salience of attributes I3: 197; MCCO*: 246
SC WAHL*: 5; QUIL*: 128
SC-UM CHIN*: 92
schemas PARI*: 206ff., 211–220, 223–226, 228, 231; COHE*: 324ff., 332f.
search LEHM*: 180, 182
security WAHL*: 11, 14, 27
self-bounded language LEHM*: 170, 182, 185, 188, 190
shell systems, *see* user modeling shell systems
short-term user model, *see* user model, short-term
shortcomings of UM components I1: 2; CHIN*: 105, 107; JAME*: 256; I4: 336
silence JAME*: 264, 266, 284, 301, 305
similarity CARB*: 147, 150
– metric MCCO*: 243f., 247, 249, 253f.
similar object MCCO*: 234
situation KOBS*: 56f., 59, 61
skill lattice KASS*: 394f.
social implications WAHL*: 4, 11, 14, 27; JAME*: 255, 307
social psychology JAME*: 269, 276, 285, 287, 292ff., 298, 308
SOPHIE KASS*: 389
source
– of misconception MCCO*: 233, 235, 239
– of modeling information SPAR*: 347
speech act MORI*: 371f., 375ff., 379, 383
– generation MORI*: 379, 385
– recognition I1: 3; KOBS*: 59, 65
SPIRIT QUIL*: 112, 127, 131
standard name KOBS*: 58
static property SPAR*: 344f., 348–351, 354, 361
stereotype
– activation RICH*: 38, 46
– adaptation I1: 3; RICH*: 43, 48
– approach I1: 2f.; QUIL*: 128; MORI*: 373; JAME*: 256
– hierarchy RICH*: 37, 40f., 44, 46
– techniques, *see* stereotype approach

stereotypes I1: 2; WAHL*: 13, 20f.; RICH*: 35–39, 41, 43–50; I2: 70f.; CHIN*: 74, 78ff., 104f.; CARB*: 161; PARI*: 200ff., 229; COHE*: 313, 317ff., 325–331; I4: 339; SPAR*: 356, 363; MORI*: 377ff.; FINI*: 414, 418ff.
– classification of FINI*: 422
– double CHIN*: 74f., 79f., 105f.
strategies LEHM*: 177; PARI*: 200f., 205, 208, 211–216, 218–226, 228, 230f.; COHE*: 313
student
– ideal KASS*: 389, 404
– model WAHL*: 7; COHE*: 318–322, 329–332; KASS*: 386–410
– modeling I4: 338
subjective property SPAR*: 344f., 348–351, 354, 361f.
submodel SPAR*: 351f., 354
substitution heuristics CARB*: 147, 150
support for misconception MCCO*: 234, 238ff., 242, 254
syntactic knowledge, *see* knowledge, syntactic
system canonical form LEHM*: 170, 175ff., 188

TAILOR CHIN*: 107; CARB*: 161; I3: 197
talkativeness JAME*: 266, 271, 277
task MORI*: 367
task domain LEHM*: 173, 175
task-oriented dialog, *see* dialog, task-oriented
task-related goal CARB*: 136
task-related plan CARB*: 134, 138f., 144, 147–152, 159, 161
taxonomy
– of beliefs and goals I1: 3; KOBS*: 52f., 65, 67
– of potential explanations QUIL*: 116–122, 128, 130f.
TEAM WAHL*: 30
TEIRESIAS CHIN*: 98
terminology MORI*: 364f., 374
text PARI*: 205
– comprehension WAHL*: 7; QUIL*: 109, 129
– generation WAHL*: 7
– natural language PARI*: 204
– naturally occurring PARI*: 201, 206, 211, 213
THEORIST COHE*: 322f., 332f.
theory of meaning CARB*: 134, 146
TMS, *see* truth maintenance system
topic
– common JAME*: 279, 281
– conceptualization of JAME*: 278, 281
– importance of JAME*: 279, 282f.
– of question JAME*: 278, 283
TRACK WAHL*: 12; KOBS*: 59f., 66; CARB*: 134, 136, 140, 142, 144; MCCO*: 251
transmutable systems WAHL*: 30f.
transportable systems WAHL*: 30
triggers RICH*: 37, 40, 43, 46f., 49; FINI*: 422
TRUMP WAHL*: 28, 33
truth maintenance system RICH*: 42, 47; FINI*: 429
tutoring component KASS*: 388
tutoring systems QUIL*: 116, 125, 131

UC WAHL*: 5, 19; RICH*: 36, 51; KOBS*: 60, 66; CHIN*: 74ff., 78f., 81f., 85ff., 89–94, 97–106; QUIL*: 128; CARB*: 161; PARI*: 202
UCEgo CHIN*: 75, 102, 104
UCExpress CHIN*: 103f.
UCTeacher CHIN*: 88
ULLY WAHL*: 5, 18
UMFE WAHL*: 15; RICH*: 36; KOBS*: 53, 65f.; PARI*: 229; KASS*: 408; FINI*: 416
uncertainty KOBS*: 53, 56ff., 61, 63; CHIN*: 77, 82, 91–94; LEHM*: 169, 172; I3: 198; FINI*: 426
UNIX CHIN*: 74f., 78, 81, 89, 106
unprovability FINI*: 424
unsolicited dialog contributions WAHL*: 14
update dialog MORI*: 368
USCSH CHIN*: 78
user
– adaptability LEHM*: 184, 191
– canonical MCCO*: 236
– canonical form LEHM*: 171f., 175, 188
– frequent LEHM*: 164, 170, 183
– naive PARI*: 201f., 204ff., 211ff., 229; COHE*: 313
– model WAHL*: 6, 9; RICH*: 38; QUIL*: 124, 128; CARB*: 134, 139, 144, 150, 160f.; LEHM*: 171, 192; PARI*: 202–205, 212–221, 223, 226, 228ff.; MCCO*: 235f., 238, 254; COHE*: 313f., 316–321; 325–333; MORI*: 366; KASS*: 386, 406f., 409f.
– – acquisition, *see* acquisition of user models
– – canonical RICH*: 38f.
– – highlighted MCCO*: 233f., 236, 239–243, 253f.

– – individual RICH*: 38ff.; CHIN*: 79, 81, 83; LEHM*: 164; FINI*: 414
– – initial WAHL*: 10f., 14; QUIL*: 110, 129
– – representation WAHL*: 14–17, 24ff.; KOBS*: 53, 65, 67; CHIN*: 80, 99; I4: 339
– – short-term RICH*: 39
– modeling
– – component I1: 3; KOBS*: 53, 67; WAHL*: 4, 6, 10; KOBS*: 59, 63; QUIL*: 114; I4: 339
– – dynamic MORI*: 374
– – shell systems I1: 3; WAHL*: 33; KOBS*: 53, 65; CHIN*: 106; I4: 339; FINI*: 412
– – static MORI*: 374
– property SPAR*: 344ff., 348–351, 353, 356ff., 362
– role SPAR*: 343f., 346ff., 350, 360

vagueness JAME*: 261, 271, 273, 276, 289, 291f., 302
valid plan assumption CARB*: 151
VIE-DPM WAHL*: 11, 32; KOBS*: 53, 57–60, 62–66; CHIN*: 85; MORI*: 375

wants, *see* goals
WEST KASS*: 389, 391
wh-questions WAHL*: 11
WUSOR KASS*: 386, 388, 390f., 394, 400ff.

XCALIBUR WAHL*: 30
XTRA WAHL*: 5, 19

List of Contributors

Dr. Sandra Carberry

Dept. of Computer Science, Univ. of Delaware, Newark, Delaware 19711, USA

Sandra Carberry received a BA degree in Mathematics from Cornell University and an MS degree in Computer Science from Rice University, and was a Member of Technical Staff at Bell Telephone Laboratories. She is now an Assistant Professor of Computer and Information Sciences at the University of Delaware, where she received her Ph.D. in 1985. Portions of her contribution are based on this research. Her current interests are robust models of plan recognition and the application of user models to the generation of helpful responses.

Dr. Jaime G. Carbonell

Carnegie-Mellon Univ., Dept. of Computer Science, Schenley Park, Pittsburgh, PA 15213, USA

Jaime G. Carbonell is Professor of Computer Science at Carnegie-Mellon University, founder and director of Carnegie Group Inc. and director of the Center for Machine Translation at CMU. He received his B.S. degrees in Physics and in Mathematics from MIT and his M.S. and Ph.D. degrees in Computer Science from Yale University. Dr. Carbonell has worked in various areas of artificial intelligence, including machine learning, natural language processing, planning and problem solving, machine translation, analogical reasoning and knowledge representation. He has authored over a hundred technical papers and has edited several books.

Dr. David N. Chin

Department of Information and Computer Sciences, University of Hawaii at Manoa, 2565 The Mall, Honolulu, HI 96822, USA

David N. Chin is Assistant Professor of Information and Computer Sciences at the University of Hawaii at Manoa. He conducted his Ph.D. studies at the University of California at Berkeley, where he worked on treating UC, the UNIX Consultant, as an intelligent agent which has its own goals, plans, and model of the user. His contribution summarizes the user modeling portion of his dissertation work.

Dr. Robin Cohen

Dept. of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L3G1

Robin Cohen is Assistant Professor of Computer Science at the University of Waterloo, working in the area of natural language understanding. She completed her Ph.D. in 1983 at the University of Toronto, under the supervision of Ray Perrault, in the area of natural language pragmatics and discourse. The joint research with Marlene Jones described in this volume reflects an interest in investigating the development of specific explanations (discourse) to users based on their background and goals (pragmatics).

Jill Fain Lehman

Carnegie-Mellon Univ., Dept. of Computer Science, Schenley Park, Pittsburgh, PA 15213, USA

Jill Fain Lehman is a Ph.D. candidate in Computer Science at Carnegie-Mellon University. She received her B.S. degree in Computer Science from Yale University in 1981, and her M.S. degree in the same field from Carnegie-Mellon in 1987. Her primary interests lie in the areas of natural language processing and machine learning, although previous research has included work in expert systems for the RAND Corporation. Her paper summarizes the current state of her thesis work on adaptive parsing.

Dr. Timothy Finin

UNISYS, Paoli Research Center, Route 252 and Central Ave., Paoli, PA 19311, USA

Timothy W. Finin is a technical director at the UNISYS Paoli Research Center. He received a master's degree in Electrical Engineering from M.I.T., and MSc. and Ph.D. degrees in Computer Science from the University of Illinois at Urbana-Champaign. He did research at M.I.T., in the area of robotics and computer vision, and was an Assistant Professor in the Department of Computer and Information Science at the University of Pennsylvania from 1980 to 1987. His current research interests include knowledge representation, expert systems and computational linguistics, as well as their applications to intelligent user interfaces. His paper describes recent work on creating a user modeling shell in order to support the development of various intelligent interactive applications.

Anthony Jameson

Psychologisch Laboratorium, Katholieke Universiteit Nijmegen, Montessorilaan 3, Postbus 9104, NL-6500 HE Nijmegen, Netherlands

Anthony Jameson is affiliated with the Department of Experimental Psychology at the University of Nijmegen, The Netherlands. He is writing a dissertation on psychological and normative models of the way people ascribe knowledge and values to others in everyday interaction. The present paper incorporates some of the theoretical ideas that have emerged from this research, while building on work done several years ago in connection with the HAM-ANS project at the University of Hamburg, Fed. Rep. of Germany.

Dr. Marlene Jones

Alberta Research Council, 6815 8th Street NE, 3rd floor, Calgary, Alberta, Canada T2E 7H7

Marlene Jones is a senior researcher in Artificial Intelligence in the Alberta Research Council's Department of Advanced Technologies and an adjunct professor in the Department of Computational Science at the University of Sakatchewan. Prior to joining the Alberta Research Council, Dr. Jones was a tenured Associate Professor at the University of Waterloo. Much of her research concernes AI applications within the field of education, including expert systems for educational diagnosis, expert environments for curriculum and course development, and user modeling.

Dr. Robert Kass

Center for Machine Intelligence, 2001 Commenwealth Blvd., Ann Arbor, MI 48105, USA

Robert Kass is a Senior Research Engineer at the Center for Machine Intelligence in Ann Arbor, Michigan. He recently completed his Ph.D. at the Department of Computer and Information Science at the University of Pennsylvania, where his research focused on the implicit acquisition of user models from users' interactions with an advisory expert system. His contribution summarizes some of his research in representing and acquiring agent models.

Dr. Alfred Kobsa

SFB 314: AI - Knowledge-Based Systems, Dept. of Computer Science, University of Saarbrücken, D-6600 Saarbrücken 11, Fed. Rep. of Germany

Alfred Kobsa received master degrees in Computer Science and the Social and Economic Sciences from the University of Linz, Austria, and his Ph.D. in Computer Science from the Technical University of Vienna. He has also pursued non-degree studies in Cognitive Psychology, Linguistics, Formal Logic and the Philosophy of Science at the University of Salzburg. Since 1985 he has been a senior researcher and project coordinator of the XTRA project in the German Collaborative Research Program on AI and Knowledge-Based Systems at the University of Saarbrücken, where he works in the fields of user modeling, knowledge representation and expert systems. His contribution is based on experiences gained both from his Ph.D. work as well as his current research.

Dr. Kathleen McCoy

Dept. of Computer Science, Univ. of Delaware, Newark, Delaware 19711, USA

Kathleen McCoy is Assistant Professor of Computer and Information Sciences at the University of Delaware. She received her Ph.D. in computer science from the University of Pennsylvania in 1985. Portions of her contribution are based on this Ph.D. research. Her current interests center on contextual effects of previous discourse on representation of user models and generation of subsequent responses.

Dr. Katharina Morik

Technical University of Berlin, FR-5-8, Franklinstr. 28/29, D-1000 Berlin 10, Fed. Rep. of Germany

Katharina Morik is internal project leader of the KIT-Lerner project at the Technical University of Berlin, working in the fields of knowledge acquisition and machine learning. She completed her Ph.D. thesis on belief systems and natural language understanding at the University of Hamburg in 1981. From 1982 to 1984 she worked on the Hamburg Application-Oriented Natural-Language System (HAM-ANS) with a special interest in user modeling and dialog strategy. Her contribution is based on experiences gained from this work.

Dr. Cécile L. Paris

Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292-6695, USA

Cécile Paris received her Bachelor degree from the University of California in Berkeley and has just completed her Ph.D. in Computer Science at Columbia University (New York). She is now working at the Information Sciences Institute, continuing her research on natural language generation and user modeling. Her thesis work has focused on how a system should tailor a response depending on how much the user knows about the domain under consideration. Her paper reports major results from this work.

Alexander E. Quilici

University of California, Computer Science Dept., Artificial Intelligence Lab., 3531 Boelter Hall, Los Angeles, CA 90024, USA

Alex Quilici is currently completing his Ph.D. studies at the University of California, Los Angeles. His research concerns the automatic detection and correction of plan-oriented misconceptions that often occur in argumentative dialogs. His contribution summarizes aspects of this research dealing with the misconceptions of novice computer users. His research interests include user modeling, intelligent tutoring systems and models of argumentation.

Dr. Elaine Rich
MCC, 9430 Research Boulevard, Austin, TX 78759, USA

Elaine Rich is a director of the artificial intelligence lab at Microelectronics and Computer Technology Corporation (MCC), where she has been working primarily in the area of knowledge-based natural language interfaces. Prior to joining MCC, she was an assistant professor of Computer Sciences at the University of Texas at Austin. Her paper surveys the use of stereotypes in user modeling, which was the topic of her Ph.D. dissertation work.

Dr. Karen Sparck Jones
University of Cambridge, Computer Laboratory, Corn Exchange Street, Cambridge CB2 3QG, England

Karen Sparck Jones is an Assistant Director of Research at the University of Cambridge Computer Laboratory. She has worked on language and information processing problems since the late fifties, concentrating in recent years on inquiry interfaces and hence on the meaning representations and entity models these require.

Dr. Wolfgang Wahlster
SFB 314: AI - Knowledge-Based Systems, Dept. of Computer Science, University of Saarbrücken, D-6600 Saarbrücken 11, Fed. Rep. of Germany

Wolfgang Wahlster received the diploma and the doctoral degree in Computer Science from the University of Hamburg, West Germany. He was a senior researcher and coordinator in the HAM-RPM and HAM-ANS natural language dialog projects, and is now Professor of Artificial Intelligence and Database Systems and head of the AI Laboratory at the University of Saarbrücken as well as director of the Saarbrücken branch of the German AI Center. His current research includes multimodal input and output, natural language access systems, user modeling, natural language generation and explanation components for expert systems.

N.J. Nilsson: Principles of Artificial Intelligence. XV, 476 pages, 139 figs., 1982

J.H. Siekmann, G. Wrightson (Eds.): Automation of Reasoning 1. Classical Papers on Computational Logic 1957–1966. XXII, 525 pages, 1983

J.H. Siekmann, G. Wrightson (Eds.): Automation of Reasoning 2. Classical Papers on Computational Logic 1967–1970. XXII, 638 pages, 1983

L. Bolc (Ed.): The Design of Interpreters, Compilers, and Editors for Augmented Transition Networks. XI, 214 pages, 72 figs., 1983

M.M. Botvinnik: Computers in Chess. Solving Inexact Search Problems. With contributions by A.I. Reznitsky, B.M. Stilman, M.A. Tsfasman, A.D. Yudin. Translated from the Russian by A.A. Brown. XIV, 158 pages, 48 figs., 1984

L. Bolc (Ed.): Natural Language Communication with Pictorial Information Systems. VII, 327 pages, 67 figs., 1984

R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.): Machine Learning. An Artificial Intelligence Approach. XI, 572 pages, 1984

A. Bundy (Ed.): Catalogue of Artificial Intelligence Tools. Second, revised edition. IV, 168 pages, 1986

C. Blume, W. Jakob: Programming Languages for Industrial Robots. XIII, 376 pages, 145 figs., 1986

J.W. Lloyd: Foundations of Logic Programming. Second, extended edition. XII, 212 pages, 1987

L. Bolc (Ed.): Computational Models of Learning. IX, 208 pages, 34 figs., 1987

L. Bolc (Ed.): Natural Language Parsing Systems. XVIII, 367 pages, 151 figs., 1987

N. Cercone, G. McCalla (Eds.): The Knowledge Frontier. Essays in the Representation of Knowledge. XXXV, 512 pages, 93 figs., 1987

G. Rayna: REDUCE. Software for Algebraic Computation. IX, 329 pages, 1987

D.D. McDonald, L. Bolc (Eds.): Natural Language Generation Systems. XI, 389 pages, 84 figs., 1988

L. Bolc, M.J. Coombs (Eds.): Expert System Applications. IX, 471 pages, 84 figs., 1988

C.-H. Tzeng: A Theory of Heuristic Information in Game-Tree Search. X, 107 pages, 22 figs., 1988

Springer Series
SYMBOLIC COMPUTATION – *Artificial Intelligence*

H. Coelho, J.C. Cotta: Prolog by Example. How to Learn, Teach and Use It. X, 382 pages, 68 figs., 1988

L. Kanal, V. Kumar (Eds.): Search in Artificial Intelligence. X, 482 pages, 67 figs., 1988

H. Abramson, V. Dahl: Logic Grammars. XIV, 234 pages, 40 figs., 1989

R. Hausser: Computation of Language. An Essay on Syntax, Semantics, and Pragmatics in Natural Man-Machine Communication. XVI, 425 pages, 1989

P. Besnard: An Introduction to Default Logic. XI, 201 pages, 1989

A. Kobsa, W. Wahlster (Eds.): User Models in Dialog Systems. XI, 471 pages, 113 figs., 1989

Math

QA76.9 H85U74 1989

User models in dialog systems.

15697721

MATHEMATICS LIBRARY

A0000156977 21

SUBJECT TO RECALL

DATE DUE

2 1990

Demco, Inc. 38-293

APPENDIX WAHLSTER01

```
leader  05162cam a2200385Ia 4500
001     9947469793406236
005     20200824163906.0
008     060804s2006    gw a     h    000 0 eng d
016     7#$a973029803 $2GyFmDB
020     ##$a3540237321 (hd.bd.)
024     3#$a9783540237327
028     0#$aOHX $bhar040151993
029     1#$aYDXCP $b2196881
035     ##$a(OCoLC)ocm78840488
035     ##$a4746979
035     ##$a(ppiu)4746979-pittdb
035     ##$a(pittdb)4746979-pittdb
040     ##$aOHX $cOHX $dBAKER $dUOH $dYDXCP $dIXA $dOrLoB-B $dUtOrBLW
049     ##$aPITT
050     #4$aQA76.9.N38 $b543 2006
072     #7$a7K $2lcco
245     00$aSmartKom: $bfoundations of multimodal dialogue systems / $cWolfgang Wahlster (ed.).
260     ##$aBerlin ; $aNew York : $bSpringer, $c2006.
300     ##$axviii, 643 p. : $bill. ; $c24 cm.
440     #0$aCognitive technologies, $x1611-2482
500     ##$aResearch report.
504     ##$aIncludes bibliographical references.
505     00$tDialogue systems go multimodal : the SmartKom experience / $rWolfgang Wahlster --  $tFacts and figures about the SmartKom project / $rAnselm Blocher
650     #0$aNatural language processing (Computer science)
650     #0$aHuman-computer interaction.
650     #0$aAutomatic speech recognition.
700     1#$aWahlster, Wolfgang.
906     ##$aULS
909     ##$ uls0002342
994     ##$aC8 $bPIT
```

University of Pittsburgh | PittCat

NEW SEARCH BROWSE JOURNAL FINDER DATABASE FINDER Sign in Menu

Search anything Books, Articles, and More ADVANCED SEARCH

ASK US

BOOK

SmartKom: foundations of multimodal dialogue systems

Wolfgang Wahlster (ed.)
2006
xviii, 643 p. : ill. ; 24 cm.

Not Available

Available Online

Report a Problem

Chapters of this book (37)

TOP
SEND TO
VIEW ONLINE
GET IT
DETAILS

Send to

CITATION EMAIL ENDNOTE/ZOTERO PRINT PERMALINK

View Online

Full text availability

SpringerLink Books - AutoHoldings

Get It

Please sign in to check if there are any request options. Sign In

BACK TO LOCATIONS

LOCATION ITEMS

LILS - Thomas Blvd
Out of library : Stacks (Request This Item) ; QA76.9.N38 S63 2006
(1 copy, 0 available, 1 request)

On hold shelf until 04/16/2025

Details

Title	SmartKom: foundations of multimodal dialogue systems
Contributor	Wahlster, Wolfgang
Language	English
Publisher	Berlin ; New York : Springer
Date	2006
Type	Book
Description	xviii, 643 p. : ill. ; 24 cm.
Series	Cognitive technologies, 1611-2482
Notes	Research report. Includes bibliographical references.

Contents	Dialogue systems go multimodal : the SmartKom experience / Wolfgang Wahlster – Facts and figures about the SmartKom project / Anselm Blocher – exemplary interaction with SmartKom / Norbert Reithinger and Gerd Herzog – SmartKom architecture : a framework for multimodal dialogue systems / Gerd Herzog and Norbert Reithinger – Modeling domain knowledge : know-how and know-what / Iryna Gurevych, Robert Porzel and Rainer Malaka – Speech recognition / Andre Berton, Alfred Kaltenmeier, Udo Haiber and Olaf Schreiner – Class-based language model adaptation / Martin C. Emele, Zica Valsan, Yin Hay Lam and Silke Gorovoy – dynamic lexicon / Silke Gorovoy, Stefan Rapp and Martin Emele – prosody module / Viktor Zeissler, Johann Adelhardt, Anton Batliner, Carmen Frank, Elmar Noth, Rui Ping Shi and Heinrich Niemann – sense of vision : gestures and real objects / Jens Kacky, Michael Lutzeler and Hans Rottger – facial expression module / Carmen Frank, Johann Adelhardt, Anton Batliner, Elmar Noth, Rui Ping Shi, Viktor Zeissler and Heinrich Niemann – Multiple biometrics / Stephan Grashey and Matthias Schuster – Natural language understanding / Ralf Engel – gesture interpretation module / Rui Ping Shi, Johann Adelhardt, Anton Batliner, Carmen Frank, Elmar Noth, Viktor Zeissler and Heinrich Niemann – Modality fusion / Ralf Engel and Norbert Pfleger – Discourse modeling / Jan Alexandersson and Norbert Pfleger – Overlay : the basic operation for discourse processing / Jan Alexandersson, Tilman Becker and Norbert Pfleger – In context : integrating domain- and situation-specific knowledge / Robert Porzel, Iryna Gurevych and Rainer Malaka – Intention recognition / Jurgen te Vrugt and Thomas Portele – Plan-based dialogue management for multiple cooperating applications / Markus Lockelt – Emotion analysis and emotion handling subdialogues / Michael Streit, Anton Batliner and Thomas Portele – Problematic, indirect, affective, and other nonstandard input processing / Michael Streit – Realizing complex user wishes with a function planning module / Sunna Torge and Christian Hying – Intelligent integration of external data and services into SmartKom / Hidir Aras, Vasu Chandrasekhara, Sven Kruger, Rainer Malaka and Robert Porzel – Multimodal fission and media design / Peter Poller and Valentin Tschernomas – Natural language generation with fully specified templates / Tilman Becker – Multimodal speech synthesis / Antje Schweitzer, Norbert Braunschweiler, Grzegorz Dogil, Tanja Klankert, Bernd Mobius, Gregor Mohler, Edmilson Morais, Bettina Sauberlich and Matthias Thomae – Building multimodal dialogue applications : system integration in SmartKom / Gerd Herzog and Alassane Ndiaye – SmartKom-English : from robust recognition to felicitous interaction / David Gelbart, John Bryant, Andreas Stolcke, Robert Porzel, Manja Baudis and Nelson Morgan – SmartKom-public / Axel Horndasch, Horst Rapp and Hans Rottger – SmartKom-home : the interface to home entertainment / Thomas Portele, Silke Gorovoy, Martin Emele, Andreas Kellner, Sunna Torge and Jurgen te Vrugt – SmartKom-mobile : intelligent interaction with a mobile system / Rainer Malaka, Jochen Haussler, Hidir Aras, Matthias Merdes, Dennis Pfisterer, Matthias Jost and Robert Porzel – SmartKom-mobile car : user interaction with mobile services in a car environment / Andre Berton, Dirk Buhler and Wolfgang Minker – Wizard-of-Oz recordings / Florian Schiel and Ulli Turk – Annotation of multimodal data / Silke Steininger, Florian Schiel and Susan Rabold – Multimodal emograms, data collection and presentation / Johann Adelhardt, Carmen Frank, Elmar Noth, Rui Ping Shi, Viktor Zeissler and Heinrich Niemann – Empirical studies for intuitive interaction / Iryna Gurevych and Robert Porzel – Evaluation of multimodal dialogue systems / Florian Schiel
Subject	Natural language processing (Computer science) ; Human-computer interaction ; Automatic speech recognition ;
Related titles	Available in other form: SmartKom : foundations of multimodal dialogue systems ;
Identifier	ISBN : 3540237321 (hd.bd.) OCLC : (OCoLC)com70849486
Derlup Member kis	9998547774406236
Source	Library Catalog

Virtual Browse

Natural language at | Semantic structures | SmartKom : foundations | Automatic development | Pr... rin

the computer proceedings	advances in natural language	of multimodal dialogue	t : a domain-specific aspect	lar pr a
c1988	1989	2006	2010	ja

Links

Find this item in Google Books
Display Source Record

Syndetics Unbound

Summary
Also Available As
Browse Shelf
Look Inside
Tags
Reader Reviews
Book Profile

SUMMARY

With contributions by leading scientists in the field, this book gives the first comprehensive overview of the results of the seminal SmartKom project - one of the most advanced multimodal dialogue systems worldwide.

LOOK INSIDE

Part I Introduction

Dialogue Systems Go Multimodal: The SmartKom Experience Wolfgang Wahlster p. 3

Facts and Figures About the SmartKom Project Anselm Blocher p. 29

SEE FULL TABLE OF CONTENTS COVER

READER REVIEWS

BE THE FIRST TO WRITE A REVIEW

ALSO AVAILABLE AS

Book

TAGS

computer science **natural language processing**

BROWSE TAGS

BOOK PROFILE

Person
Schwarzenegger, Arnold; Kidman, Nicole; Bryant, John; Tschira, Klaus

Topics
Software; Artificial intelligence; Geographic information systems; Laboratories; Linguistics; Syntax; Man machine interaction

Location
Germany; New York; Canada; United Kingdom--UK; Spain; Japan; Europe; Netherlands; Switzerland; Denmark

Organization
Association for Computational Linguistics; DaimlerChrysler; MIT Press; Cambridge University Press; University of Stuttgart; Siemens AG; University of Pennsylvania;

Massachusetts Institute of Technology; Prentice Hall Inc; Qualcomm Inc

BROWSE PROFILE

BROWSE SHELF

QA76.9.N38 S23 2014 QA76.9.N38 S78 1999

SEE FULL SHELF

Catalog enrichment powered by Syndetics Unbound

APPENDIX WAHLSTER02

3 1735 054 472 026

ang Wahlster (Ed.)

SmartKom: Foundations of Multimodal Dialogue Systems

Springer

Wahlster (Ed.)

SmartKom: Foundatio
Multimodal Dialogue

HILL
QA76.9
N38S63
2006

COGNITI

The series Cognit
subfields such as
robotics, automa
practice, and kn
science, artifici
The series inclu
graphs and mul
researchers, an

Wahlster

Smart
Multim

The result of four
involving 12 part
advanced multim
history of intellig

The system provi
with an embodie
nents are used in
practical foundat
systems that dea
cover the full spe

With contributio
hensive overview

ISBN 3-540
9 783540

springer.c

Do Not Remove This Wrapper

R14-M17-S08-T12
31735054472026
Tray ID: R14-M17-S08-T12
Call No.: QA76.9.N38 S83 2006
Volume:
Collection: GEN
Title: SmartKom: foundations of multi

Banding Label 04/09/2025 12:42:42

CIRCULATING
CIRCULATING
CIRCULATING
CIRCULATING

Do Not Remove This Wrapper

UNIVERSITY OF PITTSBURGH LIBRARIES

Cognitive Technologies

Managing Editors: D. M. Gabbay J. Siekmann

Editorial Board: A. Bundy J. G. Carbonell
M. Pinkal H. Uszkoreit M. Veloso W. Wahlster
M. J. Wooldridge

Advisory Board:

Luigia Carlucci Aiello
Franz Baader
Wolfgang Bibel
Leonard Bolc
Craig Boutilier
Ron Brachman
Bruce G. Buchanan
Anthony Cohn
Artur d'Avila Garcez
Luis Fariñas del Cerro
Koichi Furukawa
Georg Gottlob
Patrick J. Hayes
James A. Hendler
Anthony Jameson
Nick Jennings
Aravind K. Joshi
Hans Kamp
Martin Kay
Hiroaki Kitano
Robert Kowalski
Sarit Kraus
Maurizio Lenzerini
Hector Levesque
John Lloyd
Alan Mackworth
Mark Maybury
Tom Mitchell
Johanna D. Moore
Stephen H. Muggleton
Bernhard Nebel
Sharon Oviatt
Luis Pereira
Lu Ruqian
Stuart Russell
Erik Sandewall
Luc Steels
Oliviero Stock
Peter Stone
Gerhard Strube
Katia Sycara
Milind Tambe
Hidehiko Tanaka
Sebastian Thrun
Junichi Tsujii
Kurt VanLehn
Andrei Voronkov
Toby Walsh
Bonnie Webber

Wolfgang Wahlster (Ed.)

SmartKom: Foundations of Multimodal Dialogue Systems

With 193 Figures and 63 Tables

Springer

Editor:

Prof. Dr. Wolfgang Wahlster
German Research Center for AI
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany

Managing Editors:

Prof. Dov M. Gabbay
Augustus De Morgan Professor of Logic
Department of Computer Science, King's College London
Strand, London WC2R 2LS, UK

Prof. Dr. Jörg Siekmann
Forschungsbereich Deduktions- und Multiagentensysteme, DFKI
Stuhlsatzenweg 3, Geb. 43, 66123 Saarbrücken, Germany

Library of Congress Control Number: 2006928318

ACM Computing Classification (1998): H.5, H.1.2, I.2, I.3, I.6.

ISSN 1611-2482
ISBN-10 3-540-23732-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-23732-7 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover Design: KünkelLopka, Heidelberg
Typesetting: by the Editor
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig

Printed on acid-free paper 45/3100/YL 5 4 3 2 1 0

Foreword

For about 25 years research and development projects in the area of human-computer interaction (HCI) have been pursued with the objective to adapt the communication and interaction with the machine to the needs of the human user, and not vice versa. But it was only within the past ten years that significant and substantial progress in the practical realization of the results in this area was achieved with the development of individual forms of interaction like speech processing or visualization. The resulting question, then, was whether it is possible to develop easy-to-use multimodal user interfaces with an attractive market potential.

This was the starting point for the inter-disciplinary research activities in human-computer interaction conducted by six large strategic cooperative projects with 102 partners from science and industry. In 1999, these six so-called lead projects came out ahead of 89 proposals overall in an ideas competition launched by the German federal government. These recently finished research projects were supposed to allow human users in both their private and professional environments to multimodally control and diversely use technical systems via natural modalities of interaction like speech, gestures, facial expressions, tactile and graphical input. Ergonomics and user acceptance of these forms of interaction were the key criteria for the development of prototypes that were supposed to have both a strong scientific attractiveness and a high market potential.

One of these lead projects is SMARTKOM. Coordinated by the German Research Center for Artificial Intelligence in Saarbrücken, a consortium of four well-known industrial companies, two small companies and two middle-sized ones, one research institute and three universities was formed. The objective of SMARTKOM was to conduct fundamental research in the area of robust multimodal interaction under realistic conditions, i.e., interaction has to be possible even if the input is underspecified, ambiguous or partially incorrect. The basic idea was to consider and integrate several modes of interaction — in addition to speech especially gestures and facial expressions — instead of only a single modality, and thereby to achieve a substantially better interpretation of the user's intention. This assessment has been confirmed at international conferences worldwide. Thanks to the dedication of all project part-

Cognitive Technologies

Managing Editors: D. M. Gabbay J. Siekmann

Editorial Board: A. Bundy J. G. Carbonell
M. Pinkal H. Uszkoreit M. Veloso W. Wahlster
M. J. Wooldridge

Advisory Board:

Luigia Carlucci Aiello
Franz Baader
Wolfgang Bibel
Leonard Bolc
Craig Boutilier
Ron Brachman
Bruce G. Buchanan
Anthony Cohn
Artur d'Avila Garcez
Luis Fariñas del Cerro
Koichi Furukawa
Georg Gottlob
Patrick J. Hayes
James A. Hendler
Anthony Jameson
Nick Jennings
Aravind K. Joshi
Hans Kamp
Martin Kay
Hiroaki Kitano
Robert Kowalski
Sarit Kraus
Maurizio Lenzerini
Hector Levesque
John Lloyd
Alan Mackworth
Mark Maybury
Tom Mitchell
Johanna D. Moore
Stephen H. Muggleton
Bernhard Nebel
Sharon Oviatt
Luis Pereira
Lu Ruqian
Stuart Russell
Erik Sandewall
Luc Steels
Oliviero Stock
Peter Stone
Gerhard Strube
Katia Sycara
Milind Tambe
Hidehiko Tanaka
Sebastian Thrun
Junichi Tsujii
Kurt VanLehn
Andrei Voronkov
Toby Walsh
Bonnie Webber

Wolfgang Wahlster (Ed.)

SmartKom: Foundations of Multimodal Dialogue Systems

With 193 Figures and 63 Tables

Springer

Editor:

Prof. Dr. Wolfgang Wahlster
German Research Center for AI
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken
Germany

Managing Editors:

Prof. Dov M. Gabbay
Augustus De Morgan Professor of Logic
Department of Computer Science, King's College London
Strand, London WC2R 2LS, UK

Prof. Dr. Jörg Siekmann
Forschungsbereich Deduktions- und Multiagentensysteme, DFKI
Stuhlsatzenweg 3, Geb. 43, 66123 Saarbrücken, Germany

Library of Congress Control Number: 2006928318

ACM Computing Classification (1998): H.5, H.1.2, I.2, I.3, I.6.

ISSN 1611-2482
ISBN-10 3-540-23732-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-23732-7 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover Design: KünkelLopka, Heidelberg
Typesetting: by the Editor
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig

Printed on acid-free paper 45/3100/YL 5 4 3 2 1 0

Foreword

For about 25 years research and development projects in the area of human-computer interaction (HCI) have been pursued with the objective to adapt the communication and interaction with the machine to the needs of the human user, and not vice versa. But it was only within the past ten years that significant and substantial progress in the practical realization of the results in this area was achieved with the development of individual forms of interaction like speech processing or visualization. The resulting question, then, was whether it is possible to develop easy-to-use multimodal user interfaces with an attractive market potential.

This was the starting point for the inter-disciplinary research activities in human-computer interaction conducted by six large strategic cooperative projects with 102 partners from science and industry. In 1999, these six so-called lead projects came out ahead of 89 proposals overall in an ideas competition launched by the German federal government. These recently finished research projects were supposed to allow human users in both their private and professional environments to multimodally control and diversely use technical systems via natural modalities of interaction like speech, gestures, facial expressions, tactile and graphical input. Ergonomics and user acceptance of these forms of interaction were the key criteria for the development of prototypes that were supposed to have both a strong scientific attractiveness and a high market potential.

One of these lead projects is SMARTKOM. Coordinated by the German Research Center for Artificial Intelligence in Saarbrücken, a consortium of four well-known industrial companies, two small companies and two middle-sized ones, one research institute and three universities was formed. The objective of SMARTKOM was to conduct fundamental research in the area of robust multimodal interaction under realistic conditions, i.e., interaction has to be possible even if the input is underspecified, ambiguous or partially incorrect. The basic idea was to consider and integrate several modes of interaction — in addition to speech especially gestures and facial expressions — instead of only a single modality, and thereby to achieve a substantially better interpretation of the user's intention. This assessment has been confirmed at international conferences worldwide. Thanks to the dedication of all project part-

ners SMARTKOM's ambitious objectives have been more than accomplished, as for instance:

- the situation-dependent recognition of underspecified, ambiguous or partially incorrect input on both a syntactic and a pragmatic level was demonstrated successfully,
- a multimodal semantic representation language was developed (M3L) that substantially contributes to a worldwide standardization, and last but not least
- speech-based dialogic Web services for car drivers and pedestrians were developed.

Moreover, the know-how gained in the project was protected for the German economy through 52 patent applications, 29 spin-off products and six spin-off companies so far. In the scientific area, the SMARTKOM project resulted in 255 publications, 66 diploma theses, Ph.D. and habilitation theses, State doctorates as well as six appointments to professorships. This makes SMARTKOM the most successful of all 29 lead projects of the Federal Ministry of Education and Research started since 1998. SMARTKOM was funded with 16.8 million € between September 1999 and September 2003. The overall financial means including the matching funds from industry amounted to 25.7 million €.

This book provides a comprehensive overview of the broad spectrum of results of the research conducted in SMARTKOM. I thank and give credit to everyone involved in the project but especially to Professor Wolfgang Wahlster's professional project management and his competent scientific leadership of the distinguished team of researchers.

Bonn, May 2006

Dr. Bernd Reuse
Head of the Software Systems Division
German Federal Ministry of Education and Research

Acknowledgment

A book such as this one could obviously not be put together without the help and cooperation of many people.

I am particularly indebted to the authors who graciously made their contributions available in a timely fashion.

I would like to thank Dr. Anselm Blocher for his excellent editorial assistance and the production of the final camera-ready copy. Special praise goes to Leivy Michelly Kaul and Alexander Kowalski for their assistance in formatting and copy-editing the book. Special thanks go to Ronan Nugent from Springer for his continuous copy-editing and production support.

The SMARTKOM project was made possible by funding from the German Federal Ministry of Education and Research (BMBF) under contract number number 01 IL 905. I would like to thank Dr. Bernd Reuse, Head of the Software Systems Division at BMBF, for his constant and tireless support of the SMARTKOM project.

Saarbrücken, May 2006

Wolfgang Wahlster
Scientific Director of the SMARTKOM Project
CEO of the
German Research Center for Artificial Intelligence
(DFKI)

List of Contributors

J. Adelhardt
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
adelhard@
informatik.uni-erlangen.de

J. Alexandersson
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
janal@dfki.de

H. Aras
European Media Lab
Schloß-Wolfsbrunnenweg 33
D-69118 Heidelberg
Hidir.Aras@eml.villa-bosch.de

M. Baudis
Mundwerk AG
Immanuelkirchstraße 3-4
D-10405 Berlin
berlin@excelsisnet.com

A. Batliner
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
batliner@
informatik.uni-erlangen.de

T. Becker
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
becker@dfki.de

A. Berton
DaimlerChrysler AG
Wilhelm-Runge-Straße 11
D-89081 Ulm
andre.berton@
daimlerchrysler.com

A. Blocher
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
blocher@dfki.de

N. Braunschweiler
IMS, University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
braunnt@ims.uni-stuttgart.de

J. Bryant
International Computer Science Institute
1947 Center Street
Berkeley, CA, USA 94704-1105
jbryant@icsi.berkeley.edu

D. Bühler
University of Ulm
Albert-Einstein-Allee 43
D-89081 Ulm
dirk.buehler@.uni-ulm.de

V. Chandrasekhara
quadox AG
Ohmstraße 2
D-69190 Walldorf
vasu.chandrasekhara@quadox.de

G. Dogil
IMS, University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
dogil@ims.uni-stuttgart.de

R. Engel
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
ralf.engel@dfki.de

M. Emele
Sony International (Europe) GmbH
Heinrich-Hertz-Straße 1
D-70327 Stuttgart
emele@gmx.net

C. Frank
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
frank@
informatik.uni-erlangen.de

D. Gelbart
International Computer Science Institute
1947 Center Street
Berkeley, CA, USA 94704-1105
gelbart@icsi.berkeley.edu

S. Goronzy
3SOFT GmbH
Frauenweiherstraße 14
D-91058 Erlangen
Silke.Goronzy@3SOFT.de

S. Grashey
Siemens AG
Otto-Hahn-Ring 6
D-81730 Munich
stephan.grashey@siemens.com

I. Gurevych
Technical University Darmstadt
Hochschulstraße 10
D-64289 Darmstadt
Gurevych@
tk.informatik.tu-darmstadt.de

U. Haiber
DaimlerChrysler AG
Wilhelm-Runge-Straße 11
D-89081 Ulm
udo.haiber@daimlerchrysler.com

J. Häußler
European Media Lab
Schloß-Wolfsbrunnenweg 33
D-69118 Heidelberg
jochen.haeussler@
eml.villa-bosch.de

G. Herzog
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Gerd.Herzog@dfki.de

A. Horndasch
Sympalog Speech Technologies AG
Karl-Zucker-Straße 10
D-91052 Erlangen
horndasch@sympalog.de

C. Hying
Sony International (Europe) GmbH
Heinrich-Hertz-Straße 1
D-70327 Stuttgart
christian.hying@
ims.uni-stuttgart.de

M. Jöst
European Media Lab
Schloß-Wolfsbrunnenweg 33
D-69118 Heidelberg
joest@eml.org

A. Kaltenmeier
DaimlerChrysler AG
Wilhelm-Runge-Straße 11
D-89081 Ulm
alfred.kaltenmeier@
daimlerchrysler.com

A. Kellner
Philips GmbH
Weißhausstraße 2
D-52066 Aachen
Andreas.Kellner@philips.com

T. Klankert
IMS, University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
klankert@ims.uni-stuttgart.de

S. Krüger
SAP AG
Dietmar-Hopp-Allee 16
D-69190 Walldorf
sven.krueger@sap.com

Y.H. Lam
Sony International (Europe) GmbH
Heinrich-Hertz-Straße 1
D-70327 Stuttgart
lam@sony.de

M. Löckelt
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
loeckelt@dfki.de

M. Lützeler
Siemens AG
Otto-Hahn-Ring 6
D-81730 Munich
Michael.Luetzeler@siemens.com

R. Malaka
University of Bremen
Bibliothekstraße 1, MZH
D-28359 Bremen
malaka@informatik.uni-bremen.de

M. Merdes
EML Research gGmbH
Schloß-Wolfsbrunnenweg 33
D-69118 Heidelberg
matthias.merdes@
eml-r.villa-bosch.de

W. Minker
University of Ulm
Albert-Einstein-Allee 43
D-89081 Ulm
wolfgang.minker@uni-ulm.de

B. Möbius
IMS, University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
moebius@ims.uni-stuttgart.de

G. Möhler
IMS, University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
moehler@gmx.de

E. Morais
Faculdade de Eng. Elétrica e de
Computação
State University of Campinas, Brazil
emorais@decom.fee.unicamp.br

N. Morgan
International Computer Science Institute
1947 Center Street
Berkeley, CA, USA 94704-1105
morgan@icsi.berkeley.edu

A. Ndiaye
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
ndiaye@dfki.de

H. Niemann
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
niemann@
informatik.uni-erlangen.de

E. Nöth
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
noeth@
informatik.uni-erlangen.de

D. Pfisterer
University Lübeck
Ratzeburger Allee 160
D-23538 Lübeck
pfisterer@itm.uni-luebeck.de

N. Pfleger
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Norbert.Pfleger@dfki.de

P. Poller
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
poller@dfki.de

T. Portele
Philips GmbH
Weißhausstraße 2
D-52066 Aachen
Thomas.Portele@philips.com

R. Porzel
European Media Lab
Schloß-Wolfsbrunnenweg 33
D-69118 Heidelberg
porzel@eml.org

S. Rabold
University of Munich
Schellingstraße 3
D-80799 Munich
rabold@phonetik.uni-muenchen.de

J. Racky
Siemens AG
Otto-Hahn-Ring 6
D-81730 Munich
Jens.Racky@siemens.com

H. Rapp
MediaInterface Dresden GmbH
Washingtonstraße 16/16a
D-01139 Dresden
rapp@mediainterface.de

S. Rapp
Sony International (Europe) GmbH
Heinrich-Hertz-Straße 1
D-70327 Stuttgart
rapp@conante.com

N. Reithinger
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Norbert.Reithinger@dfki.de

B. Reuse
BMBF
Heinemannstraße 2
D-53175 Bonn
Bernd.Reuse@BMBF.bund400.de

H. Röttger
Siemens AG
Otto-Hahn-Ring 6
D-81730 Munich
Hans.Roettger@siemens.com

B. Säuberlich
IMS, University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
bettina.saeuberlich@
ims.uni-stuttgart.de

F. Schiel
University of Munich
Schellingstraße 3
D-80799 Munich
schiel@phonetik.uni-muenchen.de

O. Schreiner
Technical University Berlin
Straße des 17. Juni 135
10623 Berlin

M. Schuster
Siemens AG
Otto-Hahn-Ring 6
D-81730 Munich
Matthias.Schuster@siemens.com

A. Schweitzer
University of Stuttgart
Azenbergstraße 12
D-70174 Stuttgart
Antje.Schweitzer@
ims.uni-stuttgart.de

R.P. Shi
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
shi@informatik.uni-erlangen.de

S. Steininger
University of Munich
Schellingstraße 3
D-80799 Munich
steins@phonetik.uni-muenchen.de

A. Stolcke
International Computer Science Institute
1947 Center Street
Berkeley, CA, USA 94704-1198
stolcke@icsi.berkeley.edu

M. Streit
Rue de la Gare 108
F-57150 Creutzwald

M. Thomae
Technical University of Munich
Arcisstr. 21
D-80333 Munich
thomae@ei.tum.de

S. Torge
Sony International (Europe) GmbH
Heinrich-Hertz-Straße 1
D-70327 Stuttgart
sunna.torge@web.de

V. Tschernomas
TNM Software GmbH
Grubenstraße 107
D-66540 Neunkirchen
tschernomas@tnmsoft.com

U. Türk
University of Munich
Schellingstraße 3
D-80799 Munich
tuerk@phonetik.uni-muenchen.de

Z. Valsan
Sony International (Europe) GmbH
Heinrich-Hertz-Straße 1
D-70327 Stuttgart
valsan@sony.de

J. te Vrugt
Philips GmbH
Weißhausstraße 2
D-52066 Aachen
Juergen.te.Vrugt@philips.com

W. Wahlster
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
wahlster@dfki.de

V. Zeissler
Friedrich-Alexander-University
Erlangen-Nuremberg
Martenstraße 3
D-91058 Erlangen
zeissler@
informatik.uni-erlangen.de

Contents

Part I Introduction

Dialogue Systems Go Multimodal: The SmartKom Experience
Wolfgang Wahlster 3

Facts and Figures About the SmartKom Project
Anselm Blocher 29

An Exemplary Interaction with SmartKom
Norbert Reithinger, Gerd Herzog 41

Part II Multimodal Input Analysis

The SmartKom Architecture: A Framework for Multimodal Dialogue Systems
Gerd Herzog, Norbert Reithinger 55

Modeling Domain Knowledge: Know-How and Know-What
Iryna Gurevych, Robert Porzel, Rainer Malaka 71

Speech Recognition
André Berton, Alfred Kaltenmeier, Udo Haiber, Olaf Schreiner 85

Class-Based Language Model Adaptation
Martin C. Emele, Zica Valsan, Yin Hay Lam, Silke Goronzy 109

The Dynamic Lexicon
Silke Goronzy, Stefan Rapp, Martin Emele 123

The Prosody Module
Viktor Zeißler, Johann Adelhardt, Anton Batliner, Carmen Frank, Elmar Nöth, Rui Ping Shi, Heinrich Niemann 139

The Sense of Vision: Gestures and Real Objects
Jens Racky, Michael Lützeler, Hans Röttger 153

The Facial Expression Module
Carmen Frank, Johann Adelhardt, Anton Batliner, Elmar Nöth, Rui Ping Shi, Viktor Zeißler, Heinrich Niemann 167

Multiple Biometrics
Stephan Grashey, Matthias Schuster 181

Natural Language Understanding
Ralf Engel 195

The Gesture Interpretation Module
Rui Ping Shi, Johann Adelhardt, Anton Batliner, Carmen Frank, Elmar Nöth, Viktor Zeißler, Heinrich Niemann 209

Part III Multimodal Dialogue Processing

Modality Fusion
Ralf Engel, Norbert Pfleger 223

Discourse Modeling
Jan Alexandersson, Norbert Pfleger 237

Overlay: The Basic Operation for Discourse Processing
Jan Alexandersson, Tilman Becker, Norbert Pfleger 255

In Context: Integrating Domain- and Situation-Specific Knowledge
Robert Porzel, Iryna Gurevych, Rainer Malaka 269

Intention Recognition
Jürgen te Vrugt, Thomas Portele 285

Plan-Based Dialogue Management for Multiple Cooperating Applications
Markus Löckelt 301

Emotion Analysis and Emotion-Handling Subdialogues
Michael Streit, Anton Batliner, Thomas Portele 317

Problematic, Indirect, Affective, and Other Nonstandard Input Processing
Michael Streit 333

Part IV Multimodal Output Generation

Realizing Complex User Wishes with a Function Planning Module
Sunna Torge, Christian Hying 349

Intelligent Integration of External Data and Services into SmartKom
Hidir Aras, Vasu Chandrasekhara, Sven Krüger, Rainer Malaka, Robert Porzel 363

Multimodal Fission and Media Design
Peter Poller, Valentin Tschernomas 379

Natural Language Generation with Fully Specified Templates
Tilman Becker 401

Multimodal Speech Synthesis
Antje Schweitzer, Norbert Braunschweiler, Grzegorz Dogil, Tanja Klankert, Bernd Möbius, Gregor Möhler, Edmilson Morais, Bettina Säuberlich, Matthias Thomae 411

Part V Scenarios and Applications

Building Multimodal Dialogue Applications: System Integration in SmartKom
Gerd Herzog, Alassane Ndiaye 439

SmartKom-English: From Robust Recognition to Felicitous Interaction
David Gelbart, John Bryant, Andreas Stolcke, Robert Porzel, Manja Baudis and Nelson Morgan 453

SmartKom-Public
Axel Horndasch, Horst Rapp, Hans Röttger 471

SmartKom-Home: The Interface to Home Entertainment
Thomas Portele, Silke Goronzy, Martin Emele, Andreas Kellner, Sunna Torge, Jürgen te Vrugt 493

SmartKom-Mobile: Intelligent Interaction with a Mobile System
Rainer Malaka, Jochen Häußler, Hidir Aras, Matthias Merdes, Dennis Pfisterer, Matthias Jöst, and Robert Porzel 505

SmartKom-Mobile Car: User Interaction with Mobile Services in a Car Environment
André Berton, Dirk Bühler, Wolfgang Minker 523

Part VI Data Collection and Evaluation

Wizard-of-Oz Recordings
Florian Schiel, Ulli Türk 541

Annotation of Multimodal Data
Silke Steininger, Florian Schiel, Susen Rabold 571

Multimodal Emogram, Data Collection and Presentation
Johann Adelhardt, Carmen Frank, Elmar Nöth, Rui Ping Shi, Viktor Zeißler, Heinrich Niemann 597

Empirical Studies for Intuitive Interaction
Iryna Gurevych, Robert Porzel 603

Evaluation of Multimodal Dialogue Systems
Florian Schiel 617

Part I

Introduction

Dialogue Systems Go Multimodal: The SmartKom Experience

Wolfgang Wahlster

DFKI GmbH, Saarbrücken, Germany
wahlster@dfki.de

Summary. Multimodal dialogue systems exploit one of the major characteristics of human-human interaction: the coordinated use of different modalities. Allowing all of the modalities to refer to and depend upon each other is a key to the richness of multimodal communication. We introduce the notion of symmetric multimodality for dialogue systems in which all input modes (e.g., speech, gesture, facial expression) are also available for output, and vice versa. A dialogue system with symmetric multimodality must not only understand and represent the user's multimodal input, but also its own multimodal output. We present an overview of the SMARTKOM system that provides full symmetric multimodality in a mixed-initiative dialogue system with an embodied conversational agent. SMARTKOM represents a new generation of multimodal dialogue systems that deal not only with simple modality integration and synchronization but cover the full spectrum of dialogue phenomena that are associated with symmetric multimodality (including crossmodal references, one-anaphora, and backchannelling). We show that SMARTKOM's plug-and-play architecture supports multiple recognizers for a single modality, e.g., the user's speech signal can be processed by three unimodal recognizers in parallel (speech recognition, emotional prosody, boundary prosody). We detail SMARTKOM's three-tiered representation of multimodal discourse, consisting of a domain layer, a discourse layer, and a modality layer. We discuss the limitations of SMARTKOM and how they are overcome in the follow-up project SmartWeb. In addition, we present the research roadmap for multimodality addressing the key open research questions in this young field. To conclude, we discuss the economic and scientific impact of the SMARTKOM project, which has led to more than 50 patents and 29 spin-off products.

1 The Need for Multimodality

In face-to-face situations, human dialogue is not only based on speech but also on nonverbal communication including gesture, gaze, facial expression, and body posture. Multimodal dialogue systems exploit one of the major characteristics of human-human interaction: the coordinated use of different modalities. The term *modality* refers to the human senses: vision, audition, olfaction, touch, and taste. In addition, human communication is based on socially shared code systems like natural languages, body languages, and pictorial languages with their own syntax, semantics,

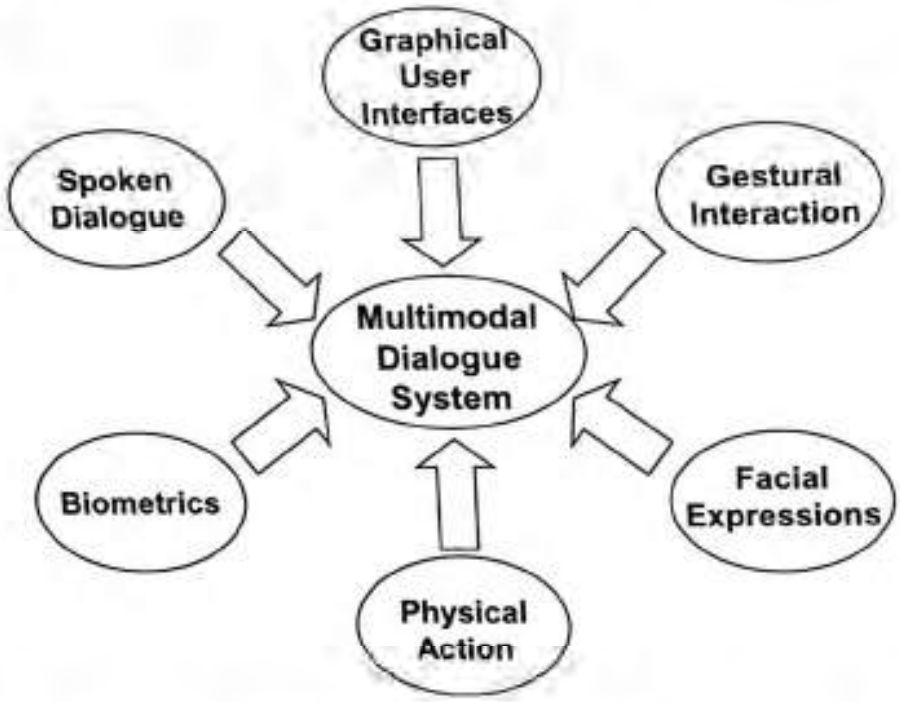

Fig. 1. Merging various dialogue and interaction paradigms into multimodal dialogue systems

and pragmatics. A single semiotic code may be supported by many modalities (Maybury and Wahlster, 1998). For instance, language can be supported visually (i.e., written language), aurally (i.e., spoken language) or tactilely (i.e., Braille script) – in fact, spoken language can have a visual component (e.g., lipreading). Allowing all of the modalities to refer to and depend upon each other is a key to the richness of multimodal communication (see Fig. 1).

Unlike traditional keyboard and mouse interfaces or unimodal spoken dialogue systems, multimodal dialogue systems permit flexible use of input and output modes (Oviatt, 2003). Since there are large individual differences in ability and preference to use different modalities, a multimodal dialogue system permits diverse user groups to exercise control over how they interact with application systems. Especially for mobile tasks, multimodal dialogue systems permit the modality choice and switching that is needed during the changing situational conditions.

With the proliferation of embedded computers in everyday life and the emergence of ambient intelligence, multimodal dialogue systems are no longer only limited to traditional human-computer communication, but become more generally a key component for advanced human-technology interaction and even multimodal human-environment communication (Wahlster and Wasinger, 2006).

Self-service systems, online help systems, Web services, mobile communication devices, remote control systems, smart appliances, and dashboard computers are providing ever more functionality. However, along with greater functionality, the user must also come to terms with the greater complexity and a steeper learning curve.

This complexity is compounded by the sheer proliferation of different systems lacking a standard user interface. The growing emphasis on multimodal dialogue systems is fundamentally inspired by the aim to support natural, flexible, efficient, and powerfully expressive means of human-computer communication that are easy to learn and use. Advances in human language technology and in perceptive user interfaces offer the promise of pervasive access to online information and Web services. The long-term goal of the research in multimodal dialogue systems is to allow the average person to interact with computerized technologies anytime and anywhere without special skills or training, using such common devices as a smartphone or a PDA.

We begin by describing the scientific goals of the SMARTKOM project in Sect. 2, before we introduce the notion of symmetric multimodality in Sect. 3. In Sect. 4, we introduce SMARTKOM as a flexible and adaptive multimodal dialogue shell and show in Sect. 5 that SMARTKOM bridges the full loop from multimodal perception to physical action. SMARTKOM's distributed component architecture, realizing a multiblackboard system, is described in Sect. 6. Then in Sects. 7 and 8, we describe SMARTKOM's methods for multimodal fusion and fission. Section 9 discusses the role of the three-tiered multimodal discourse model in SMARTKOM. Section 10 gives a brief introduction to SmartWeb, the follow-up project to SMARTKOM, which supports open-domain question answering. Open research questions in the young field of multimodal dialogue systems are presented in Sect. 11 in a research roadmap for multimodality. Finally, we discuss the economic and scientific impact of the SMARTKOM project in Sect. 12.

2 SmartKom: A Massive Approach to Multimodality

In this book, we present the theoretical and practical foundations of multimodal dialogue systems using the results of our large-scale project SMARTKOM as the background of our discussion. Our SMARTKOM system (http://www.smartkom.org) is designed to support a wide range of collaborative and multimodal help dialogues that allow users to intuitively and efficiently access the functionalities needed for their task. The application of the SMARTKOM technology is especially motivated in non-desktop scenarios, such as smart rooms, kiosks, or mobile environments. SMARTKOM features the situated understanding of possibly imprecise, ambiguous or incomplete multimodal input and the generation of coordinated, cohesive, and coherent multimodal presentations. SMARTKOM's interaction management is based on representing, reasoning, and exploiting models of the user, domain, task, context, and modalities. The system is capable of real-time dialogue processing, including flexible multimodal turn-taking, backchannelling, and metacommunicative interaction. The four major scientific goals of SMARTKOM were to:

- explore and design new symbolic and statistical methods for the seamless fusion and mutual disambiguation of multimodal input on semantic and pragmatic levels
- generalize advanced discourse models for spoken dialogue systems so that they can capture a broad spectrum of multimodal discourse phenomena

- explore and design new constraint-based and plan-based methods for multimodal fission and adaptive presentation layout
- integrate all these multimodal capabilities in a reusable, efficient and robust dialogue shell that guarantees flexible configuration, domain independence and plug-and-play functionality

3 Towards Symmetric Multimodality

SMARTKOM provides *full symmetric multimodality* in a mixed-initiative dialogue system. Symmetric multimodality means that all input modes (speech, gesture, facial expression) are also available for output, and vice versa. A dialogue system with symmetric multimodality must not only understand and represent the user's multimodal input, but also its own multimodal output.

In this sense, SMARTKOM's modality fission component provides the inverse functionality of its modality fusion component, since it maps a communicative intention of the system onto a coordinated multimodal presentation (Wahlster, 2002). SMARTKOM provides an anthropomorphic and affective user interface through an embodied conversational agent called Smartakus. This life-like character uses coordinated speech, gesture and facial expression for its dialogue contributions.

Thus, SMARTKOM supports face-to-face dialogic interaction between two agents that share a common visual environment: the human user and Smartakus, an autonomous embodied conversational agent. The "i"-shape of Smartakus is analogous to that used for *information* kiosks (see Fig. 2). Smartakus is modeled in 3D Studio Max. It is a self-animated interface agent with a large repertoire of gestures, postures and facial expressions. Smartakus uses body language to notify users that it is waiting for their input, that it is listening to them, that is has problems in understanding their input, or that it is trying hard to find an answer to their questions.

Most of the previous multimodal interfaces do not support symmetric multimodality, since they focus either on multimodal fusion (e.g., QuickSet, see Cohen et al. (1997), or MATCH, see Johnston et al. (2002)) or multimodal fission (e.g., WIP, see Wahlster et al. (1993)). But only true multimodal dialogue systems like SMARTKOM create a natural experience for the user in the form of daily human-to-human communication, by allowing both the user and the system to combine the same spectrum of modalities.

SMARTKOM is based on the situated delegation-oriented dialogue paradigm (SDDP, see Fig. 2): The user delegates a task to a virtual communication assistant (Wahlster et al., 2001). This cannot however be done in a simple command-and-control style for more complex tasks. Instead, a collaborative dialogue between the user and the agent elaborates the specification of the delegated task and possible plans of the agent to achieve the user's intentional goal. The user delegates a task to Smartakus and helps the agent, where necessary, in the execution of the task. Smartakus accesses various digital services and appliances on behalf of the user, collates the results, and presents them to the user.

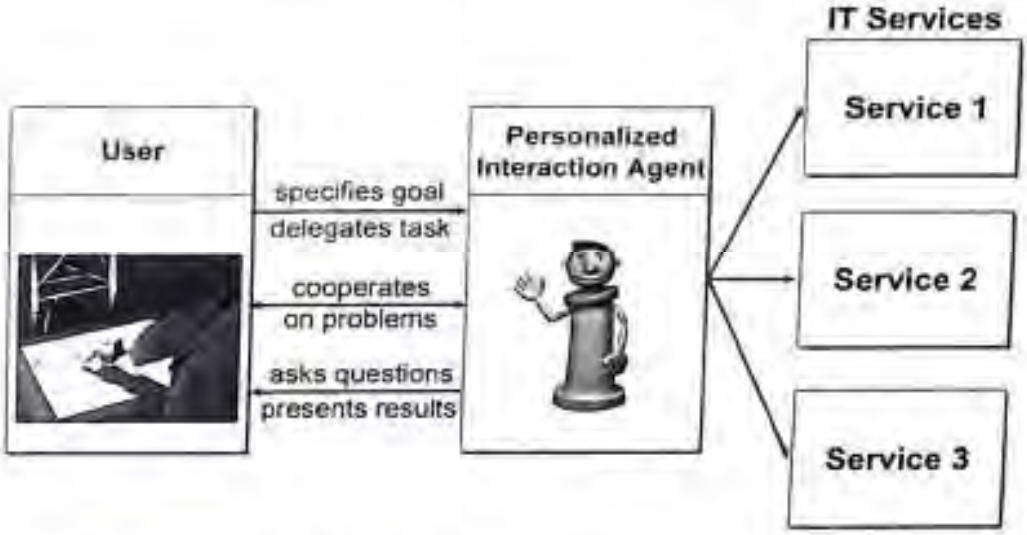

Fig. 2. SMARTKOM's SDDP interaction metaphor

SMARTKOM represents a new generation of multimodal dialogue systems that deal not only with simple modality integration and synchronization but cover the full spectrum of dialogue phenomena that are associated with symmetric multimodality. One of the technical goals of our research in the SMARTKOM project was to address the following important discourse phenomena that arise in multimodal dialogues:

- mutual disambiguation of modalities
- multimodal deixis resolution and generation
- crossmodal reference resolution and generation
- multimodal anaphora resolution and generation
- multimodal ellipsis resolution and generation
- multimodal turn-taking and backchannelling

Symmetric multimodality is a prerequisite for a principled study of these discourse phenomena.

4 Towards a Flexible and Adaptive Shell for Multimodal Dialogues

SMARTKOM was designed with a clear focus on flexibility, as a transmutable system that can engage in many different types of tasks in different usage contexts. The same software architecture and components are used in various roles that Smartakus can play in the following three fully operational experimental application scenarios (see Fig. 3):

- a *communication companion* that helps with phone, fax, email, and authentication tasks

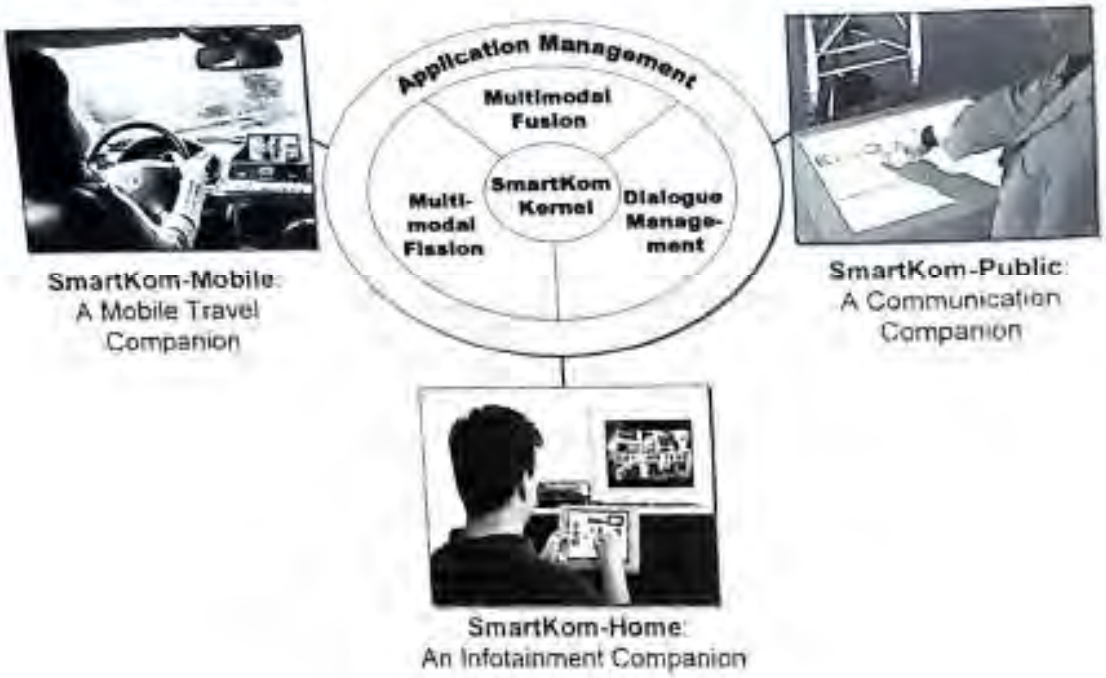

Fig. 3. The three application scenarios of SMARTKOM

- an *infotainment companion* that helps to select media content and to operate various TV appliances (using a tablet computer as a mobile client)
- a *mobile travel companion* that helps with navigation and point-of-interest information retrieval in location-based services (using a PDA as a mobile client)

Currently, the user can delegate 43 types of complex tasks to Smartakus in multimodal dialogues. The SMARTKOM architecture supports not only simple multimodal command-and-control interfaces, but also coherent and cooperative dialogues with mixed initiative and a synergistic use of multiple modalities. SMARTKOM's plug-and-play architecture supports easy addition of new application services.

Figure 4 shows a three-camera configuration of SMARTKOM that can be used as a multimodal communication kiosk for airports, train stations, or other public places where people may seek information on facilities such as hotels, restaurants, and movie theatres. Users can also access their personalized Web services. The user's speech input is captured with a directional microphone. The user facial expressions of emotion are captured with a CCD camera and his gestures are tracked with an infrared camera. A video projector is used for the projection of SMARTKOM's graphical output onto a horizontal surface. Two speakers under the projection surface provide the speech output of the life-like character. An additional camera that can automatically tilt and pan is used to capture images of documents or 3D objects that the user would like to include in multimedia messages composed with the help of SMARTKOM.

As a resource-adaptive multimodal system, the SMARTKOM architecture supports a flexible embodiment of the life-like character that is used as a conversational

Fig. 4. Multimodal input and output devices of SMARTKOM-Public

partner in multimodal dialogue. The Smartakus agent is visualized either simply as a talking head together with an animated hand, when screen space is scarce, or as a full-body character, when enough screen space is available (see Fig. 2). Thus, Smartakus is embodied on a PDA differently than on a tablet computer or on the large top-projected screen used in the public information kiosk.

5 Perception and Action Under Multimodal Conditions

SMARTKOM bridges the full loop from multimodal perception to physical action. Since the multimodal interaction with Smartakus covers both communicative and physical acts, the mutual understanding of the user and the system can often be validated by checking whether the user and the system "do the right thing" for completing the task at hand.

In a multimodal dialogue about the TV program, the user may browse a TV show database, create a personalized TV listing, and finally ask Smartakus to switch on the TV and tune to a specific program. Smartakus can also carry out more complex actions like programming a VCR to record the user's favourite TV show. Moreover, it can scan a document or a 3D object with its camera and then send the captured image to another person as an email attachment. Fig. 5 shows Dr. Johannes Rau, the former German Federal President, using SMARTKOM's multimodal dialogue capabilities to scan the "German Future Award" trophy and send the scanned image via email to a colleague. This example shows that, on the one hand, multimodal dialogue contributions can trigger certain actions of Smartakus. On the other hand, Smartakus

Fig. 5. The former German Federal President e-mailing a scanned image with the help of Smartakus

Fig. 6. Interactive biometric authentication by hand contour recognition

may also ask the user to carry out certain physical actions during the multimodal dialogue.

For example, Smartakus will ask the user to place his hand with spread fingers on a virtual scanning device, or to use a write-in field projected on the screen for his signature, when biometric authentication by hand contour recognition or signature verification is requested by a security-critical application. Fig. 6 shows a situation in which Smartakus has found an address book entry for the user, after he has introduced himself by name. Since the address book entry, which is partially visualized

Fig. 7. Adaptive perceptual feedback on the system state

by SMARTKOM on the left part of the display, requests hand contour authentication for this particular user. Smartakus asks the user to place his hand on the marked area of the projected display, so that the hand contour can be scanned by its camera (see Fig. 6).

Since quite complex tasks can be delegated to Smartakus, there may be considerable delays in replying to a request. Our WOZ (Wizard-of-Oz) experiments and user tests with earlier prototypes of SMARTKOM showed clearly that users want simple and fast feedback on the state of the system in such situations. Therefore, a variety of adaptive perceptual feedback mechanisms have been realized in SMARTKOM.

In the upper-left corner of a presentation, SMARTKOM can display a "magic eye" icon, that lights up while the processing of the user's multimodal input is proceeding (see the left part of Fig. 7). "Magic eye" is the common name applied to the green-glow tubes used in 1930s radio equipment to visually assist the listener in tuning a radio station to the point of greatest signal strength. Although SMARTKOM works in real-time, there may be some processing delays caused by corrupted input or complex disambiguation processes.

An animated dashed line (see the left part of Fig. 7) circles the Smartakus character, while the system is engaged in an information retrieval task (e.g., access to maps, EPG (Electronic Program Guide), Web sites). This type of feedback is used when screen space is scarce. When more screen space is available, an animation sequence that shows Smartakus working on a laptop is used for the same kind of feedback. When Smartakus is downloading a large file, it can show a progress bar to indicate to the user how the data transfer is going (see the right part of Fig. 7).

6 A Multiblackboard Platform with Ontology-Based Messaging

SMARTKOM is based on a distributed component architecture, realizing a multiblackboard system. The integration platform is called MULTIPLATFORM (**Multiple Language Target Integration Platform for Modules**, see Herzog et al. (2003)) and is built on top of open source software. The natural choice to realize an open, flexible and scalable software architecture is that of a distributed system, which is able to integrate heterogeneous software modules implemented in diverse programming languages and running on different operating systems. SMARTKOM includes more than 40 asynchronously running modules coded in four different programming languages: C, C++, Java, and Prolog.

The MULTIPLATFORM testbed includes a message-oriented middleware. The implementation is based on PVM, which stands for *parallel virtual machine*. On top of PVM, a message-based communication framework is implemented based on the so-called publish/subscribe approach. In contrast to unicast routing known from multiagent frameworks, which realize a direct connection between a message sender and a known receiver, MULTIPLATFORM is based on the more efficient multicast addressing scheme. Instead of addressing one or several receivers directly, the sender publishes a notification on a named message queue, so that the message can be forwarded to a list of subscribers. This kind of distributed event notification makes the communication framework very flexible as it focuses on the data to be exchanged and it decouples data producers and data consumers. Compared with point-to-point messaging used in multiagent frameworks like OAA (Martin et al., 1999), the publish/subscribe scheme helps to reduce the number and complexity of interfaces significantly.

GCSI, the **G**alaxy **C**ommunicator **S**oftware **I**nfrastructure (Seneff et al., 1999) architecture is also fundamentally different from our approach. The key component of GCSI is a central hub, which mediates the interaction among various servers that realize different dialogue system components. Within MULTIPLATFORM there exists no such centralized controller component, since this could become a bottleneck for more complex multimodal dialogue architectures.

In order to provide publish/subscribe messaging on top of PVM, we have added another software layer called PCA (**P**ool **C**ommunication **A**rchitecture). In MULTIPLATFORM, the term *data pool* is used to refer to named message queues. Every single pool can be linked with a pool data format specification in order to define admissible message contents. The messaging system is able to transfer arbitrary data contents, and provides excellent performance characteristics (Herzog et al., 2003).

In SMARTKOM, we have developed M3L (**M**ultimodal **M**arkup **L**anguage) as a complete XML language that covers all data interfaces within this complex multimodal dialogue system. Instead of using several quite different XML languages for the various data pools, we aimed at an integrated and coherent language specification, which includes all substructures that may occur on the different pools. In order to make the specification process manageable and to provide a thematic organization, the M3L language definition has been decomposed into about 40 schema specifica-

tions. The basic data flow from user input to system output continuously adds further processing results so that the representational structure will be refined, step-by-step.

The ontology that is used as a foundation for representing domain and application knowledge is coded in the ontology language OIL. Our tool OIL2XSD (Gurevych et al., 2003) transforms an ontology written in OIL (Fensel et al., 2001) into an M3L compatible XML Schema definition. The information structures exchanged via the various blackboards are encoded in M3L. M3L is defined by a set of XML schemas. For example, the word hypothesis graph and the gesture hypothesis graph, the hypotheses about facial expressions, the media fusion results, and the presentation goal are all represented in M3L. M3L is designed for the representation and exchange of complex multimodal content. It provides information about segmentation, synchronization, and the confidence in processing results. For each communication blackboard, XML schemas allow for automatic data and type checking during information exchange. The XML schemas can be viewed as typed feature structures. SMARTKOM uses unification and a new operation called OVERLAY (Alexandersson and Becker, 2003) of typed feature structures encoded in M3L for discourse processing.

Application developers can generate their own multimodal dialogue system by creating knowledge bases with application-specific interfaces, and plugging them into the reusable SMARTKOM shell. It is particularly easy to add or remove modality analyzers or renderers, even dynamically while the system is running. This plug and play of modalities can be used to adjust the system's capability to handle different demands of the users, and the situative context they are currently in. Since SMARTKOM's modality analyzers are independent from the respective device-specific recognizers, the system can switch in real-time, for example, between video-based, pen-based or touch-based gesture recognition. SMARTKOM's architecture, its dialogue backbone, and its fusion and fission modules are reusable across applications, domains, and modalities.

MULTIPLATFORM is running on the SMARTKOM server that consists of 3 dual Xeon 2.8 GHz processors. Each processor uses 1.5 GB of main memory. One processor is running under Windows 2000, and the other two under Linux. The mobile clients (an iPAQ Pocket PC for the mobile travel companion and a Fujitsu Stylistic 3500X webpad for the infotainment companion) are linked to the SMARTKOM server via WaveLAN.

7 Reducing Uncertainty and Ambiguity by Modality Fusion

The analysis of the various input modalities by SMARTKOM is typically plagued by uncertainty and ambiguity. The speech recognition system produces a word hypothesis graph with acoustic scores, stating which word might have been spoken in a certain time frame. The prosody component generates a graph of hypotheses about clause and sentence boundaries with prosodic scores. The gesture analysis component produces a set of scored hypotheses about possible reference objects in the visual context. Finally, the interpretation of facial expressions leads to various scored hypotheses about the emotional state of the user. All the recognizers produce

time-stamped hypotheses, so that the fusion process can consider various temporal constraints. The key function of modality fusion is the reduction of the overall uncertainty and the mutual disambiguation of the various analysis results. By fusing symbolic and statistical information derived from the recognition and analysis components for speech, prosody, facial expression and gesture, SMARTKOM can correct various recognition errors of its unimodal input components and thus provide a more robust dialogue than a unimodal system.

In principle, modality fusion can be realized during various processing stages like multimodal signal processing, multimodal parsing, or multimodal semantic processing. In SMARTKOM, we prefer the latter approach, since for the robust interpretation of possibly incomplete and inconsistent multimodal input, more knowledge sources become available on later processing stages. An early integration on the signal level allows no backtracking and reinterpretation, whereas the multimodal parsing approach has to prespecify all varieties of crossmodal references, and is thus unable to cope robustly with unusual or novel uses of multimodality. However, some early fusion is also used in SMARTKOM, since the scored results from a recognizer for emotional prosody (Batliner et al., 2000) are merged with the results of a recognizer for affective facial expression. The classification results are combined in a synergistic fashion, so that a hypothesis about the affective state of the user can be computed.

In SMARTKOM, the user state is used, for example, in the dialogue-processing backbone to check whether the user is satisfied or not with the information provided by Smartakus. It is interesting to note that SMARTKOM's architecture supports multiple recognizers for a single modality. In the current system, prosody is evaluated by one recognizer for clause boundaries and another recognizer for emotional speech. This means that the user's speech signal is processed by three unimodal recognizers in parallel (speech recognition, emotional prosody, boundary prosody).

The time stamps for all recognition results are extremely important since the confidence values for the classification results may depend on the temporal relations between input modalities. For example, experiments in SMARTKOM have shown that the results from recognizing various facial regions (like eye, nose, and mouth area) can be merged to improve recognition results for affective states like anger or joy. However, while the user is speaking, the mouth area does not predict emotions reliably, so that the confidence value of the mouth area recognizer must be decreased. Thus, SMARTKOM's modality fusion is based on adaptive confidence measures that can be dynamically updated depending on the synchronization of input modalities.

One of the fundamental mechanisms implemented in SMARTKOM's modality fusion component is the extended unification of all scored hypothesis graphs and the application of mutual constraints in order to reduce the ambiguity and uncertainty of the combined analysis results. This approach was pioneered in our XTRA system, an early multimodal dialogue system that assisted the user in filling out a tax form with a combination of typed natural language input and pointing gestures (Wahlster, 1991). QuickSet uses a similar approach (Cohen et al., 1997).

In SMARTKOM, the intention recognizer has the task to finally rank the remaining interpretation hypotheses and to select the most likely one, which is then passed

```
<intentionLattice>
[...]
  <hypothesisSequences>
   <hypothesisSequence>
    <score>
     <source>   acoustic   </source>
     <value>    0.96448    </value>
    </score>
    <score>
     <source>   gesture    </source>
     <value>    0.99791    </value>
    </score>
    <score>
     <source>   understanding   </source>
     <value>    0.91667         </value>
    </score>
    <hypothesis>
     <discourseStatus>
      <discourseAction> set </discourseAction>
      <discourseTopic><goal> epg_info </goal></discourseTopic>
     [...]
     <event id="dim868">
      <informationSearch id="dim869">
       <pieceOfInformation>
        <broadcast id="dim883">
         <avMedium>
          <avMedium id="dim866">
           <avType> featureFilm </avType>
           <title>  Enemy of the State  </title>
          [...]
   </hypothesisSequence>
[...]
  </hypothesisSequences>
</intentionLattice>
```

Confidence in the Speech Recognition Result

Confidence in the Gesture Recognition Result

Confidence in the Speech Understanding Result

Planning Act

Object Reference

Fig. 8. M3L representation of an intention lattice fragment

on to the action planner. The modality fusion process is augmented by SMARTKOM's multimodal discourse model, so that the final ranking of the intention recognizer becomes highly context sensitive. The discourse component produces an additional score that states how good an interpretation hypothesis fits to the previous discourse (Pfleger et al., 2002). As soon as the modality fusion component finds a referential expression that is not combined with an unambiguous deictic gesture, it sends a request to the discourse component asking for reference resolution. If the resolution succeeds, the discourse component returns a completely instantiated domain object.

Figure 8 shows an excerpt from the intention lattice for the user's input "I would like to know more about this [deictic pointing gesture]". It shows one hypothesis sequence with high scores from speech and gesture recognition. A potential reference object for the deictic gesture (the movie title "Enemy of the State") has been found in the visual context. SMARTKOM assumes that the discourse topic relates to an electronic program guide and the intended action of Smartakus refers to the retrieval of information about a particular broadcast.

8 Plan-Based Modality Fission in SmartKom

In SMARTKOM, modality fission is controlled by a presentation planner. The input to the presentation planner is a presentation goal encoded in M3L as a modality-free representation of the system's intended communicative act. This M3L structure is generated by either an action planner or the dynamic help component, which can initiate clarification subdialogues. The presentation planning process can be adapted to various application scenarios via presentation parameters that encode user preferences (e.g., spoken output is preferred by a car driver), output devices (e.g., size of the display), or the user's native language (e.g., German vs. English). A set of XSLT (Extensible Stylesheet Language Transformations) stylesheets is used to transform the M3L representation of the presentation goal, according to the actual presentation parameter setting. The presentation planner recursively decomposes the presentation goal into primitive presentation tasks using 121 presentation strategies that vary with the discourse context, the user model, and ambient conditions. The presentation planner allocates different output modalities to primitive presentation tasks, and decides whether specific media objects and presentation styles should be used by the media-specific generators for the visual and verbal elements of the multimodal output.

The presentation planner specifies presentation goals for the text generator, the graphics generator, and the animation generator. The animation generator selects appropriate elements from a large catalogue of basic behavioral patterns to synthesize fluid and believable actions of the Smartakus agent. All planned deictic gestures of Smartakus must be synchronized with the graphical display of the corresponding media objects, so that Smartakus points to the intended graphical elements at the right moment. In addition, SMARTKOM's facial animation must be synchronized with the planned speech output. SMARTKOM's lip synchronization is based on a simple mapping between phonemes and visemes. A viseme is a picture of a particular mouth position of Smartakus, characterized by a specific jaw opening and lip rounding. Only plosives and diphthongs are mapped to more than one viseme.

One of the distinguishing features of SMARTKOM's modality fission is the explicit representation of generated multimodal presentations in M3L. This means that SMARTKOM ensures dialogue coherence in multimodal communication by following the design principle "no presentation without representation". The text generator provides a list of referential items that were mentioned in the last turn of the system. The display component generates an M3L representation of the current screen content, so that the discourse modeler can add the corresponding linguistic and visual objects to the discourse representation. Without such a representation of the generated multimodal presentation, anaphoric, crossmodal, and gestural references of the user could not be resolved. Thus, it is an important insight of the SMARTKOM project that a multimodal dialogue system must not only understand and represent the user's multimodal input, but also its own multimodal output.

Figure 9 shows the modality-free presentation goal that is transformed into the multimodal presentation shown in Fig. 10 by SMARTKOM's media fission component and unimodal generators and renderers. Please note that all the graphics and layout shown in Fig. 10 are generated on the fly and uniquely tailored to the dialogue

```
<presentationTask>
 <presentationGoal>
  <inform><informFocus>
   <RealizationType>list</RealizationType>
  </informFocus></inform>
  <abstractPresentationContent>
      <discourseTopic><goal>epg_browse</goal></discourseTopic>
      <informationSearch id="dim24"><tvProgram id="dim23">
          <broadcast><timeDeictic id="dim16">now</timeDeictic>
                <between>2003-03-20T19:42:32 2003-03-20T22:00:00</between>
            <channel><channel id="dim13"/></channel>
            < /broadcast></tvProgram>
        < /informationSearch>
    <result><event>
        <pieceOfInformation>
         <tvProgram id="ap_3">
           <broadcast><beginTime>2003-03-20T19:50:00</beginTime>
                        <endTime>2003-03-20T19:55:00</endTime>
                       <avMedium><title>Today's Stock News</title></avMedium>
                       <channel>ARD</channel>
           < /broadcast>.......</event>
    < /result>
< /presentationGoal>
< /presentationTask>
```

Fig. 9. A fragment of a presentation goal, as specified in M3L

situation, i.e., nothing is canned or preprogrammed. The presentation goal shown in Fig. 9 is coded in M3L and indicates that a list of broadcasts should be presented to the user. Since there is enough screen space available and there are no active constraints on using graphical output, the strategy operators applied by the presentation planner lead to a graphical layout of the list of broadcasts. In an eyes-busy situation (e.g., when the user is driving a car), SMARTKOM would decide that Smartakus should read the list of retrieved broadcasts to the user. This shows that SMARTKOM's modality fission process is highly context aware and produces tailored multimodal presentations.

The presentation planner decides that the channel should be rendered as an icon, and that only the starting time and the title of the individual TV item should be mentioned in the final presentation.

In the next section, we show how the visual, gestural and linguistic context stored in a multimodal discourse model can be used to resolve crossmodal anaphora. We will use the following dialogue excerpt as an example:

1. User: I would like to go to the movies tonight.
2. Smartakus: [displays a list of movie titles] This is a list of films showing in Heidelberg.

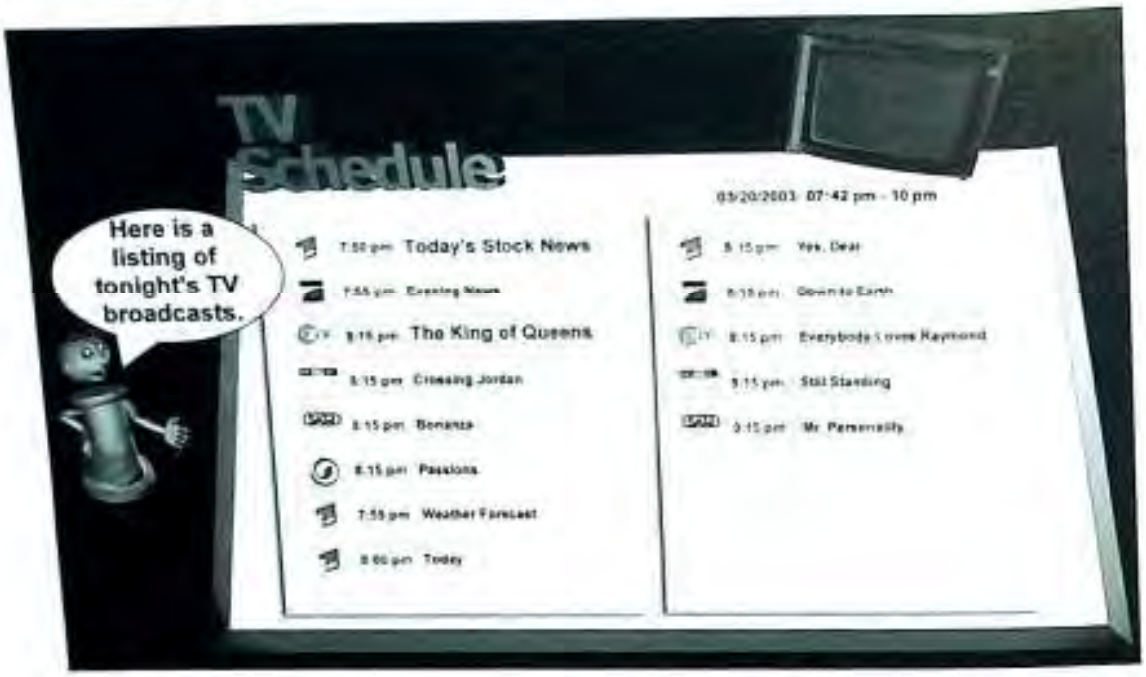

Fig. 10. A dynamically generated multimodal presentation based on a presentation goal

3. User: Hmm, none of these films seem to be interesting ... Please show me the TV program.
4. Smartakus: [displays a TV listing] Here [points to the listing] is a listing of tonight's TV broadcasts (see Fig. 10).
5. User: Please tape the third one!

9 A Three-Tiered Multimodal Discourse Model

Discourse models for spoken dialogue systems store information about previously mentioned discourse referents for reference resolution. However, in a multimodal dialogue system like SMARTKOM, reference resolution relies not only on verbalized, but also on visualized information. A multimodal discourse model must account for entities not explicitly mentioned (but understood) in a discourse, by exploiting the verbal, the visual and the conceptual context. Thus, SMARTKOM's multimodal discourse representation keeps a record of all objects visible on the screen and the spatial relationships between them.

An important task for a multimodal discourse model is the support of crossmodal reference resolution. SMARTKOM uses a three-tiered representation of multimodal discourse, consisting of a domain layer, a discourse layer, and a modality layer. The modality layer consists of linguistic, visual, and gestural objects that are linked to the corresponding discourse objects. Each discourse object can have various surface realizations on the modality layer. Finally, the domain layer links discourse objects with instances of SMARTKOM's ontology-based domain model (Löckelt et al.,

2002). SMARTKOM's three-tiered discourse representation makes it possible to resolve anaphora with nonlinguistic antecedents. SMARTKOM is able to deal with multimodal one-anaphora (e.g., "the third one") and multimodal ordinals ("the third broadcast in the list").

SMARTKOM's multimodal discourse model extends the three-tiered context representation of LuperFoy (1991) by generalizing the linguistic layer to that of a modality layer (see Fig. 11). An object at the modality layer encapsulates information about the concrete realization of a referential object depending on the modality of presentation (e.g., linguistic, gestural, visual). Another extension is that objects at the discourse layer may be complex compositions that consist of several other discourse objects (Salmon-Alt, 2001). For example, the user may refer to an itemized list shown on SMARTKOM's screen as a whole, or he may refer to specific items displayed in the list. In sum, SMARTKOM's multimodal discourse model provides a unified representation of discourse objects introduced by different modalities, as a sound basis for crossmodal reference resolution.

The modality layer of SMARTKOM's multimodal discourse model contains three types of modality objects:

- Linguistic Objects (LOs): For each occurrence of a referring expression in SMARTKOM's input or output, one LO is added.
- Visual Objects (VOs): For each visual presentation of a referrable entity, one VO is added.
- Gesture Objects (GOs): For each gesture performed either by the user or the system, a GO is added.

Each modality object is linked to a corresponding discourse object. The central layer of the discourse model is the discourse object layer. A Discourse Object (DO) represents a concept that can serve as a candidate for referring expressions, including objects, events, states and collections of objects. When a concept is newly introduced by a multimodal communicative act of the user or the system, a DO is created. For each concept introduced during a dialogue, there exists only one DO, regardless of how many modality objects mention this concept.

The compositional information for the particular DOs that represent collections of objects, is provided by partitions (Salmon-Alt, 2001). A partition provides information about possible decompositions of a domain object. Such partitions are based either on perceptual information (e.g., a set of movie titles visible on the screen) or discourse information (e.g., "Do you have more information about the first and the second movie?" in the context of a list of movie titles presented on the screen). Each element of a partition is a pointer to another DO, representing a member of the collection. The elements of a partition are distinguishable from one another by at least one differentiation criterion like their relative position on the screen, their size, or color. For instance, the TV listing shown in Fig. 10 is one DO that introduces 13 new DOs corresponding to particular broadcasts.

The domain object layer provides a mapping between a DO and instances of the domain model. The instances in the domain model are Ontological Objects (OO) that

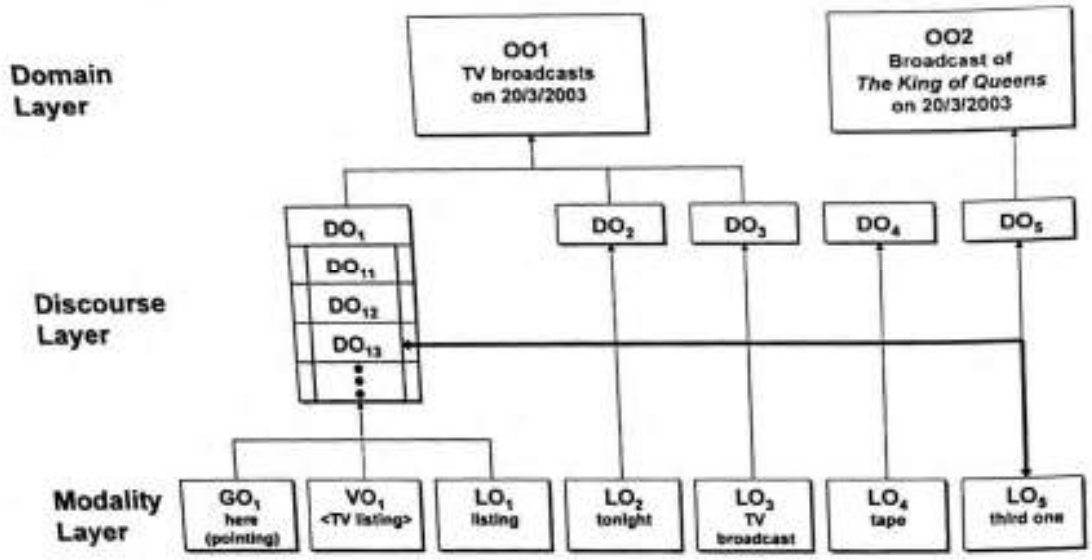

Fig. 11. An excerpt from SMARTKOM's multimodal discourse model

provide a semantic representation of actions, processes, and objects. SMARTKOM's domain model is described in the ontology language OIL (Fensel et al., 2001).

Let us discuss an example of SMARTKOM's methodology for multimodal discourse modeling. The combination of a gesture, an utterance, and a graphical display that is generated by SMARTKOM's presentation planner (see Fig. 10) creates the gestural object GO_1, the visual object VO_1 and the linguistic object LO_1 (see Fig. 11). These three objects at the modality layer are all linked to the same discourse object DO_1 that refers to the ontological object OO1 at the domain layer. Note that DO_1 is composed of 13 subobjects. One of these subobjects is DO_{13}, which refers to OO2, the broadcast of "The King of Queens" on 20 March 2003 on the ZDF channel. Although there is no linguistic antecedent for the one-anaphora "the third one", SMARTKOM can resolve the reference with the help of its multimodal discourse model. It exploits the information that the spatial layout component has rendered OO1 into a horizontal list, using the temporal order of the broadcasts as a sorting criterion. The third item in this list is DO_{13}, which refers to OO2. Thus, the cross-modal one-anaphora "the third one" is correctly resolved and linked to the broadcast of "The King of Queens" (see Fig. 11).

During the analysis of turn (3) in the dialogue excerpt above, the discourse modeler receives a set of hypotheses. These hypotheses are compared and enriched with previous discourse information, in this example stemming from (1). Although (3) has a different topic to (1) (it requests information about the cinema program, whereas (3) concerns the TV program), the temporal restriction (tonight) of the first request is propagated to the interpretation of the second request. In general, this propagation of information from one discourse state to another is obtained by comparing a current intention hypothesis with previous discourse states, and by enriching it (if possible) with consistent information. For each comparison, a score has to be computed re-

fleeting how well this hypothesis fits in the current discourse state. For this purpose, the nonmonotonic OVERLAY operation (an extended probabilistic unification-like scheme, see Alexandersson and Becker (2003)) has been integrated into SMARTKOM as a central computational method for multimodal discourse processing.

10 Beyond Restricted Domains: From SmartKom to SmartWeb

Although SMARTKOM works in multiple domains (e.g., TV program guide, telecommunication assistant, travel guide), it supports only restricted-domain dialogue understanding. Our follow-up project SmartWeb (duration: 2004–2008) goes beyond SMARTKOM in supporting *open-domain question answering* using the entire Web as its knowledge base.

Recent progress in *mobile broadband communication* and *semantic Web technology* is enabling innovative mobile Internet information services, that offer much higher retrieval precision than current Web search engines like Google or Yahoo!. The goal of the SmartWeb project (Reithinger et al., 2005) is to lay the foundations for multimodal interfaces to wireless Internet terminals (e.g., smart phones, Web phones, PDAs) that offer flexible access to various Web services. The SmartWeb consortium brings together experts from various research communities: mobile Web services, intelligent user interfaces, multimodal dialogue systems, language and speech technology, information extraction, and semantic Web technologies (see http://www.smartweb-project.org).

SmartWeb is based on the fortunate confluence of three major efforts that have the potential to form the basis for the next generation of the Web. The first effort is the *Semantic Web* (Fensel et al., 2003) which provides the tools for the explicit markup of the content of webpages; the second effort is the development of *semantic Web services* which results in a Web where programs act as autonomous agents to become the producers and consumers of information and enable automation of transactions. The third important effort is information extraction from huge volumes of rich text corpora available on the Web. There has been substantial progress in extracting named entities (such as person names, dates, locations) and facts relating these entities, for example "Winner[World Cup, Germany, 1990, Italy]", from arbitrary text.

The appeal of being able to ask a question to a mobile Internet terminal and receive an answer immediately has been renewed by the broad availability of always-on, always-available Web access, which allows users to carry the Internet in their pockets. Ideally, a multimodal dialogue system that uses the Web as its knowledge base would be able to answer a broad range of questions. Practically, the size and dynamic nature of the Web and the fact that the content of most webpages is encoded in natural language makes this an extremely difficult task. However, SmartWeb exploits the machine-understandable content of semantic webpages for intelligent question-answering as a next step beyond today's search engines. Since semantically annotated webpages are still very rare due to the time-consuming and costly manual markup, SmartWeb is using advanced language technology, information extraction

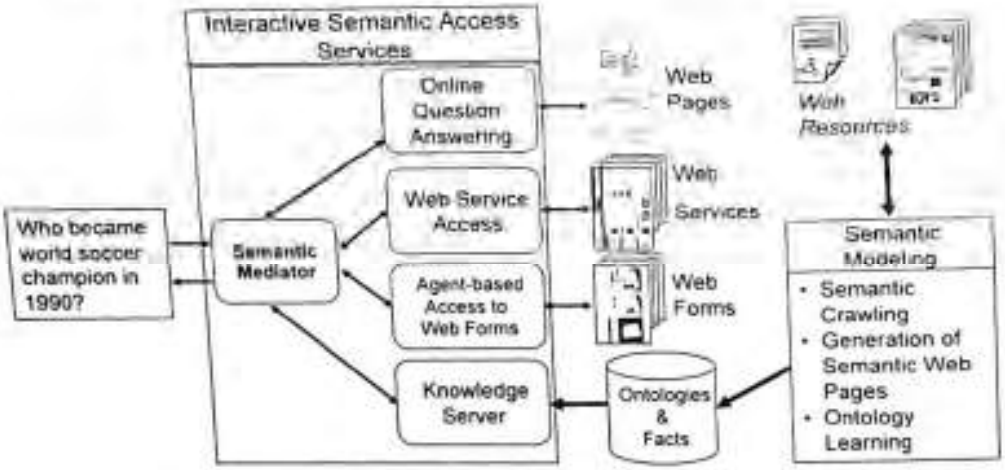

Fig. 12. The semantic mediator of SmartWeb

methods and machine learning for the automatic annotation of traditional webpages encoded in HTML or XML. SmartWeb generates such semantic webpages offline and stores the results in an ontology-based database of facts that can be accessed via a knowledge server (see Fig. 12). In addition, the semantic mediator of SmartWeb uses online question answering methods based on real-time extraction of relevant information from retrieved webpages.

But SmartWeb does not only deal with information-seeking dialogues but also with task-oriented dialogues, in which the user wants to perform a transaction via a Web service (e.g., program his navigation system to find the soccer stadium). Agent-based access to web forms allows the semantic mediator to explore the so-called Deep Web, including webbed databases, archives, dynamically created webpages and sites requiring login or registration.

SmartWeb provides a *context-aware user interface*, so that it can support the user in different roles, e.g., as a car driver, a motor biker, a pedestrian or a sports spectator. One of the demonstrators of SmartWeb is a personal guide for the 2006 FIFA World Cup in Germany, which provides mobile infotainment services to soccer fans, anywhere and anytime (see Fig. 13). The academic partners of SmartWeb are the research institutes DFKI (consortium leader), FhG FIRST, and ICSI together with university groups from Erlangen, Karlsruhe, Munich, Saarbrücken, and Stuttgart. The industrial partners of SmartWeb are BMW, DaimlerChrysler, Deutsche Telekom, and Siemens as large companies, as well as EML, Ontoprise, and Sympalog as small businesses. The German Federal Ministry of Education and Research (BMBF) is funding the SmartWeb consortium with grants totaling 13.7 million €.

11 The Roadmap for Multimodality

This book presents the foundations of multimodal dialogue systems using our fully fledged SmartKom system as an end-to-end working example. However, in this

Fig. 13. SmartWeb: open-domain and multimodal question-anwering

young research field many foundational questions are still open, so that intensive research on multimodality will be needed throughout the next decade. Our research roadmap for 2006–2010 shown in Fig. 14 outlines the research agenda for multimodality (Bunt et al., 2005).

Three "lanes" in the road identify three areas of research and development, including empirical and data-driven models of multimodality, advanced methods for multimodal communication and toolkits for multimodal systems. From 2006 to 2010, in the area of models of multimodality, we envision biologically inspired intersensory coordination models, test suites and benchmarks for comparing, evaluating and validating multimodal systems, and eventually computational models of the acquisition of multimodal communication skills, among other advancements. Advanced methods will include affective, collaborative, multiparty, and multicultural multimodal communication. Toolkits will advance from real-time localization and motion/eye tracking, to the incorporation of multimodality into virtual and augmented reality environments, and resource-bounded multimodality on mobile devices. The annual international conference on multimodal interfaces (ICMI) has become the premier venue for presenting the latest research results on multimodal dialogue systems (see, e.g., Oviatt et al. (2003)).

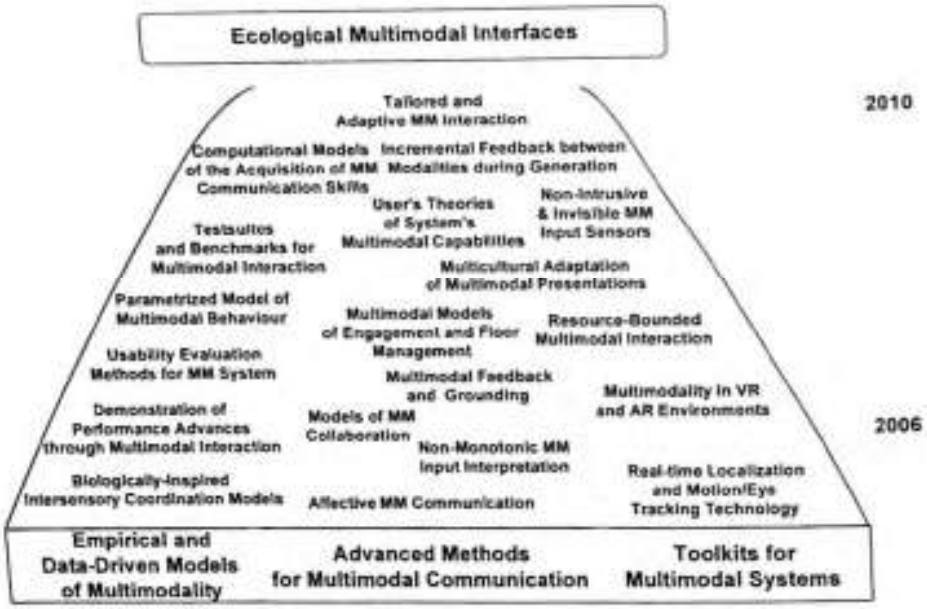

Fig. 14. Research roadmap for advanced multimodal systems

12 The Economic and Scientific Impact of SmartKom

The industrial and economic impact of the SMARTKOM project is remarkable. Up to now, 52 patents concerning SMARTKOM technologies have been filed by members of the SMARTKOM consortium, in areas such as speech recognition (13), dialogue management (10), biometrics (6), video-based interaction (3), multimodal analysis (2), and emotion recognition (2).

In the context of SMARTKOM, 59 new product releases and prototypes have surfaced during the project's life span. 29 spin-off products have been developed by the industrial partners of the SMARTKOM consortium at their own expense. The virtual mouse, which was invented by Siemens, is a typical example of such a technology transfer result. The virtual mouse has been installed in a cell phone with a camera. When the user holds a normal pen about 30 cm in front of the camera, the system recognizes the tip of the pen as a mouse pointer. A red point then appears at the tip on the display. For multimodal interaction, the user can move the pen and point to objects on the cell phone's display.

Former researchers from the SMARTKOM consortium have founded six start-up companies, including Sonicson, Eyeled, and Mineway.[1] The product spectrum of these companies includes multimodal systems for music retrieval, location-aware mobile systems, and multimodal personalization systems.

In addition to its economic impact, SMARTKOM has a broad scientific impact. The scientific results of SMARTKOM have been reported in 255 publications and 117 keynotes or invited lectures. During the project, six SMARTKOM researchers were

[1] http://www.sonicson.com, http://www.eyeled.com, and http://www.mineway.de

awarded tenured professorship. 66 young researchers have finished their master's or doctoral theses in the context of the SMARTKOM project.

SMARTKOM's MULTIPLATFORM software framework (see Sect. 6) is being used at more than 15 industrial and academic sites all over Europe and has been selected as the integration framework for the COMIC (COnversational Multimodal Interaction with Computers) project funded by the EU (Catizone et al., 2003). SMARTKOM's multimodal markup language M3L had an important impact on the definition of MMIL, which is now actively used in the ISO standardization effort towards a multimodal content representation scheme in ISO's Technical Committee 37, Subcommittee 4 "International Standards of Terminology and Language Resource Management". In addition, M3L had an obvious impact on the W3C effort towards a standard for a natural language semantics markup language (see http://www.w3.org/TR/nl-spec/).

The sharable multimodal resources collected and distributed during the SMARTKOM project will be useful beyond the project's life span, since these richly annotated corpora will be used for training, building, and evaluating components of multimodal dialogue systems in coming years. 448 multimodal Wizard-of-Oz sessions resulting in 1.6 terabytes of data have been processed and annotated (Schiel et al., 2002). The annotations contain audio transcriptions combined with gesture and emotion labeling.

Acknowledgments

The SMARTKOM project was made possible by funding from the German Federal Ministry of Education and Research (BMBF) under grant 01 IL 905. I would like to thank my SMARTKOM team at DFKI: Jan Alexandersson, Tilman Becker, Anselm Blocher (project management), Ralf Engel, Gerd Herzog (system integration), Heinz Kirchmann, Markus Löckelt, Stefan Merten, Jochen Müller, Alassane Ndiaye, Rainer Peukert, Norbert Pfleger, Peter Poller, Norbert Reithinger (module coordination), Michael Streit, Valentin Tschernomas, and our academic and industrial partners in the SMARTKOM project consortium: DaimlerChrysler AG, European Media Laboratory GmbH, Friedrich-Alexander University Erlangen-Nuremberg, International Computer Science Institute, Ludwig-Maximilians University Munich, MediaInterface GmbH, Philips GmbH, Siemens AG, Sony International (Europe) GmbH, and Stuttgart University for the excellent and very successful cooperation.

References

J. Alexandersson and T. Becker. The Formal Foundations Underlying Overlay. In: *Proc. 5th Int. Workshop on Computational Semantics (IWCS-5)*, pp. 22–36, Tilburg, The Netherlands, February 2003.

A. Batliner, R. Huber, H. Niemann, E. Nöth, J. Spilker, and K. Fischer. The Recognition of Emotion. In: W. Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 122–130, Berlin Heidelberg New York, 2000. Springer.

H. Bunt, M. Kipp, M. Maybury, and W. Wahlster. Fusion and Coordination for Multimodal Interactive Information Presentation. In: O. Stock and M. Zancanaro (eds.), *Multimodal Intelligent Information Presentation*, vol. 27 of *Text, Speech and Language Technology*, pp. 325–340, Berlin Heidelberg New York, 2005. Springer.

R. Catizone, A. Setzer, and Y. Wilks. Multimodal Dialogue Management in the CoMIC Project. In: *Proc. EACL-03 Workshop on "Dialogue Systems: Interaction, Adaptation and Styles of Management"*. Budapest, Hungary, April 2003. European Chapter of the Association for Computational Linguistics (EACL).

P.R. Cohen, M. Johnston, D. McGee, S.L. Oviatt, J.A. Pittman, I. Smith, L. Chen, and J. Clow. QuickSet: Multimodal Interaction for Distributed Applications. In: *Proc. 5th Int. Multimedia Conference (ACM Multimedia '97)*, pp. 31–40, Seattle, WA, 1997. ACM.

D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster (eds.). *Spinning the Semantic Web. Bringing the World Wide Web to Its Full Potential*. MIT Press, Cambridge, MA, 2003.

D. Fensel, F. van Harmelen, I. Horrocks, D.L. McGuinness, and P.F. Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

I. Gurevych, S. Merten, and R. Porzel. Automatic Creation of Interface Specifications from Ontologies. In: H. Cunningham and J. Patrick (eds.), *Proc. HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pp. 59–66, Edmonton, Canada, 2003. Association for Computational Linguistics.

G. Herzog, H. Kirchmann, S. Merten, A. Ndiaye, P. Poller, and T. Becker. MULTIPLATFORM Testbed: An Integration Platform for Multimodal Dialog Systems. In: H. Cunningham and J. Patrick (eds.), *Proc. HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pp. 75–82, Edmonton, Canada, 2003. Association for Computational Linguistics.

M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. MATCH: An Architecture for Multimodal Dialogue Systems. In: *Proc. 10th ACM Int. Symposium on Advances in Geographic Information Systems*, pp. 376–383, Washington, DC, 2002.

M. Löckelt, T. Becker, N. Pfleger, and J. Alexandersson. Making Sense of Partial. In: C.M. Johan Bos, Mary Ellen Foster (ed.), *Proc. 6th Workshop on the Semantics and Pragmatics of Dialogue (EDILOG 2002)*, pp. 101–107, Edinburgh, UK, September 2002.

S. LuperFoy. *Discourse Pegs: A Computational Analysis of Context-Dependent Referring Expressions*. PhD thesis, University of Texas at Austin, December 1991.

D.L. Martin, A.J. Cheyer, and D.B. Moran. The Open Agent Architecture: A Framework for Building Distributed Software Systems. *Applied Artificial Intelligence*, 13(1–2):91–128, 1999.

M.T. Maybury and W. Wahlster. Intelligent User Interfaces: An Introduction. In: M.T. Maybury and W. Wahlster (eds.), *Readings in Intelligent User Interfaces*, pp. 1–13, San Francisco, CA, 1998. Morgan Kaufmann.

S. Oviatt. Multimodal Interfaces. In: J.A. Jacko and A. Sears (eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pp. 286–304, Mahwah, NJ, 2003. Lawrence Erlbaum.

S.L. Oviatt, T. Darrell, M.T. Maybury, and W. Wahlster (eds.). *Proc. Int. Conf. on Multimodal Interfaces (ICMI'03)*, Vancouver, Canada, November 5–7 2003. ACM.

N. Pfleger, J. Alexandersson, and T. Becker. Scoring Functions for Overlay and Their Application in Discourse Processing. In: *Proc. KONVENS 2002*, pp. 139–146, Saarbruecken, Germany, September–October 2002.

N. Reithinger, S. Bergweiler, R. Engel, G. Herzog, N. Pfleger, M. Romanelli, and S. Sonntag. A Look Under the Hood — Design and Development of the First SmartWeb System Demonstrator. In: *Proc. Int. Conf. on Multimodal Interfaces (ICMI'05)*, pp. 159–166, Trento, Italy, 2005.

S. Salmon-Alt. Reference Resolution Within the Framework of Cognitive Grammar. In: *Int. Colloquium on Cognitive Science*, pp. 1–15, San Sebastian, Spain, May 2001.

F. Schiel, S. Steininger, and U. Türk. The SmartKom Multimodal Corpus at BAS. In: *Proc. 3rd Int. Conf. on Language Resources and Evaluation (LREC 2002)*, pp. 35–41, Las Palmas, Spain, 2002.

S. Seneff, R. Lau, and J. Polifroni. Organization, Communication, and Control in the Galaxy-II Conversational System. In: *Proc. EUROSPEECH-99*, pp. 1271–1274, Budapest, Hungary, 1999.

W. Wahlster. User and Discourse Models for Multimodal Communication. In: J.W. Sullivan and S.W. Tyler (eds.), *Intelligent User Interfaces*, pp. 45–67, New York, 1991. ACM.

W. Wahlster. SmartKom: Fusion and Fission of Speech, Gestures, and Facial Expressions. In: *Proc. 1st Int. Workshop on Man-Machine Symbiotic Systems*, pp. 213–225, Kyoto, Japan, 2002.

W. Wahlster, E. André, W. Finkler, H.J. Profitlich, and T. Rist. Plan-Based Integration of Natural Language and Graphics Generation. *Artificial Intelligence*, 63:387–427, 1993.

W. Wahlster, N. Reithinger, and A. Blocher. SmartKom: Multimodal Communication with a Life-like Character. In: *Proc. EUROSPEECH-01*, vol. 3, pp. 1547–1550, Aalborg, Denmark, September 2001.

W. Wahlster and R. Wasinger. The Anthropomorphized Product Shelf: Symmetric Multimodal Interaction with Instrumented Environments. In: E. Aarts and J. Encarnação (eds.), *True Visions: The Emergence of Ambient Intelligence*, Berlin Heidelberg New York, 2006. Springer.

Facts and Figures About the SmartKom Project

Anselm Blocher

DFKI GmbH, Saarbrücken, Germany
blocher@dfki.de

Summary. The SMARTKOM project *Dialogue-Based Human Computer Interaction by Coordinated Analysis and Generation of Multiple Modalities* was one of six lead projects in the area of human computer interaction funded by the Federal Ministry of Education and Research and the Federal Ministry of Economics and Labour of Germany. We describe the intention of this initiative and summarize the organizational and funding structure of the SMARTKOM project. The final functionalities of the demonstrator system are compiled as well as an overview of the reception of the project in the research community and in the media.

1 Lead Projects in Human Computer Interaction

In 1998 the Federal Government of Germany initiated an ideas competition for lead projects on human computer interaction. Based on the main interaction and communication means of humans including speech, gestures, facial expressions, haptic signals and visualization and integrated in a multimodal approach these projects were to develop new assistance systems and agents for intelligent information and knowledge processing.

The projects were to be strongly interdisciplinary and to generate both scientific impact for further research and application-oriented prototypes as demonstrators addressing a broader public. Inviting research partners from science and industry to participate in this kind of large strategic collaborative project was aimed at developing attractive scientific solutions with a high market potential and it resulted in a total of 89 project proposals. High emphasis was put on the usability and the ergonomics of the new forms of interaction and their user acceptance. Design and development of systems and prototypes were based on conducting preliminary user studies as well as accompanying and final evaluations.

Finally emerging out of the project ideas proposed, six major lead projects were selected by a jury and recommended for funding. Starting in mid-1999, the Federal Ministry of Education and Research and the Federal Ministry of Economics and Labour launched the four-year projects with a funding volume of 82.6 million €. The total funding requirement for a total of 102 project partners yielded 152.2 million €.

Covering a large range in human computer interaction the lead projects funded addressed additional but disjunct areas focussing on the following highlights:

- ARVIKA:
  Mobile systems for action in mixed real/virtual working environments. Practical concepts for the design of augmented reality systems
- EMBASSI:
  Intelligent user interfaces for consumer electronics. Individually adaptable access to public terminal systems for disabled persons
- INVITE:
  Combining intuitive visualization and personalized navigation, semantic search, automated classification and implicit recording of knowledge
- MAP:
  New ways of using and integrating computers in mobile activities. A wireless, networked pocket-size personal assistant
- MORPHA:
  Service robots for use in private homes and in production. Quick intuitive programming by pointing and showing what is to be done
- SMARTKOM:
  Human computer interaction in dialogues. Situation-based understanding of vague, ambiguous or incomplete multimodal input at the semantic and pragmatic levels

In parallel to the start of the lead projects a scientific advisory board with international membership was installed. The advisory board continuously monitored and evaluated the projects during the funding period and advised the two ministries on the steering decisions to be taken. The members of the advisory board were:

- Professor R. Reichwald, TU Munich (chairman)
- U. Klotz, IG Metall, Frankfurt/M. (SMARTKOM reviewer)
- M. Bartels, IPmotion GmbH, Marburg
- Professor P. Cohen, Oregon Graduate Institute, Beaverton, USA
- Professor M. Gross, ETH Zurich, Switzerland
- Dr. G.B. Hantsch, Deutsches Handwerksinstitut e.V., Karlsruhe
- Professor R. Hoffmann, TU Dresden
- Professor M.K. Lang (Ordinarius i.R.), TU Munich
- Professor S. Maaß, University of Bremen (SMARTKOM reviewer)
- Dr. M. Maybury, MITRE Corporation, Bedford, USA
- Professor A. Ourmazd, Communicant AG, Frankfurt/Oder
- Dr. J. Redmer, Remshalden-Grunbach
- Professor G. Rigoll, TU Munich
- Professor J. Sauter, ART+COM AG, Berlin
- P. Zoche, Fraunhofer-ISI, Karlsruhe

For a comprehensive overview of the six lead projects on human computer interaction see Krahl and Günther (2003).

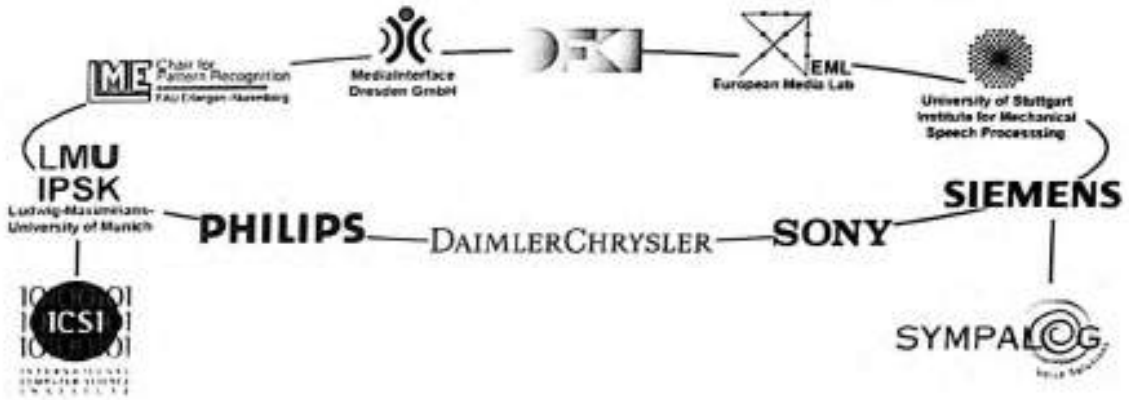

Fig. 1. The SMARTKOM consortium

2 SmartKom: A Lead Project in Human Computer Interaction

The SMARTKOM project was designed as a four-year initiative. The network plan was set up in spring 1999 defining nine subprojects with a total of 70 workpackages. Within the twelve research groups (see next section) a total of 218 person years were allocated. The project started September 1, 1999 and ended September 30, 2003.

2.1 The SmartKom Partners

The SMARTKOM consortium consisted of twelve partners, integrating research centers, large-scale industry, SME and universities (see Fig. 1).

- **Research centers:**
  - German Research Center for Artificial Intelligence GmbH, Kaiserslautern and Saarbrücken (DFKI, Main Contractor)
  - European Media Laboratory GmbH, Heidelberg (EML)
  - International Computer Science Institute, Berkeley/USA (ICSI)
- **Large-scale industry:**
  - DaimlerChrysler AG, Stuttgart (DCAG)
  - Philips Speech Processing, Aachen
  - Siemens Aktiengesellschaft, Munich
  - Sony International (Europe) GmbH, Stuttgart
- **SME:**
  - MediaInterface Dresden GmbH (MID)
  - Sympalog Voice Solutions GmbH, Erlangen
- **Universities:**
  - Chair for Pattern Recognition, Friedrich-Alexander-University Erlangen-Nuremberg (FAU)
  - Institute for Natural Language Processing, University of Stuttgart (IMS)
  - Institute of Phonetics and Speech Communication, Ludwig-Maximilians-University Munich (LMU)

2.2 Funding and Controlling

The SMARTKOM project was funded by the German Federal Ministry of Education and Research (BMBF, see Fig. 2, grant no. 01 IL 905) and additionally financed by the industrial and SME partners. Until the year 2003, 16.8 million € was allocated to the project. In addition the industrial and SME partners brought in 8.9 million € (see Table 1). Universities and research centers received a full 100% funding, while industrial partners contributed 60% of their costs.

Table 1. Funding

BMBF-Funding, 01.09.1999 – 30.09.2003	16.8 Mio. €
Industrial and SME investment	8.9 Mio. €
Overall volume	**25.7 Mio. €**

The project was controlled by the German Aerospace Research Establishment (DLR), Berlin (see Fig. 3).

SPONSORED BY THE

Fig. 2. German Federal Ministry of Education and Research

Fig. 3. German Aerospace Research Establishment

2.3 Management and Organization Structure

The SMARTKOM project was jointly managed by the Scientific Management, the SMARTKOM Steering Committee and the Group of Module Managers (see Fig. 4). The Scientific Management organised, coordinated and supervised the project and evaluated the progress of the implementation in close contact with the funding agency.

The assignment of the SMARTKOM Steering Committee was to support the Scientific Management in ensuring the scientific excellence of the work. The SMARTKOM Steering Committee consisted of ten project partner leaders to adjust the project

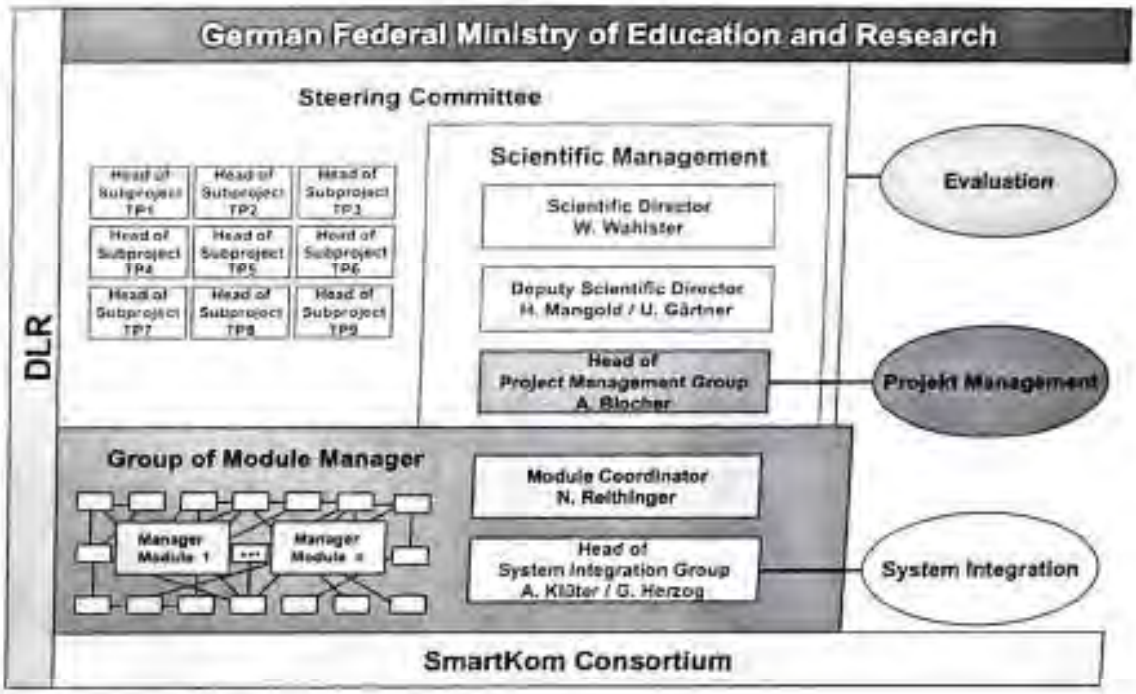

Fig. 4. The management structure of the SMARTKOM Project

plan to new scientific results or trends in HCI technology and to react immediately to problems that arose in the development and implementation process. A total of 15 meetings (including the constituent one) were held. Within the SMARTKOM Steering Committee there were assigned responsibilities for subprojects and working areas (WA) (see Table 2).

Table 2. Subprojects and working areas

Subproject/WA	Title	Lead partner
1	Modality specific analysis	DCAG
2	Multimodal interaction	FAU
3	Application interface	Sony
4	Genaration and multimodal media design	IMS
5	SMARTKOM-Public	Siemens
6	SMARTKOM-Mobile	EML
7	SMARTKOM-Home/Office	Philips
8	System integration	DFKI
9	Project management	DFKI
WA	Evaluation	LMU
WA	Technical integration	MID

The main task of the Group of Module Managers was to tie together and bundle the distributed development forces. In the meetings of the Group of Module Man-

Table 3. Large project workshops

Date	Topic	Place
11.-13.10.1999	Data collection	Munich
17.-18.11.1999	System integration	Kaiserslautern
29.-30.11.1999	Domains	Aachen
10.-11.04.2000	System integration	Kaiserslautern
11.-12.05.2000	Interpretation and ontology	Stuttgart
08.-09.08.2000	Module coordination	Stuttgart
14.12.2000	Module coordination	Munich
14.-15.03.2001	System integration	Kaiserslautern
18.-19.04.2001	Function modeling	Stuttgart
26.06.2001	Module coordination	Ulm
09.-10.10.2001	System integration	Kaiserslautern
13.11.2001	Module coordination	Saarbrücken
15.01.2002	Module coordination	Stuttgart
10.-11.04.2002	Domain ontology/XML	Saarbrücken/Heidelberg
18.-19.06.2002	System integration	Kaiserslautern
18.02.2003	Module coordination	Kaiserslautern
19.05.-20.05.2003	UserState processing	Saarbrücken

agers all important decisions concerning interface definitions, offline data flow, delivery schedules, and integration cycles were prepared, discussed and communicated. The implementation process was tightly monitored and supervised. It also provided contact points for the Scientific Management and gave feedback to the SMARTKOM Steering Committee.

Beside numerous bilateral and multilateral workshops, 17 large project meetings were held to discuss the scientific matters, to fine-tune details of the realization process, and to synchronize the inter-module communication (see Table 3).

Table 4. Project steering and review meetings

Project steering meetings	
10.05.2000	Stuttgart
22.05.2001	Heidelberg
10.05.2002	Heidelberg

Project review meetings	
13.12.2000	Munich
18.12.2002	Saarbrücken
05.09.2003	Stuttgart

2.4 Reviews and HCI Events

The SMARTKOM reviewers and the BMBF/DLR were informed in three milestone review meetings and in three steering meetings about the progress of the project (see Table 4). During these large project meetings the results achieved as well as the ongoing and planned work were presented in a total of 51 scientific talks with more

than 1000 slides. In the accompanging demonstration sesssions the current SMARTKOM systems were shown and industrial SMARTKOM spin-offs of the partners were first presented.

In the framework of the six HCI lead projects two International Status Conferences were held to reach a high degree in visibility both in the international research community and in the media. In order to optimize the cooperation with and to strengthen the understanding of the HCI partner projects several workshops on topics of general interest were organized (see Table 5).

Table 5. HCI events

Date	HCI event
28.09.1999	HCI Start Meeting
29.09.2000	WS Multimodal Interaction and Modeling
29.01.2001	HCI Congress
22.02.2001	WS Domain Modeling and Ontologies
17.05.2001	WS Architecture
03.07.2001	WS Tracking
25.09.2001	WS Adaptivity
26.10.2001	Int. Status Conference Lead Projects "Human Computer Interaction" 2001
25.02.2002	WS Usability
07.03.2002	WS Benchmarking
03.-04.06.2003	Int. Status Conference Lead Projects "Human Computer Interaction" 2003

3 The SmartKom System

3.1 Milestones and Final System Functionalities

The first important milestone, the first fully integrated SMARTKOM demonstrator, was reviewed in December 2000 (13.12.2000). This first SMARTKOM demonstrator was unveiled to the public at the national conference Mensch&Computer 2001 (04.-08.03.2001). The first English SMARTKOM demonstrator was presented at the international conference Eurospeech (03.-07.09.2001, see Wahlster et al. (2001)). The first SMARTKOM demonstrator system covered basic functionalities in the SMARTKOM-Public scenario and worked in the cinema reservation domains: selection of cinemas, information on movies, and seat reservation.

The final SMARTKOM demonstrator was presented in June 2003 at the HCI Status Conference (03.-04.06.2003; scenarios Public and Home/Office) and during Mensch&Computer 2003 (07.-10.09.2003; scenario Mobile). At CeBIT'04 (18.-24.03.2004) all three scenarios were shown. The final SMARTKOM demonstrator allows the user symmetric multimodal interaction in 14 applications with 52 functionalities (see Table 6). An exemplary interaction with the SMARTKOM system can be found in Reithinger and Herzog (2006). For technical data of the SMARTKOM demonstrator see Herzog and Ndiaye (2006).

Table 6. Multimodal addressable functionalities in SMARTKOM

Home			
EPG (Electronic-Programming Guide)	General program	Information for one single broadcast	
	Channel selection	Time-based operations	
	Channel information	Help functions for genres	7
	Selection based on genre		
TV	On/off	Channel selection	2
VCR control	On/off	Wind/rewind	
	Record	Programming using EPG and the calendar	
	Play		
	Pause		6
Lean-Forward/ Lean-Backward	Select Lean-Backward	Context-aware presentations	
	Deactivate Lean-Backward		3
Total Home			**18**
Public			
Telephone	Manipulative key operations	Audio handling	
	Telephony functions	Address book	4
Hand contour biometry	Selection of biometry type	Presentation and camera control	
	Hand contour biometry	Address book (see above)	3
Voice biometry	Presentation and audio control	Address book (see above)	
	Voice biometry	Selection of biometry type (see above)	2
Signature biometry	Presentation and tablet control	Address book (see above)	
	Signature biometry	Selection of biometry type (see above)	2
Fax	Presentation and interaction	Address book (see above)	
	Fax handling	Camera control	3
E-mail	Presentation and interaction	Address book (see above)	
	E-mail handling	Camera control (see above)	2
Cinema	General program	Seat reservation	
	Movie information	Cinema location	4
Total Public			**20**
Mobile			
Car navigation	Selection of start and goal city	Selection of parking garage	
	Route type selection	Information about parking garages	5
	Car route computation		
Pedestrian navigation	Selection of map type	Selection of points of interest	
	Selection of start and goal	Information for points of interest	
	Route computation	Integrated car and pedestrian route planning	6
Map manipulation	Resize	Help functions for map interactions	
	Change viewpoint		3
Total Mobile			**14**
Total SMARTKOM			**52**

The economic and scientific impact of the SMARTKOM project is described in Wahlster (2006). A list of SMARTKOM-related scientific papers, keynotes, talks and presentations is available at the web site http://www.smartkom.org.

3.2 Public Relations

SMARTKOM has been shown at two press conferences and in various presentations to a large number of representatives of science, industry and politics (see Table 7).

Table 7. Press conferences and highlight presentations

Press conferences	
14.12.2000	Munich
05.09.2003	Stuttgart
Highlight presentations	
03.-07.09.2001	EUROSPEECH'01
22.05.2002	Dr. Angela Merkel (Party President CDU)
22.08.2002	Josef Brauner (Deutsche Telekom AG)
	Saarland Minister for Economic Affairs and Labour Dr. Hanspeter Georgi
26.08.2002	German Federal President Johannes Rau
	Minister President of the Saarland Peter Müller
	Saarland Minister for Education, Culture and Science Jürgen Schreier
	President of Saarland University Prof. Margret Wintermantel
17.-20.09.2002	ICSLP'02
07.01.2003	EU executive Mr. Horst Forster
24.09.2003	Indian State Secretary of the Department of Science and Technology Prof. V. S. Ramamurthy
03.12.2003	Information Day "Mensch-Technik-Interaktion: Impulse für den Maschinenbau" of the German Engineering Federation (VDMA)
18.-24.03.2004	CeBIT'04
11.05.2004	German Language Technology Congress "LT Summit" at DFKI
29.-30.06.2004	Congress and Exposition "Empower Deutschland"
15.04.2005	Jürgen Gallmann,
	General Manager, Microsoft Germany, Vice President, Microsoft EMEA
	Dr. Said Zahedani, Director Developer Platform & Strategy Group
	Pierre-Yves Saintoyant, Director European Microsoft Innovation Center
	Walter Seemayer, NTO Microsoft Germany

In addition the SMARTKOM system has been demonstrated — amongst others — to representatives of: Adam Opel AG, ALCATEL, ATR Spoken Language Communication Research Labs, Bentley Motors, BMW AG, CapInfo, FUJI RIC, Harvard University, Hitachi Central Research Lab, NTT DoCoMo and DoCoMo Euro-Labs, NTT (Nomura Research Institute), RICOH (Japan), Robert Bosch GmbH, Siemens AG, Siemens Medical Solutions, Volkswagen Autostadt and Volkswagen Research.

The project was covered in several scientific or news TV programs and far more than 100 relevant press articles were published about SMARTKOM (see Table 8).

Table 8. Selected TV programs and press articles

Selected TV programs	
11.03.2001	3sat "hitec"
21.04.2001	N-TV
29.11.2001	zdf "heute"
20.01.2002	RTL "SPIEGEL TV Magazin"
19.08.2002	RTL "future Trend"
29.09.2002	ZDF "heute"
05.09.2003	Saarländischer Rundfunk "Aktueller Bericht"
03.02.2004	ZDF "heute-journal"
08.10.2004	3sat "Nano - Die Welt von Morgen"

Selected press articles	
03.03.1999	Saarbrücker Zeitung *Bonn fördert Saarbrücker Forscher mit Millionen*
15.03.1999	c't *Rechner ohne Tastatur und Maus*
22.03.1999	Computerworld *Nur wegbeamen kann man sich nicht*
06.10.1999	Technisch Weekblad *Spraakgestuurde databankprojecten*
30.01.2001	Die Welt *Eine Maschine versteht Gesten des Menschen*
19.02.2001	Handelsblatt *Sprechen und Zeigen statt Tippen*
23.03.2001	VDI nachrichten *Maschinen lesen Wünsche von den Augen ab*
01.06.2001	c't-News *Maschine versteht Sprache, Gesten und Augenbewegungen*
07.06.2001	ComputerZeitung *Mimik und Augenbewegungen steuern Computer*
04.02.2002	CHIP *Er hört aufs Wort*
12.03.2002	Frankfurter Allgemeine Zeitung *Sprich mit mir*
14.03.2002	Financial Times Deutschland *Verständige Maschinen*
22.04.2002	Salzburger Nachrichten Online *Deutscher Zukunftspreisträger entwickelt einen Computer-Butler ohne Maus und Tastatur*
06/2002	Spektrum der Wissenschaft *Redselige Chips*
27.08.2002	Saarbrücker Zeitung *Mit dem Bus zum roten Teppich*
29.09.2002	Handelsblatt *Maschinen, die Menschen besser verstehen*
08.10.2002	Süddeutsche Zeitung *Grimassen für den Computer*
26/2002	c't *Talkmaster*
05.03.2003	Hindustan Times *Talking computer in the making*
14.04.2003	i-com - Zeitschrift für interaktive und kooperative Medien *SmartKom - Multimodale Mensch-Technik-Interaktion*
10.06.2003	Berliner Zeitung *Verständnisvolle Computer*
13.06.2003	VDI Nachrichten *Computer und Roboter lernen Menschen zu verstehen*
16.06.2003	c't *"Wenn wir schreiben Seit' an Seit' ..."*
07.07.2003	dpa *Die Technik soll den Menschen verstehen lernen*
15.09.2003	Computerzeitung *Intelligente Schnittstellen erhöhen Produktakzeptanz*
06.11.2003	Handelsblatt *Computer reagieren künftig auch auf Fingerzeig*
28.01.2004	Saarbrücker Zeitung *Der Computer gehorcht aufs Wort / Das Handy lernt sehen*
13.03.2004	Handelsblatt *Elektronische Sekretärinnen erkennen Gesten*
07.04.2004	Die Zeit *Und wie viele Chips hat Ihrer?*
13.05.2004	Saarbrücker Zeitung *Mit den Maschinen reden lernen*

To give a deeper insight into the final SMARTKOM system, its scenarios, applications and functionalities and into multimodal interaction in general terms, videos — in English and German — can be downloaded from the SMARTKOM web site http://www.smartkom.org:

- SMARTKOM Complete: Covering all three scenarios Public, Home and Mobile (MPEG, ≈ 170 MB)
- SMARTKOM Mobile: Special long version of the Mobile scenario (MPEG, ≈ 67 MB)

Fig. 5. The SMARTKOM logo

References

G. Herzog and A. Ndiaye. Building Multimodal Dialogue Applications: System Integration in SmartKom, 2006. In this volume.

R. Krahl and D. Günther (eds.). *Proc. Human Computer Interaction Status Conference 2003*, Berlin, Germany, June 2003. DLR.

N. Reithinger and G. Herzog. An Exemplary Interaction with SmartKom, 2006. In this volume.

W. Wahlster. Dialogue Systems Go Multimodal: The SmartKom Experience, 2006. In this volume.

W. Wahlster, N. Reithinger, and A. Blocher. SmartKom: Multimodal Communication with a Life-like Character. In: *Proc. EUROSPEECH-01*, vol. 3, pp. 1547–1550, Aalborg, Denmark, September 2001.

An Exemplary Interaction with SmartKom

Norbert Reithinger and Gerd Herzog

DFKI GmbH, Saarbrücken, Germany
{reithinger,herzog}@dfki.de

Summary. The different instantiations of the SMARTKOM demonstration system offer a broad range of application functions and sophisticated dialogue capabilities. We provide a first look at the final SMARTKOM prototype from the point of view of the end user. In particular, a typical interaction sequence is presented in order to illustrate the functionality of the integrated multimodal dialogue system.

1 Introduction

The three different usage scenarios of SMARTKOM and its various application functions allow for a wide range of possible interactions. Aiming at flexible and natural multimodal dialogues, we need to define the intended behaviour of the system as well as its general look and feel. In particular, we have to lay out basic dialogue steps, which the user can freely combine during his or her interaction with the system. This design task leads to an iterative process, which takes the initial project definition as a starting point. System developers, scenario experts and prospective users (e.g., through wizard-of-oz experiments) collaborate to design and refine the capabilities of the multimodal system to be build. Following the paradigm of scenario-based design (Carroll, 2000; Rosson and Carroll, 2003), the initial focus of the design activity is not a formal functional specification but a description of how people will use the system to accomplish work tasks and other activities.

In the next section, we first present the dialogue descriptions we use to define and document the basic interactions in SMARTKOM as well as the illustrative dialogue protocols that can be generated automatically from the extensive log data resulting from a system run. Then we walk through an original sample dialogue between a test user and the system prototype to provide an insight into the capabilities of the SMARTKOM demonstrator. A presentation on paper, of course, can only provide a rough sketch of a multimodal interaction. The SMARTKOM Web site, located at `http://www.smartkom.org`, provides a comprehensive video that complements the description of the integrated SMARTKOM prototype given here.

2 From Dialogue Drafts to Dialogue Protocols

For a large, distributed project it is necessary to coordinate the design and development efforts on various levels. One important task is to agree on those dialogue steps and discourse phenomena that the system should be able to process. At the beginning, the different scenario experts performed user studies (see, e.g., Horndasch et al. (2006)). The goal was to come up with preferred interaction metaphors and interaction sequences that should be realised. In addition, the wizard-of-oz data collection (Schiel and Türk, 2006) provided important insights concerning natural interaction sequences.

With this in mind, we designed a template for dialogue descriptions, which were the basis for the communication between interface designers and scenario experts. The idea was to collect all relevant information for turns, i.e., input/output sequences between end user and system, and turn sequences that were to be realised.

Figure 1 shows a typical excerpt from such a document and presents one turn for the English Mobile scenario. A turn description starts with possible inputs from the user. The list of utterances is not exhaustive and contains category types like names. The explanatory notes section below contains remarks about the processing of these input utterances. In the example, it contains all entries of the English base lexicon for sights in the town of Heidelberg. This is also the place to document limitations of the system. The turn description ends with possible verbal system reactions and example screenshots.

Initially, we took the interaction descriptions as defined by the scenario experts and wrote one document for each functionality. The documents have been made available via the SMARTKOM intranet for project-wide discussion and potential enhancements. During the realization phase, the information in the descriptions was further augmented with real processing results from the system. The scenario experts, who sometimes had no immediate access to the latest version of the development system, could then comment on the results and provide feedback. To ensure consistency, the head of the system integration group was in charge of all additions and changes.

Of course, it is difficult to argue about a dialogue-based interaction unless you have a video of the interaction or some other sort of protocol. The SMARTKOM testbed (Herzog et al., 2004) is able to trace all data communication between the various modules of the system. This option is very useful during system development to debug the system on various levels. As the log contains the results from speech recognition, the modality analysis components and the presentation modules, it provides all information necessary to automatically create a protocol of a particular interaction sequence. Even though it does not contain the animations, important changes in the screen display and the final screenshot are included in the log file and can be extracted together with the textual output of the system. The information in the trace file is encoded in M3L (Herzog and Reithinger, 2006; Herzog and Ndiaye, 2006), the XML-based language that is used for data exchange between SMARTKOM software components. XSLT style sheets (see, e.g., Gardner and Rendon (2001)) are employed to automatically extract and format the relevant information from an interaction log.

SmartKom Dialog Description english.doc
(Gerd Herzog Page 5 of 18 12.12.2003)

Example dialog for the English demonstrator, which is based on the mobile scenario with integrated route planning.

SMA: = SmartKom system, *[system action]* , **USR:** = User, *<user action>*

6. Turn

USR:
a) **Where is a *Category*?**
b) **Show me a *Category*.**
c) **Show me *Category*.**
d) **What *Category* are there?**
e) **Show me a map with *Category*, please.**
f) **I am looking for a *Category*.**
g) **I m looking for a *Category*.**

Non-exhaustive list of possible user utterances

Explanatory Notes:
All objects of the given type will be shown on the map. In addition to [illegible] types, specific location names may be used as well.
The English base lexicon already contains the following sights: *Philosophy Bridge, Saint Peters Church, Grain Market, Palais Morass, Castle* and the following names for object categories: *church, square, cinema, museum, café, pub, park, theater, garage, parking, restaurant, churches, squares, cinemas, museums, cafés, pubs, parks, restaurants, theaters, garages*.

Remarks about processing

SMA:
a) **Here is a map with *Category*.**
b) **There is no information. Here is the city map.**

Non-exhaustive list of system reactions

Screen: *[Presents city map, with marked objects where applicable]*

Example screenshot

Additional Notes:

Fig. 1. An example page of the English dialogue description

As a result, a browsable HTML page and a self-contained document in PDF format are created, which contain the condensed information for the specific interaction. This documentation of an interaction sequence is used to discuss the implemented dialogue functionality and may initiate a new development iteration, which can also lead to changes in the corresponding dialogue description.

Figure 2 shows a page of the dialogue protocol for the example dialogue presented in the next section. Each entry starts with the internal message number of the corresponding data exchange. On the top of the page, there are two screenshots which are the result of the previous user interaction, where the user asked for the cinema program. The next message in the protocol contains the best hypothesis of the speech recogniser, followed by the word chain that has been selected as the basis for input interpretation in the system. The chosen hypothesis can be different from the best chain since the speech analysis module is able to parse the complete word lattice in order to select the most appropriate semantic interpretation. The next entry represents the derived user intention, which is used by the action planning component to select a suitable system reaction. In this case, the style sheet transformations create a compact predicate-argument structure from the original M3L description, which condenses the sometimes rather lengthy XML markup. Finally, the system output is printed as a text string, followed by the final screenshot of the animated presentation.

In the end, the final dialogue descriptions and meaningful protocols of current interactions provide a comprehensive overview of the detailed capabilities of the multimodal dialogue system.

3 An Extended Example Dialogue

In this section, we will walk through an example dialogue to provide the reader with a feeling of how the actual system works. As the public information kiosk integrates most of the functionality and features of SMARTKOM (Berton et al., 2006; Horndasch et al., 2006; Malaka et al., 2006; Portele et al., 2006), the presentation will use the scenario SMARTKOM-Public instead of the English Mobile system (Gelbart et al., 2006). The dialogue interaction in SMARTKOM-Public is based on German, so we also provide English translations.

The assumed location of the system installation is in the main railway station in the city of Heidelberg, and the current date is September 6, 2003. Imagine a user who just arrived in Heidelberg. She accesses the system and is presented with the initial display (see Fig. 3, left). To activate the system, the user can, for example, place a hand in the focus of the gesture recognition camera.

The interaction agent "Smartakus" appears on the screen (see Fig. 3, right) and greets the user:

(1) SYS: Herzlich willkommen beim SMARTKOM-System. Ich bin Smartakus wie kann ich Ihnen helfen?
(Welcome to the SMARTKOM *information system. I am Smartakus. How may I help you?)*

SmartKom Dialog Protocol

0000001200

0000001303 **#PAUSE# welche Filme mit Arnold_Schwarzenegger laufen Rennen #PAUSE#**

0000001332 **#PAUSE#**

0000001346 welche Filme mit Arnold_Schwarzenegger laufen Rennen.. (0.916)
welche Filme mit Arnold_Schwarzenegger laufen Rennen.. (0.916)

0000001433 *informationSearch pieceOfInformation (cinemaProgram (performance (performance (beginTime (time (analysis (instant (day (dayDeictic='today') daytime='evening')) function (between (from='2003-09-05T18:00:00' to='2003-09-05T23:59:59'))) avMedium (avMedium (cast (cast (member (actor (name (unparsedName='Arnold Schwarzenegger')))))) cinema (cinema))))*

0000001553 hier sehen Sie die Filme die heute abend laufen

0000001646

SmartKom Dialog Protocol 2

Fig. 2. A page of the interaction protocol for the example dialogue

Fig. 3. Start screens of the SMARTKOM system

Fig. 4. Smartakus presenting the evening program and a map of the cinemas in Heidelberg

The user wants to go to the movies tonight and therefore asks:

(2) USR: Was kommt heute Abend im Kino?
(What is playing at the cinema tonight?)

Figure 4 shows the processing results: On the left hand-side is the cinema program, and on the right-hand side is a city map of Heidelberg with cinema locations. Note that SMARTKOM anticipated that the user might be interested to know where

the cinemas are. This is particularly important, if, for example, a movie is playing at various theatres. Smartakus provides as spoken information:

(3) SYS: Hier sehen Sie die Filme die heute abend laufen. Auf dieser Karte sind die Kinos markiert.
(These are the movies playing tonight. The cinemas have been marked on the map.)

Our user recalls that there is a new movie with Arnold Schwarzenegger, so she asks:

(4) USR: Welche Filme mit Arnold Schwarzenegger laufen denn?
(Are there any movies featuring Arnold Schwarzenegger?)

It should be noted that the words "Arnold" and "Schwarzenegger" are not in the base lexicon of the system. The result of a database query—in this case movie titles and names of actors—may contain words that are not in the lexicon of the speech recognizer or the speech synthesis component. Hence, those modules have to be informed about possible words the user or the system might utter when presenting or addressing the titles, actors and other information. In SMARTKOM, the dedicated lexicon module (Goronzy et al., 2006) creates the phonetic transcriptions for new terms—since titles are often in foreign languages it has to provide pronunciation variants—and propagates the added lexicon entries to the speech modules.

The user query results in an information display that contains only movies with Arnold Schwarzenegger, as shown in Fig. 5 on the left side. The system utters:

(5) SYS: Hier sehen Sie die Filme die heute abend laufen.
(These are the movies playing tonight.)

Now the user employs multimodality and, while pointing to one of the movie tiles, says:

(6) USR: Gib mir Informationen zu diesem.
(Give me information about this one.)

SMARTKOM replaces the movie information frame and displays the description for this specific film (see the enlarged display in Fig. 5, on the right), telling the user:

(7) SYS: Ich habe diese Informationen zu "Terminator 3 – Rebellion der Maschinen".
(I have got this information about "Terminator 3: Rise of the Machines".)

The user wants to visit this cinema showing with her friend and tells Smartakus:

(8) USR: Bitte 2 Karten für diesen Film reservieren.
(Please reserve 2 seats for this film.)

In the context of the ongoing dialogue it is not ambiguous which performance she addresses, and the system presents the floor plan of the selected cinema (see Fig. 6, on the left), asking:

Fig. 5. Information about the film *Terminator 3*

Fig. 6. Floor plan of the cinema

(9) SYS: Zeigen Sie mir wo Sie sitzen wollen.
(Show me where you want to sit.)

The user approximately encircles the desired area and says:

(10) USR: Dort möchte ich sitzen.
(I want to sit there.)

Again, the discourse context provides the information that she wants two seats and thus a proposal can be made. Smartakus marks two convenient seats (see Fig. 6, on the right) and asks:

(11) SYS: Ist das so richtig?
(Is this correct?)

The user is convinced, agrees and gets the reservation number from Smartakus.

(12) USR: Ja, genau.
(Yes, perfect.)

(13) SYS: Ich habe die gewünschte Reservierung für Sie durchgeführt. Ihre Reservierungsnummer ist 20. Sie können die Karten bis spätestens eine halbe Stunde vor Vorstellungsbeginn an der Kinokasse abholen.
(I reserved your seat. Your reservation number is 20. Please get your tickets not later than 30 minutes before the movie starts.)

Now she wants to know the route to the cinema and asks:

(14) USR: Wie komme ich von hier zum Kino "Studio Europa"?
(Can you show me the way from here to "Studio Europa" cinema?)

Fig. 7. Pedestrian route from main station to cinema

Originally, the functionality for pedestrian navigation was conceived for the scenario SMARTKOM-Mobile. The consistent, knowledge-based approach of SMARTKOM, however, facilitated the additional inclusion of this application in other scenarios. So, later on, the pretrip planning was simply added to SMARTKOM-Public. The system knows its own location from its general context knowledge source and is

thus able to resolve the adverb "here". The public information kiosk is located at the main station in Heidelberg, and SMARTKOM can compute the pedestrian route from there to the selected cinema. Figure 7 shows the map with the route inserted. Since the computation of the route takes some time, Smartakus also informs the user to be patient and says:

(15) SYS: Die Route wird berechnet. Hier sehen Sie die Karte mit der Route von Hauptbahnhof zu Kino "Studio Europa".
(The route is being computed. Here you can see the map with the route from main station to cinema "Studio Europa".)

After she has taken care of today's evening entertainment, our user recalls that there is an important document that needs to be sent to her director. She tells Smartakus:

(16) USR: Ich will ein Dokument an Wolfgang Wahlster verschicken.
(I would like to send a document to Wolfgang Wahlster.)

Fig. 8. Document scanning area and address information

Sending a document requires the address of the recipient. SMARTKOM therefore must combine two different applications to satisfy a single user request. It looks up its address database for the coordinates of Wolfgang Wahlster. Being successful, Smartakus presents a scanning area and the address information (see Fig. 8, on the left), saying:

(17) SYS: Ich habe den Eintrag von Wolfgang Wahlster gefunden. Bitte legen Sie das Dokument auf die dafür vorgesehene Fläche.
(I found the address for Wolfgang Wahlster. Please put the document on the scanning area.)

Following the request of the system, the user places the document on the area. The document camera detects the object and scans it. At the end of the scan process, Smartakus tells the user:

(18) SYS: Bitte nehmen Sie es jetzt wieder weg.
(Please remove it now.)

The system displays the scanned document in the scanning area (see Fig. 8, on the left) and finally sends it via electronic mail.

(19) SYS: Das Dokument wurde aufgenommen. Das Dokument wurde abgeschickt.
(The document has been scanned. The document has been sent.)

The user is satisfied with SMARTKOM and leaves the system to meet her friend, saying:

(20) USR: Tschau.
(Good bye.)

(21) SYS: Auf Wiedersehen.
(See you again.)

4 Conclusion

In this contribution, we gave an impression of how the multimodal dialogue capabilities of SMARTKOM are defined, starting from dialogue description of scenario experts, resulting in detailed documentation of dialogues and their individual turns. The recorded example dialogue presented here provides an idea of one particular interaction sequence, which combines seven different applications—cinema program, city information, seat reservation, pedestrian navigation, document scanning, address book and email—in a coherent, seamless interaction. The subsequent chapters of this volume will provide a thorough discussion of the underlying methods and techniques that are required to achieve this kind of advanced multimodal dialogue functionality.

References

A. Berton, D. Bühler, and W. Minker. SmartKom-Mobile Car: User Interaction With Mobile Services in a Car Environment, 2006. In this volume.

J.M. Carroll. *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press, Cambridge, MA, 2000.

J.R. Gardner and Z.L. Rendon. *XSLT and XPATH: A Guide to XML Transformations*. Prentice Hall, 2001.

D. Gelbart, J. Bryant, A. Stolcke, R. Porzel, M. Baudis, and N. Morgan. SmartKom English: From Robust Recognition to Felicitous Interaction, 2006. In this volume.

S. Goronzy, S. Rapp, and M. Emele. The Dynamic Lexicon, 2006. In this volume.

G. Herzog and A. Ndiaye. Building Multimodal Dialogue Applications: System Integration in SmartKom, 2006. In this volume.

G. Herzog, A. Ndiaye, S. Merten, H. Kirchmann, T. Becker, and P. Poller. Large-Scale Software Integration for Spoken Language and Multimodal Dialog Systems. *Natural Language Engineering*, 10(3–4):283–305, 2004. Special Issue on Software Architecture for Language Engineering.

G. Herzog and N. Reithinger. The SmartKom Architecture: A Framework for Multimodal Dialogue Systems, 2006. In this volume.

A. Horndasch, H. Rapp, and H. Röttger. SmartKom-Public, 2006. In this volume.

R. Malaka, J. Häußler, H. Aras, M. Merdes, D. Pfisterer, M. Jöst, and R. Porzel. SmartKom-Mobile: Intelligent Interaction With a Mobile System, 2006. In this volume.

T. Portele, S. Goronzy, M. Emele, A. Kellner, S. Torge, and J. te Vrugt. SmartKom-Home: The Interface to Home Entertainment, 2006. In this volume.

M.B. Rosson and J.M. Carroll. Scenario-Based Design. In: J.A. Jacko and A. Sears (eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pp. 1032–1050, Mahwah, NJ, 2003. Lawrence Erlbaum.

F. Schiel and U. Türk. Wizard-of-Oz Recordings, 2006. In this volume.

Part II

Multimodal Input Analysis

The SmartKom Architecture: A Framework for Multimodal Dialogue Systems

Gerd Herzog and Norbert Reithinger

DFKI GmbH, Saarbrücken, Germany
{herzog,reithinger}@dfki.de

Summary. SMARTKOM provides an adaptive and reusable dialogue shell for multimodal interaction that has been employed successfully to realize fully-fledged prototype systems for various application scenarios. Taking the perspective of system architects, we review the overall design and specific architecture framework being applied within SMARTKOM. The basic design principles underlying our approach are described, and the different instantiations of the conceptual architecture are presented to illustrate the adaptibility and flexibility of the generic framework.

1 Introduction

The next generation of user interfaces aims at intelligent systems that are able to adapt to common forms of human dialogues and hence provide more intuitive and natural ways of interaction (Bernsen and Dybkjær, 1998; Blattner and Dannenberg, 1992; Maybury, 1993; Maybury and Wahlster, 1998b; Oviatt, 2003; Sullivan and Tyler, 1991). This ambitious goal, however, poses new challenges for the design and implemention of interactive software systems. Compared with traditional user interface software, these kinds of intelligent interfaces, which are capable of employing natural language as well as other modalities for communication, require more sophisticated architecture models.

The development of a complex multimodal system like SMARTKOM, which realizes three different application scenarios, requires an architecture that accomodates various, sometimes conflicting goals:

- Natural interaction behaviour: Facilitate timely, user adapted interaction.
- Flexibility: Handle a large number of modalities and applications.
- Openness: Allow for different processing approaches.
- Manageability: Support distributed development.

In this chapter, we elucidate the architectural design of SMARTKOM, which provides a practical framework for the successful realization of effective multimodal dialogue applications. The following section discusses essential requirements for multimodal dialogue systems and presents a more elaborate high-level architecture for

intelligent interfaces. Based on these design foundations, a detailed description of the specific architecture framework that has been developed within the SMARTKOM project will be provided in Sect. 3. Following the introduction of the underlying framework, the different instantiations of the generic SMARTKOM architecture are presented in Sect. 4. There we highlight the flexibility and adaptibility of our approach.

2 Architectural Requirements for a Multimodal Interaction System

Conceptual architectures like the ones described in Allen et al. (2000), Bordegoni et al. (1997), Elting et al. (2003), Feiner (1991), Hill et al. (1992) and Maybury and Wahlster (1998a) contrast the system design of traditional handcrafted interfaces with intelligent user interfaces. Within the classical Seeheim reference model (Pfaff, 1985), the internal structure of the user interface component that mediates between the end user and the underlying application system is partitioned into three distinct conceptual tiers:

- The *presentation* level, or display management, determines the appearance and low-level behaviour of the user interface.
- *Dialogue control* constitutes the intermediary processing stage which, manages the flow and order of interaction.
- The *application interface* as the third logical component provides the semantic model of the application, which is needed to access the functional core of the system.

This coarse modularization, however, is not sufficiently detailed to capture the basic structure of multimodal interfaces and to serve as a starting point for the development of the modular structure of a complex multimodal system.

Figure 1 provides a sketch of a more refined architecture at a high level of granularity. The conceptual architecture is based on the functional blocks of a generic multimodal dialogue system and decomposes the processing task into the following parts:

- Sensor-specific input processing: This layer of functionalities comprises all technical input devices like microphones and gesture recognition technologies. It provides a standardized, hardware-independent signal- or symbol-level interface for further processing.
- Modality-specific analysis: This processing level consists of the recognizers that transform sensor signals from the environment into symbolic information and analyzers that provide a meaning description for these signals.
- Modality fusion: This stage merges the meanings recorded in different modalities to one coherent, unified meaning representation. A core aspect of modality fusion is the mutual disambiguation of multimodal input on semantic and pragmatic levels.

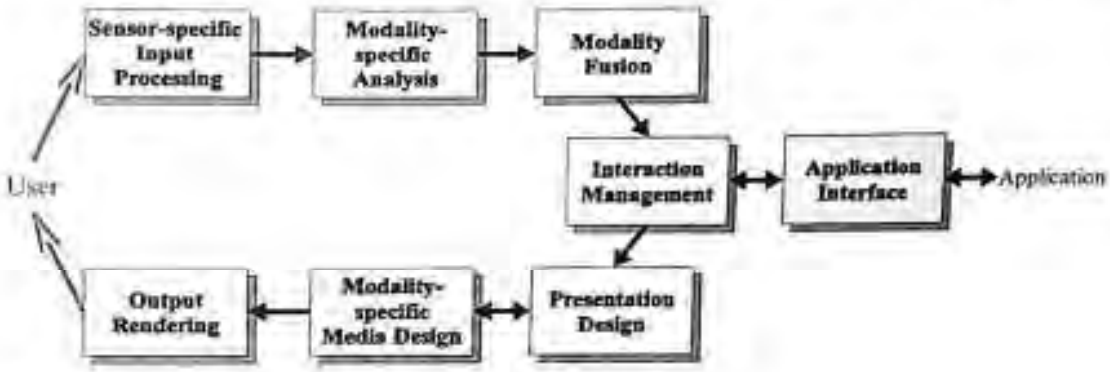

Fig. 1. A high-level architecture for multimodal interaction

- Interaction management: Here resides the "mind" of the machine. This part of the system identifies the intention of the user, determines the next steps to be taken by the system, and addresses the various application functions.
- Application interface: This layer defines an abstract interface for the interaction with the background applications. It hides application-specific technical details and thus provides a generic way to control the applications.
- Presentation design: Given a system intention, the information to be conveyed needs to be transformed into coordinated multimodal output. This functional block constitutes the first stage of modality fission and deals with the overall organization of the intended multimodal presentation. It includes in particular subtasks like content selection, media and modality allocation, layout design, and coordination.
- Modality-specific media design: In close cooperation with the presentation design stage, the processing on this layer is specifically concerned with the conversion of abstract content structures into presentable media objects.
- Output rendering: The final function block comprises the technical means to present the coordinated system reaction in the defined media channels like screens, loudspeakers, or force-feedback devices.

Of course, this breakdown of the architecture cannot and does not define sharp lines between the functional blocks. For example, one could argue about the output interface of sensor-specific input processing. For speech, it could either be a standardized audio stream after echo compensation or the output of the speech recognizer, if that component already incorporates all lower-level signal processing functionalities. In practice, the exact boundaries between the processing steps may depend on the constraints imposed by available components that are to be used for building a multimodal system.

In addition to the processing steps mentioned above, a rich variety of knowledge sources that are not shown in the diagram are also vital within the system and may require shared access from multiple components. Important knowledge sources for intelligent multimodal interaction include explicit models of the user, domain, con-

text, task, and discourse as well as the different media and modalities. Consider, for example, reference resolution processes, which need access to data from discourse context storage. This information has to be provided by the output components and is delivered on demand to all interested parties.

The architecture sketch highlights the major flow of information through the system, leading to a linear structure that resembles a processing pipeline. However, this depicts only the main and principal succession of processing steps, including analysis, execution, and generation. It does not rule out additional interconnections. The individual components of the system could be regarded as communicating processes rather than just pipeline stages.

If the system designer now refines this high-level architecture, various additional aspects have to be considered (Bunt et al., 2001), including

- Declarative data models: The benefits of a flexible and configurable knowledge-based approach can only be exploited if system design strictly follows the principle that there should be no processing and presentation without explicit representation.
- Time-coordinated parallelism: Modality-specific analyzers and generators run in parallel. Therefore the analysis and generation of events must be coordinated, and the architecture must support this behaviour. Especially important are time constraints on processing as well as the provision of timing information within data formats.
- Standardised interfaces supporting the handling of probabilistic information: Modality recognizers usually provide alternative interpretation hypotheses augmented with information about the recognition quality. The design of the integration platform has to support the maintenance and handling of multiple, competing hypotheses.
- Incremental processing: It cannot be assumed that only complete representations are passed between the function blocks in the system. Increments of information, either originating in the user's input style or caused by processing methods, must be handled properly in the system.
- Iterative development: A complex multimodal system usually starts from a limited core of modalities and functionalities. Step-by-step, the coverage of the system is extended, with new input/output modalities being added. Therefore, the design must be extensible, not limiting additions by hardwired interaction and control paths. This requires accessible interfaces and extensive support for debugging and monitoring.
- Distributed development: Depending on specific organizational and technical constraints, software modules from various sites and with different backgrounds have to be integrated. For instance, it is very typical that off-the-shelf recognizers are to be used together with bespoke analysis components.

In the next sections, we will show how the SMARTKOM system realizes these requirements in three different application scenarios.

3 SmartKom as an Adaptive and Reusable Shell for Multimodal Dialogues

SMARTKOM represents a new generation of multimodal dialogue systems, dealing not only with basic modality integration and synchronization but addressing the full spectrum of dialogue phenomena arising in multimodal interaction (Reithinger et al., 2003b; Wahlster, 2002). With a clear focus on reusability, SMARTKOM has been designed as a transmutable system that can engage in many different types of tasks in various usage scenarios (Reithinger et al., 2003a; Wahlster, 2003).

3.1 Distinguishing Features of the SmartKom Approach

The SMARTKOM system relies on the so-called *situated, delegation-oriented dialogue paradigm* (Wahlster et al., 2001) to provide an anthropomorphic and affective user interface. This interaction metaphor is based upon the idea that the user delegates a task to the virtual communication assistant, which is visualized as a life-like character. The interface agent recognises the user's intentions and goals, asks the user for feedback if collaboration is necessary, accesses the various application services on behalf of the user, and presents the collated results in an appropriate form. The interaction style being employed within SMARTKOM targets in particular nondesktop scenarios, such as smart rooms, kiosks, or mobile application contexts.

An interesting property of the SMARTKOM dialogue shell is its ability to integrate multiple services, i.e., different application functions that have been developed independently from each other, into a coherent, value-added system. Another peculiarity of the system is a build-in self-monitoring capability which continuously determines the overall processing state using status information from the individual components. In the sense of a reflective architecture (Maes and Nardi, 1988), this kind of metalevel reasoning and control can be exploited to provide the user with appropriate feedback concerning system activity and potential error situations.

3.2 Anatomy of the SmartKom Core System

The SMARTKOM kernel shown in Fig. 2 provides the core building blocks for a multimodal interactive system. The diagram documents the software architecture using a component-and-connector viewtype as defined in Clements et al. (2003). Each component represents one of the principal processing units of the executing system, and the highlighted components are being reused in all application cases. Since SMARTKOM has been implemented as a distributed system (see Herzog et al. (2004) for details), every component corresponds to an independent process. The connectors illustrate the pathways of interaction. The underlying architectural style is based on communicating processes, more specifically on the publish–subscribe style. In this style, the interaction is event-driven using directed communication links between a message sender, who acts as data producer or event source, and a set of recipients, the data consumers.

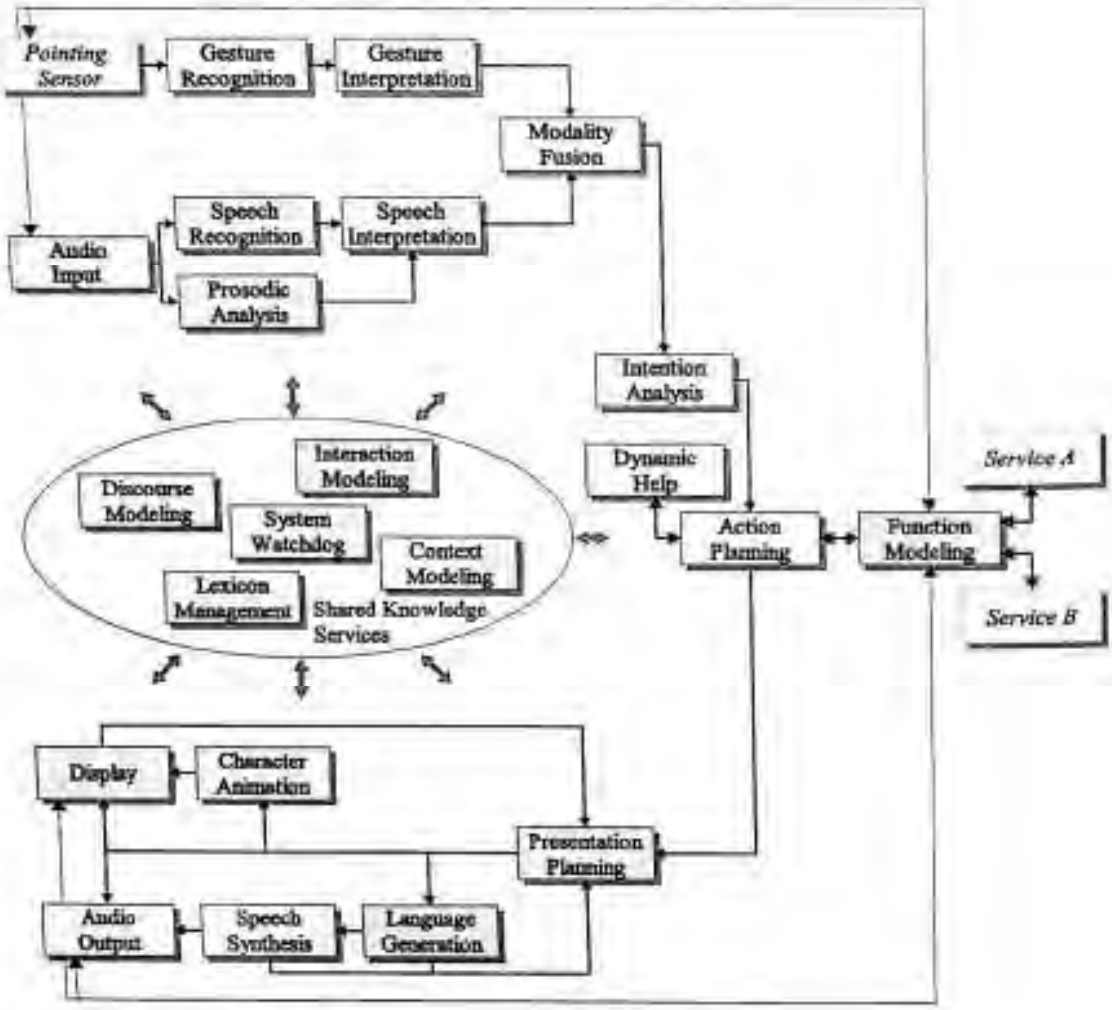

Fig. 2. SMARTKOM architecture blueprint

To provide a clearer outline of the system structure, the connectors chosen for Fig. 2 denote major information flows and may combine several publish–subscribe communication links. The various publish–subscribe connections between the shared knowledge sources and the other components of the dialogue backbone have only been sketched.

A basic SMARTKOM system supports multimodal input using speech plus gestures and integrates some application-specific services. The specific sensor type for gesture input depends on the technical setup of the given application scenario. Practical options that have been used for the SMARTKOM prototypes include either pen-based pointing or a vision system for hand tracking.

The separation of sensor-specific input processing into distinct components, which connect some technical device to the system and which encapsulate access to the underlying hardware, constitutes a first decomposition principle of the architecture framework. In order to provide maximum flexibility for system deployment,

the same approach is followed with respect to the output devices that are needed for the rendering stage.

Modality-specific input analysis can be modularized into separate components, and it has proven useful to differentiate further between modality-specific recognition, i.e., processing of sensory data on the lexical and syntactical layer, and subsequent semantic interpretation of the derived symbolic representation.

Interaction management within SMARTKOM comprises several components. The main task of intention analysis is to select the most plausible input interpretation from the given set of hypotheses. Action planning constitutes the heart of dialogue control and is backed by a supplementary help component that is activated whenever difficulties occur during the interaction or if additional help is needed. The main processing stages are supported by different components that actively maintain shared knowledge sources:

- The multimodal discourse model is utilized for the semantic and pragmatic interpretation during input and output processing. It is dynamically updated as system output progresses and performs contextual reasoning and scoring.
- Contextual information, as it is needed to handle references to situative parameters like current place and time, is provided by the context model.
- Interaction modeling is concerned with different aspects like available modalities and user preferences for specific forms of communication as well as the affective state of the user. The interaction model allows one to dynamically adapt the communicative behaviour of the system.
- The lexicon is a dynamic knowledge source, which is updated with additional lexical entries depending on dynamic application data as it is received from the external information services. Lexicon updates are propagated to all components that process natural language input and output.
- The system watchdog monitors the processing status of all individual components to offer up-to-date information concerning the system state. This is used to provide immediate feedback or to initiate helpful reactions in case of processing problems.

The function modeling component realizes the application interface and in addition controls all input and output devices to coordinate access depending on an explicit state model. The application layer itself is modularized into multiple service components, each of which realizes some application-specific functionality.

The remaining elements of the component ensemble deal with modality fission. A second planning component is responsible for the presentation design. Part of the modality-specific output design is carried out by the character animation component, which realizes the perceivable behaviour of the embodied conversational agent.

In addition to the integration of application-specific service components, the architecture framework provides other extension points to expand the core dialogue components with even more modalities. A practical example for an optional input modality is the interpretation of facial expressions to obtain information about the affective state of the user. The SMARTKOM prototypes focus mainly on audiovisual sensor input and presentations. With respect to other human senses, it would be

feasible to include haptic sensations as well, e.g., using a force-feedback device. A limited form of tactile interaction is already provided when using pen-based input.

3.3 Representing Data with Multimodal Markup Language

The interaction between the distributed components within the SMARTKOM architecture is based on the exchange of structured data through messages. The external data format for the representation of structured information employs XML notation. XML, the *eXtensible Markup Language*, provides a flexible standard for the definition of specific data exchange formats that ensures interoperability between heterogeneous software components.

In the context of the SMARTKOM project we have developed M3L (Multimodal Markup Language) to cover all data interfaces within the entire multimodal dialogue system in a declarative way. Instead of using several unrelated XML languages for the various information flows, we aimed at an integrated and coherent language specification, which includes all substructures that may occur on the different communication links.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<intentionLattice>
 <latticeStart>2003-01-24T11:01:55.000</latticeStart>
 <latticeDuration>PT3.02S</latticeDuration>
 <hypothesisSequences>
  <hypothesisSequence>
   <score>
    <source>acoustic</source><value>0.82</value>
   </score>
   <score>
    <source>gesture</source><value>0.73</value>
   </score>
   <score>
    <source>understanding</source><value>0.89</value>
   </score>
   <hypothesis>
    [...]
    <discourseStatus>
     <discourseAction> set </discourseAction>
     <discourseTopic>
      <goal> epg-info </goal>
    [...]
    <event>
     <informationSearch id="dim769">
      <pieceOfInformation>
       <broadcast id="dim763">
        <avMedium>
         <avMedium id="dim767">
          <avType> featureFilm </avType>
          <title> Casablanca  </title>
 [...]
 </hypothesisSequences>
</intentionLattice>
```

Time stamp

Confidence in speech recognition result

Confidence in gesture recognition result

Confidence in speech understanding result

Planning act

Object reference

Fig. 3. Excerpt from an M3l structure

The basic information flow from user input to system output continuously adds further processing results so that the representational structure will be refined step-

by-step. A typical example of such a complex M3L expression is shown in Fig. 3. The partial listing contains a so-called intention lattice that represents the interpretation result for a multimodal user input which combines a verbal information request ("Tell me more about this film") with a pointing gesture.

Conceptual taxonomies provide the foundation for the representation of domain knowledge as it is required within a dialogue system to enable a natural conversation in the given application scenario. In order to exchange instantiated knowledge structures between different system components they need to be encoded in M3L. The corresponding parts of the M3L definition are derived from an underlying ontology that captures the required terminological knowledge (Gurevych et al., 2003). For example, in Fig. 3 the representation of the event structure inside the intention lattice originates from the ontology. M3L can easily be updated as the ontology evolves while novel application domains are being modeled.

4 Instantiations of the Generic Architecture

The SMARTKOM architecture framework is designed to support a wide range of collaborative and multimodal dialogues that allow users to intuitively and efficiently access the functionalities needed for their task. The SMARTKOM prototypes address three different application scenarios that exhibit varying characteristics that influence the architectural design.

4.1 Public Information and Communication Kiosk

SMARTKOM-Public realizes an advanced multimodal information and communication kiosk for train stations, airports, or other public places. It supports users seeking information concerning movie programs, offers reservation facilities, and provides personalized communication services using telephone, fax, or electronic mail.

Figure 4 provides an overview concerning the specific software architecture of SMARTKOM-Public. Scenario-specific components are highlighted using gray boxes.

Gesture input is based on an infrared camera system that tracks the user's hand as it moves across a video projection of the graphical output. This kind of vision-based sensor does not require direct physical contact with the horizontal projection surface and hence allows for more natural hand gestures.

SMARTKOM-Public integrates facial expressions as an additional input modality. In combination with acoustic indicators derived from prosodic analysis, the interpretation of facial expressions is used to recognize the current affective state of the user.

The kiosk system is equipped with two application-specific input components for pen-based signature input and an additional camera system that takes digital images of objects placed on the projection panel. In the given application scenario, the following functions are made available through the main service components:

- Biometric authentication using either voice, signature, or hand contour

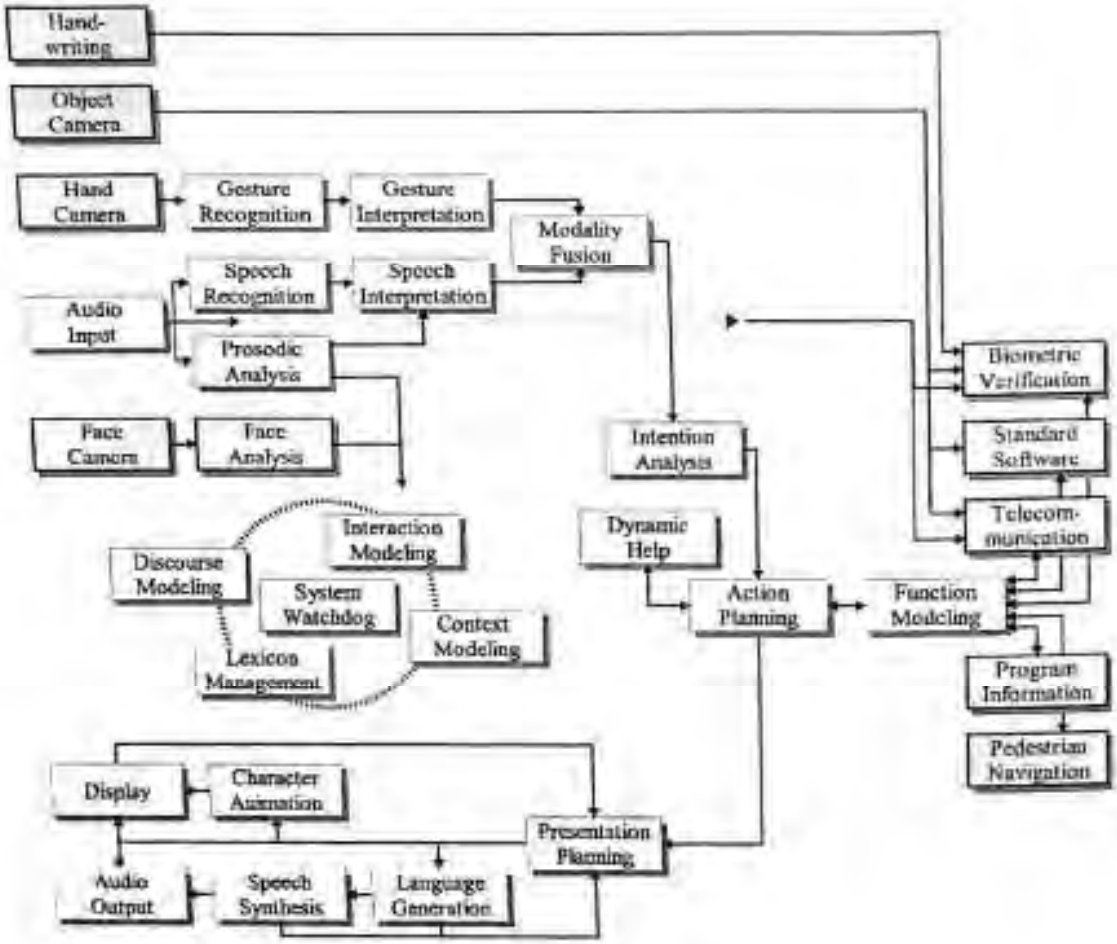

Fig. 4. Component architecture of SMARTKOM-Public

- Access to standard applications like address book and e-mail transfer
- Telephone connections and fax transmission
- Cinema program and reservation service
- Provision of maps with cinema locations and marked routes to the desired destination

The function model of the application interface coordinates the interplay of services to carry out specific tasks. The need for flexible application-specific use of input and output components is the reason why the basic control of all shared devices has been allocated to the function modeling component. The corresponding control lines have been omitted from Fig. 4.

Depending on the current application context, audio input can be rerouted to the biometrics component or to the telecommunication service. Input from the object camera is used to send documents (via e-mail or fax), and the input component can also record the hand contour for biometric verification.

4.2 Infotainment Assistant for the Living Room at Home

In its basic configuration, SMARTKOM-Home aims to provide a multimodal portal to home entertainment services. Using a portable tablet PC, the user is able to utilize the system as an electronic program guide or to easily control consumer electronics devices like a television set or a digital video casette recorder.

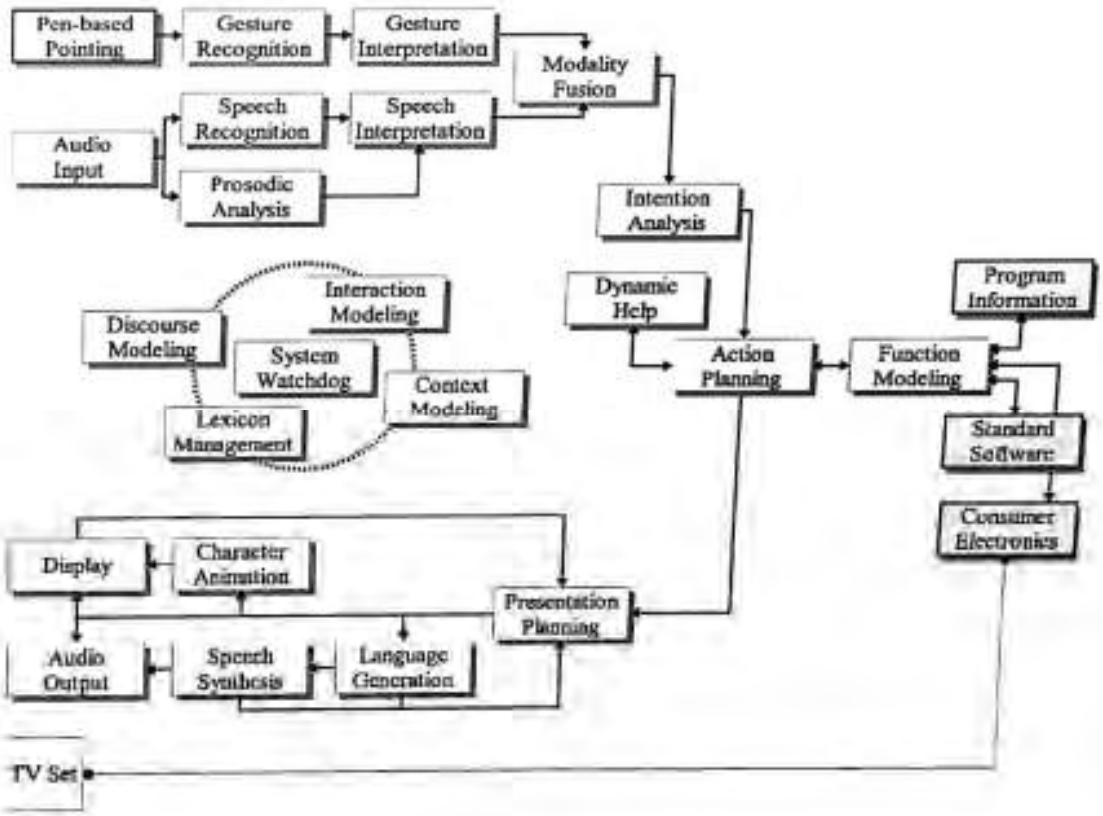

Fig. 5. Component architecture of SMARTKOM-Home

Figure 5 highlights the specific components of the SMARTKOM-Home prototype. The system supports the basic modalities that are associated with the core multimodal system and the user can employ a pen for stylus-drawn gestures on the tablet PC's screen. In the context of the Home scenario, however, two different styles of interaction modes are supported. In so-called *lean-forward* mode, coordinated speech and gesture input can be used for multimodal interaction with the system. Alternatively, the user is able to put away the tablet PC and change to so-called *lean-backward* mode. This interaction style is constrained to verbal communication. To compensate for the unavailable graphical view in this case, the presentation design needs to dynamically adapt its presentation strategies using more elaborate natural language output. The user can easily request the activation or deactivation of the display to switch between the two modes.

SMARTKOM-Home shares two of its service components with the Public scenario but employs different application functions from these services. The program database is primarly used to obtain detailed information concerning upcoming televison broadcasts and the calender management utility from the standard software component is utilized to schedule off-air recordings of selected programs. The third service component provides a high-level interface to various appliances. In particular, the consumer electronics service provides remote control of a TV set, a VCR, and a satellite receiver. The TV screen can be regarded as an application-specific output device, which is not directly available for system output that is routed through the interface components.

SMARTKOM offers a very flexible and open approach for application integration. Services constitute connector components that provide a well-defined link to some application-specific functionality and often encapsulate complete and complex application-specific subsystems. A typical case is the integration of commercial off-the-shelf software. The standard software component in SMARTKOM, for example, constitutes a bridge to external standard applications provided by a separate installation of IBM Lotus software. The service component for consumer electronics reuses already available software modules from the related Embassi project (Elting et al., 2003). It is basically a wrapper for a complex multiagent system, which in fact relies on a different kind of middleware solution than the publish–subscribe style used to interconnect the SMARTKOM components.

4.3 Mobile Travel Companion

SMARTKOM-Mobile provides a travel companion that can be used inside the car and while walking around. This application scenario comprises typical services like integrated trip planning and incremental route guidance through a city via GPS. Outside the car, the user operates the system using a handheld computer as a front end. Inside the instrumented car, the onboard equipment of the vehicle is used for displaying visual information and for audio input/output. A pecularity of the mobile travel companion is the seamless transition from one device to the other without losing the current dialogue context.

Figure 6 shows the basic components of SMARTKOM Mobile. Multimodal interaction is possible using pen-based input on the handheld device. The in-car display, however, is not touch-sensitive.

The handover between alternative devices constitutes one of the central technical challenges imposed by this specific application scenario. The component named *mobile device* provides a generic solution to manage different remote front-end devices. In order to avoid time-consuming component restarts and shutdowns during system runtime, audio input and output as well as the display component have been enhanced with streaming capabilities for input and output redirection. RTP, the Internet-standard protocol for the transport of real-time data, is used for bidirectional audio connections and VNC (Richardson et al., 1998) enables remote access to the display component. The mobile device component serves as a special kind of input device.

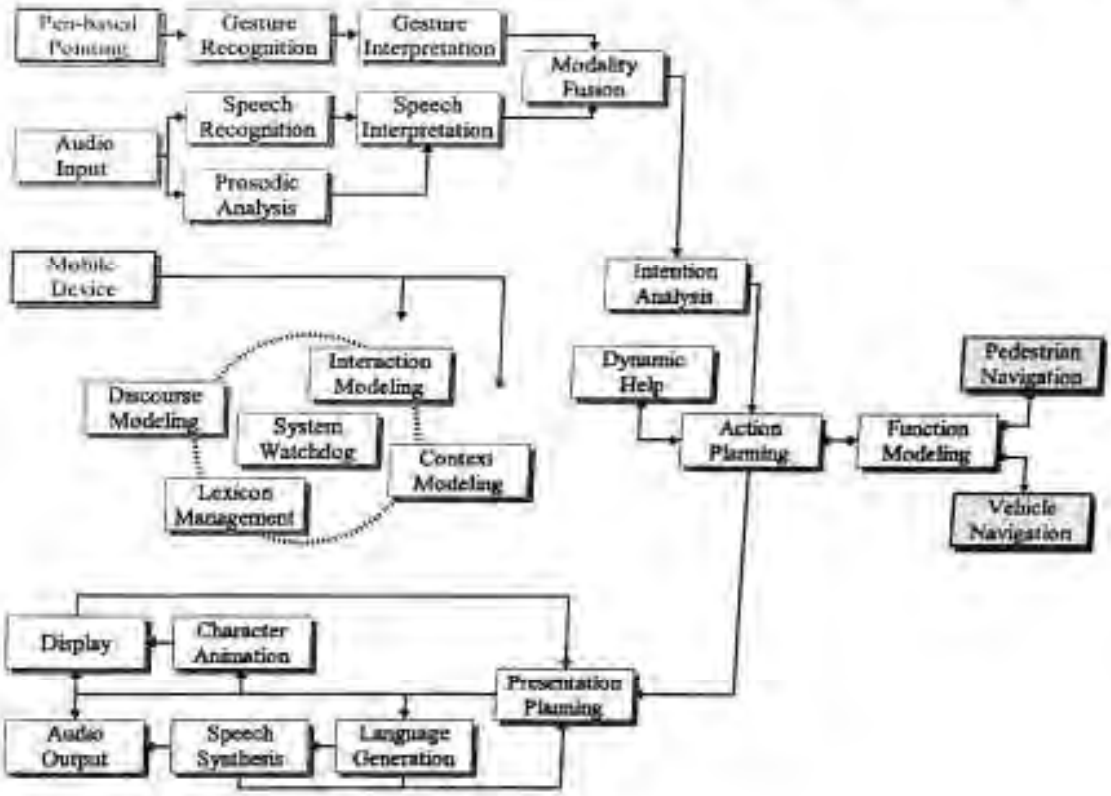

Fig. 6. Component architecture of SMARTKOM-Mobile

After pressing the push-to-activate button on the handheld or in the car, the characteristics of the new front-end device—also including address parameters for video and audio redirection—are propagated by the mobile device component to dynamically update the interaction model. The significant difference in screen format of the handheld computer and the in-car display enforces the presentation design to automatically adapt the visual output accordingly.

The vehicle navigation component includes commercial off-the-shelf navigation software to generate map displays with marked vehicle routes. It also provides information concerning parking locations. The pedestrian navigation service offers feature-rich functions for map generation, touristic information, pretrip planning, and incremental route guidance in Heidelberg. The component encapsulates a complex subsystem that is realized in a modular fashion, using a multiagent framework.

SMARTKOM-Mobile introduces an important new aspect for the dialogue management. In addition to user inputs, the system also needs to handle external events since navigation instructions are generated on the fly by the navigation service to provide incremental route guidance.

4.4 Other System Configurations

The SMARTKOM prototypes for the three sample scenarios use German for speech input and output. A special English system is based on a subset of SMARTKOM-Mobile, excluding incremental route guidance. SMARTKOM-English uses adapted versions of language-specific recognition, analysis, generation, and synthesis components.

In addition to the basic prototypes, many other plug-and-play variations of system setups with extended or restricted capabilities can be defined easily, using a flexible configuration mechanism for system instantiations. For example, for demonstration purposes, the SMARTKOM reference installation is often used as an extended Public system that integrates all available service components.

5 Conclusion

Multimodal dialogue systems hold the promise of providing more intuitive and natural ways of human–computer interaction. In order to construct such advanced user system interfaces, an effective architectural design is of utmost importance. SMARTKOM provides an adaptive and reusable dialogue shell that has been employed successfully to realize fully-fledged prototype systems for various application scenarios.

SMARTKOM is based on an open architecture for multimodal dialogue systems that is flexible and adaptive. All available components can be reused directly, and modular organization into different service components simplifies the integration of application-specific functions. With respect to the dialogue backbone, the realization of new application scenarios primarily becomes a knowledge engineering task instead of an extensive programming exercise. In order to extend the domain-specific dialogue behaviour, mainly the component-specific declarative knowledge sources have to be adapted. The generic SMARTKOM architecture with its core components and their interconnections promises to provide a solid base structure, even in cases where a customized multimodal dialogue system may require more advanced adaptations and extensions.

References

J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. An Architecture for a Generic Dialogue Shell. *Natural Language Engineering*, 6(3):1–16, 2000.

N.O. Bernsen and L. Dybkjær. Is Speech the Right Thing for Your Application? In: *Proc. ICSLP-98*, pp. 3209–3212, Sydney, Australia, 1998.

M.M. Blattner and R.B. Dannenberg (eds.). *Multimedia Interface Design*. ACM, New York, 1992.

M. Bordegoni, G. Faconti, S. Feiner, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. A Standard Reference Model for Intelligent Multimedia Presentation Systems. *Computer Standards & Interfaces*, 18(6–7):477–496, 1997.

H. Bunt, M.T. Maybury, and W. Wahlster. Coordination and Fusion in Multimodal Interaction. Dagstuhl Seminar Report 325, IBFI, Wadern, Germany, 2001.

P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, Boston, MA, 2003.

C. Elting, S. Rapp, G. Möhler, and M. Strube. Architecture and Implementation of Multimodal Plug and Play. In: *Proc. 5th Int. Conf. on Multimodal Interfaces*, pp. 93–100, Vancouver, Canada, 2003. ACM.

S. Feiner. An Architecture for Knowledge-Based Graphical Interfaces. In: J.W. Sullivan and S.W. Tyler (eds.), *Intelligent User Interfaces*, pp. 259–279, New York, 1991. ACM.

I. Gurevych, S. Merten, and R. Porzel. Automatic Creation of Interface Specifications from Ontologies. In: H. Cunningham and J. Patrick (eds.), *Proc. HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pp. 59–66, Edmonton, Canada, 2003. Association for Computational Linguistics.

G. Herzog, A. Ndiaye, S. Merten, H. Kirchmann, T. Becker, and P. Poller. Large-Scale Software Integration for Spoken Language and Multimodal Dialog Systems. *Natural Language Engineering*, 10(3–4):283–305, 2004. Special Issue on Software Architecture for Language Engineering.

W. Hill, D. Wroblewski, T. McCandless, and R. Cohen. Architectural Qualities and Principles for Multimodal and Multimedia Interfaces. In: Blattner and Dannenberg (1992), pp. 311–318.

P. Maes and D. Nardi (eds.). *Meta-Level Architecture and Reflection*. North-Holland Amsterdam, The Netherlands, 1988.

M.T. Maybury (ed.). *Intelligent Multi-Media Interfaces*. AAAI/MIT Press, Menlo Park, CA, 1993.

M.T. Maybury and W. Wahlster. Intelligent User Interfaces: An Introduction. In: M.T. Maybury and W. Wahlster (eds.), *Readings in Intelligent User Interfaces*, pp. 1–13, San Francisco, CA, 1998a. Morgan Kaufmann.

M.T. Maybury and W. Wahlster (eds.). *Readings in Intelligent User Interfaces*. Morgan Kaufmann, San Francisco, CA, 1998b.

S. Oviatt. Multimodal Interfaces. In: J.A. Jacko and A. Sears (eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pp. 286–304, Mahwah, NJ, 2003. Lawrence Erlbaum.

G.E. Pfaff (ed.). *User Interface Management Systems: Proc. the Seeheim Workshop*. Springer, Berlin Heidelberg New York, 1985.

N. Reithinger, J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pfleger, P. Poller, M. Streit, and V. Tschernomas. SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications. In: *Proc. 5th Int. Conf. on Multimodal Interfaces*, pp. 101–108, Vancouver, Canada, 2003a. ACM.

N. Reithinger, G. Herzog, and A. Ndiaye. Situated Multimodal Interaction in SmartKom. *Computers & Graphics*, 27(6):899–903, 2003b.

T. Richardson, Q. Stafford-Fraser, K.R. Wood, and A. Hopper. Virtual Network Computing. *IEEE Internet Computing*, 2(1):33–38, 1998.

J.W. Sullivan and S.W. Tyler (eds.). *Intelligent User Interfaces*. ACM, New York, 1991.

W. Wahlster. SmartKom: Fusion and Fission of Speech, Gestures, and Facial Expressions. In: *Proc. 1st Int. Workshop on Man-Machine Symbiotic Systems*, pp. 213–225, Kyoto, Japan, 2002.

W. Wahlster. SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In: R. Krahl and D. Günther (eds.). *Proc. Human Computer Interaction Status Conference 2003*, pp. 47–62, Berlin, Germany, June 2003. DLR.

W. Wahlster, N. Reithinger, and A. Blocher. SmartKom: Multimodal Communication with a Life-like Character. In: *Proc. EUROSPEECH-01*, vol. 3, pp. 1547–1550, Aalborg, Denmark, September 2001.

Modeling Domain Knowledge: Know-How and Know-What

Iryna Gurevych*, Robert Porzel, and Rainer Malaka

European Media Laboratory GmbH, Heidelberg, Germany
{robert.porzel,rainer.malaka}@eml-d.villa-bosch.de
iryna.gurevych@eml-r.villa-bosch.de
* Current affiliation EML Research gGmbH

Summary. The approach to knowledge representation taken in the multimodal, multidomain, and multiscenario dialogue system — SMARTKOM — is presented. We focus on the ontological and representational issues and choices helping to construct an ontology, which is shared by multiple components of the system and can be reused in different projects and applied to various tasks. Finally, two applications of the ontology that highlight the usefulness of our approach are described.

1 Introduction

The ways in which knowledge has been represented in multimodal dialogue systems (MMDSs) show that individual representations with different semantics and heterogeneously structured content can be found in various formats within single natural language processing (NLP) systems and applications. For example, a typical NLP system, such as TRAINS (Allen et al., 1996) employs different knowledge representations for parsing, action planning, and generation, despite the fact that what is being represented is common to all those representations, e.g., the parser representation for *going from A to B* has no similarity to the action planner's representation thereof (Ferguson et al., 1996). Also central concepts, for example, *city*, are represented in multiple ways throughout the system.

The origin for this state of affairs is that the respective knowledge stores are handcrafted individually for each task. Sometimes they are compiled into code and cease to be externally available. Where an explicit knowledge representation is used, we find a multitude of formats and inference engines, which often cause both performance and tractability problems. In this paper we introduce the results of an effort to employ a single knowledge representation, i.e., an *ontology*, throughout a complete multimodal dialogue system. Therefore, we will describe the underlying modeling principles and the benefits of such a rigorously crafted knowledge store for the actual and future MMDS.

The SMARTKOM system comprises a large set of input and output modalities which the most advanced current systems feature, together with an efficient fusion

and fission pipeline. SMARTKOM features speech input with prosodic analysis, gesture input via infrared camera and recognition of facial expressions and their emotional states. On the output side, the system features a gesturing and speaking life-like character together with displayed generated text and multimedia graphical output. It currently comprises a set of modules running on a parallel virtual machine-based integration software called *Multiplatform* (Herzog et al., 2003).[1]

As mentioned in the introduction, complex MMDSs such as SMARTKOM can benefit from a homogeneous world model. This model serves as a common knowledge representation for various modules throughout the system. It represents a general conceptualization of the world (top-level or generic ontology) as well as of particular domains (domain-specific ontologies). This way, the ontology represents language-independent knowledge. The language-specific knowledge is stored elsewhere, e.g., in the lexicon containing lexical items together with their meanings defined in terms of ontology concepts.

The initial version of the ontology described herein was designed as a general-purpose component for knowledge-based NLP. It includes a top-level developed following the procedure outlined by Russell and Norvig (1995) and originally only covered the tourism domain encoding knowledge about sights, historical persons, and buildings. Then, the ontology was successfully extended and modified until it covered SMARTKOM domains, e.g., new media and program guides. The top-level ontology was reused with some slight extensions. Further developments were motivated by the need of a *process hierarchy*. This hierarchy models processes which are domain-independent in the sense that they can be relevant for many domains, e.g., *InformationSearchProcess* (see Sect. 3.3 in this chapter for more details).

Currently, the ontology employed by the system has about 730 concepts and 200 relations. The acquisition of the ontology went in two directions: top-down to create a top level of the ontology and bottom-up to satisfy the need of mapping lexical items to concepts. The purpose of the top level ontology is to provide a basic structure of the world, i.e., abstract classes to divide the universe in distinct parts as resulting from the ontological analysis (Guarino and Poli, 1995). The domain concepts emerged through a comprehensive corpus analysis. The most important modeling decisions will be discussed in Sect. 3. Once available, the ontology was augmented with comments containing definitions, assumptions and examples that facilitate its appropriate use in a multicomponent system such as SMARTKOM and its possible reuse in other systems. Such descriptions of ontology classes are particularly important as the meanings associated with them may vary considerably from one ontology to another.

In Sect. 2 we will introduce the representational formats pertinent to our ontology. In Sect. 3 we discuss the modeling principles underlying the ontology. Sect. 4 presents a few examples of the various ways in which the common ontology is employed throughout the system. Concluding remarks are given in Sect. 5.

[1] The abbreviation stands for "MUltiple Language/Target Integration PLATform FOR Modules".

2 The Representational Formalism Used

We give a brief outline of the formalism pertinent to the following description of the ontology. Various standardization efforts originating, e.g., in W3C and Semantic Web projects came up with a number of several knowledge modeling standards: Resource Description Framework (RDF), DARPA Agent Mark-up Language (DAML), Ontology Interchange Language (OIL) and Web Ontology Language (OWL). [2] The XML-based languages OIL/DAML+OIL are of particular interest for domain and discourse knowledge modeling. The advantages of the XML standard, availability of tools and their wide use let us choose the OIL-RDFS syntax. A detailed characterization of the formal properties of the OIL language has been given by Fensel et al. (2001).

We used the OIL language in particular as a whole range of software is freely available to support the ontology construction as mentioned above. In addition, the usage of the ontology in Semantic Web applications would be simplified. The FACT[3] system can be used as a reasoning engine for OIL ontologies, providing some automated reasoning capabilities, such as class consistency or subsumption checking. Graphical ontology engineering front ends and visualization tools are available for editing, maintaining, and visualizing the ontology. [4]

The semantics of OIL is based on description logic extended with concrete datatypes. The language employs a combination of frame and description logic. It provides most of the modeling primitives commonly used in the frame-based knowledge representation systems. Frames are used to represent concepts. These frames consist of a collection of classes along with a list of slots and attributes. Under the term *class* or *class expression*, a class name, or an enumeration, or a property-restriction, or a Boolean combination of class expressions is to be understood. Slots are interpreted as a collection of properties. They are divided into those that relate classes to other classes (so-called *object properties*) and those that relate classes to datatype values (so-called *datatype properties*). Slots can be filled by: class names, names of the atomic elements, collection of the above (conjunctive sets *and*, disjunctive sets *or*, or negation *not*) or concrete datatypes (*integers* and *strings*).

Then, domain and range restrictions of the slots can be defined. A domain restriction asserts that the property only applies to the instances of particular class expressions. A range restriction specifies that the property only assumes values that are instances of the respective class expressions. Slot fillers can have several types of further constraints, also called *facets*. These include *value-type* restrictions (all fillers must be of a particular class) and *has-value* restrictions (there must be at least one filler of a particular class). The *value-type* restriction corresponds to the universal quantifier of the predicate logic. The *has-value* restriction is analogous to the

[2] See http://www.w3c.org/RDF, http://www.ontoknowledge.org/oil, http://www.daml.org and http://www.w3.org/2001/sw/WebOnt/ for the individual specifications.

[3] See also http://www.cs.man.ac.uk/~horroks/FaCT/.

[4] See OilEd (http://oiled.man.ac.uk) for editing and FrodoRDFSViz (http://www.dfki.uni-kl.de/frodo/RDFSViz) for visualization.

existential quantifier. Another constraint on the slot fillers is *cardinality*, which limits the number of possible fillers of the given class. Atomic elements or individuals can also be associated with a class definition via slot constraints. The decision to restrict ourselves to schemes based on the description logic that allows us to represent enough knowledge for the effective operation of envisaged NLP applications, e.g., those described in Sect. 4.

3 Our Approach to Knowledge Representation

3.1 Top-Level Ontology

Following the distinctions made by Guarino and Welty (2000), we initially defined a collection of concepts that have *primary* ontological status. The guiding principle was to differentiate between the basic ontological entities and the roles taken by them in particular situations, events, or processes. For example, a building can be a hospital, a railway station, a school, etc. But while taking all these roles, it does not cease to be a building. Another example is a person who can take the role of a school teacher, a mother, etc., but it still remains a person for their entire life.

Here the question arises of how deep the differentiation should go. Consider the example of a person: We give a concept *Person* a primary ontological status, but what about the concepts *Man* and *Woman*? Should they be given the same status? Our answer is positive and is based, on one hand, on the assumption that sex is the primary property that defines a person as a man or a woman, and on the other hand, a functional approach shows that relations of these two classes to other classes and their other attributes can be determined by this property. In this way, the basic top-level ontological categorization in our system divides all concepts into two classes *Type* and *Role* (Fig. 1). As the class *Type* includes concepts with primary ontological status independent of the particular application, every system using the ontology for its specific purposes deals with the class *Role*.

Role is the most general class in the ontology representing concrete roles that any entity or process can perform in a specific domain. It is divided into *Event* and *AbstractEvent*. Along with concrete events, i.e., freestanding entities existing essentially in space or in time, our model includes abstract objects, e.g., numbers or abstract properties, such as spatial relations, and abstract processes or rather abstracted states every real process can go through, such as *Start*, *Done*, *Interrupt*, etc. These are modeled separately thereby allowing a uniform description of the processes throughout the ontology.

On the *Role* level we distinguish between *Event* and *AbstractEvent*. *Event* is used to describe a kind of role any entity or process may have in a real situation or process, e.g., a school or an information search. It is contrasted with *AbstractEvent*, which is abstracted from a set of situations and processes. It reflects no reality and is used for the general categorization and description, e.g., *Number*, *Set*, *SpatialRelation*. *AbstractEvent* has subclasses *AbstractObject* and *AbstractProcess*. *Events* are further classified in *PhysicalObject* and *Process*. In contrast to abstract objects, they have a

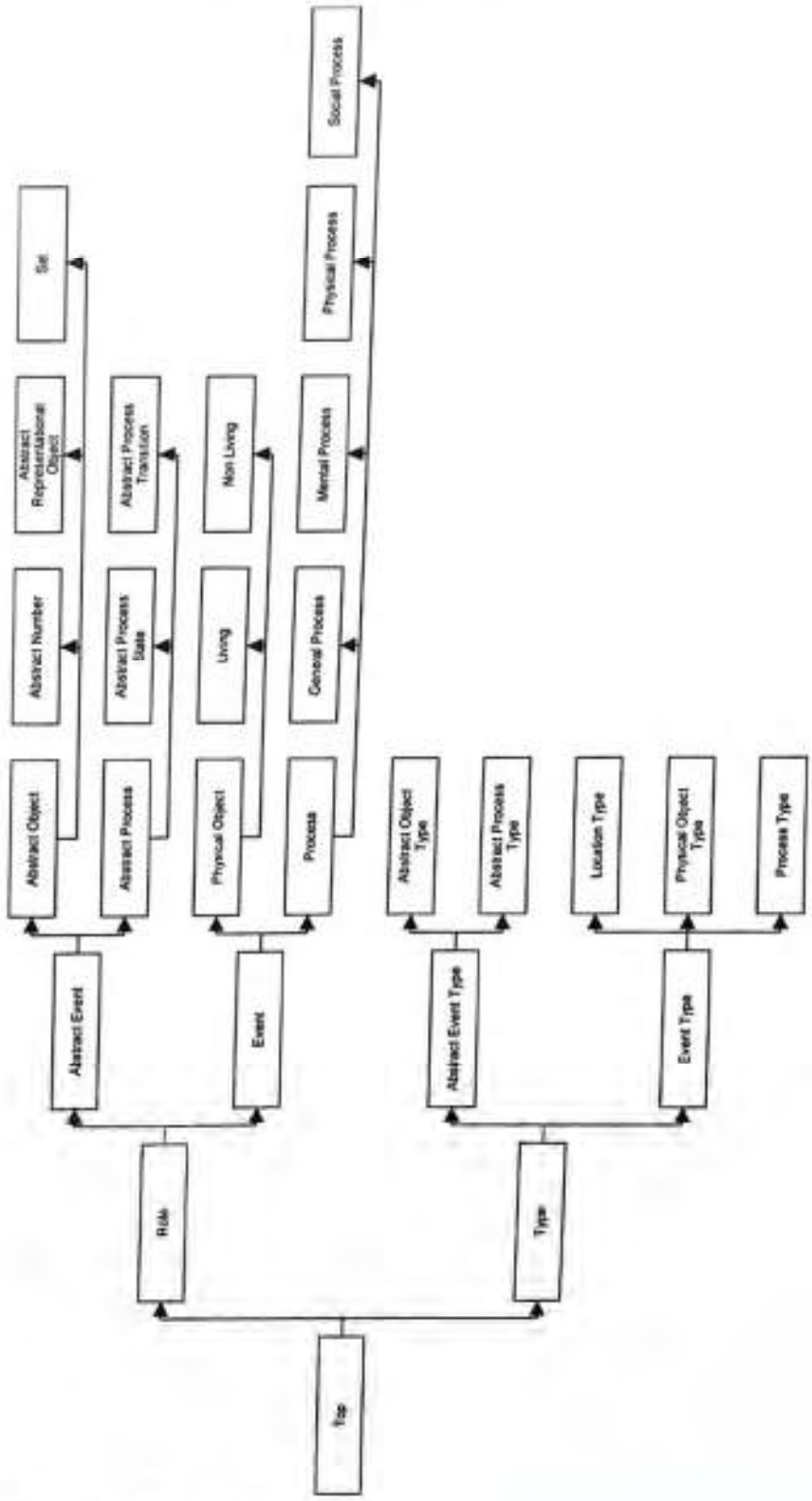

Fig. 1. Top-level part of the ontology

location in space and time. The class *PhysicalObject* describes any kind of objects we come in contact with — living as well as nonliving. These objects refer to different domains, such as *Sight* and *Route* in the tourism domain, *AvMedium* and *Actor* in the TV and cinema domain, etc., and can be associated with certain relations in the processes via slot constraint definitions.

The modeling of *Process* as a kind of event that is continuous and homogeneous in nature follows the frame semantic analysis used for generating the FRAMENET data (Baker et al., 1998). Based on the analysis of our dialogue data, we developed the following classification of processes (Fig. 2):

- *GeneralProcess*, a set of the most general abstract processes such as duplication, imitation, or repetition processes
- *MentalProcess*, a set of processes such as cognitive, emotional, or perceptual processes
- *PhysicalProcess*, a set of processes such as motion, transaction, or controlling processes
- *SocialProcess*, a set of processes such as communication or instruction processes

The *MentalProcess* subtree includes *CognitiveProcess*, *EmotionProcess*, and *PerceptualProcess*. Under *CognitiveProcess* we understand a group of processes that aim at acquiring information or making plans about the future. The further division of *EmotionProcess* into the following subclasses — *EmotionExperiencerObjectProcess* and *EmotionExperiencerSubjectProcess* — is due to the fact that an emotion can be either provoked by an object (e.g., The cry scared me) or can be experienced by an agent towards some object (e.g., I want to go home).

The *PhysicalProcess* has the following subclasses: The semantics of *ControllingProcess* presupposes the controlling of a number of artifacts, e.g., devices; *MotionProcess* models different types of agent's movement regarding some object or point in space; *PresentationProcess* describes a process of displaying some information by an agent, e.g., a TV program by an artificial character embedding the SMARTKOM system; *StaticSpatialProcess* consists in the agent's dwelling in some point in space; *TransactionProcess* presupposes an exchange of entities or services among different participants of the process. Another subclass of the *Process – SocialProcess* includes *CommunicativeProcess*, which consists in communicating by the agent a message to the addressee by different means, and *InstructiveProcess* which describes an interaction between an agent and a trainee.

3.2 Slot Hierarchy

The slot structure also reflects the general intention to keep abstract and concrete elements apart. A set of most general properties has been defined with regard to the role an object can play in a process: *agent*, *theme*, *experiencer*, *instrument* (or *means*), *location*, *source*, *target*, *path*. These general roles applied to concrete processes may also have subslots. Thus an agent in a process of buying (*TransactionProcess*) is a *buyer*; the one in the process of cognition is a *cognizer*. This way, slots can also build

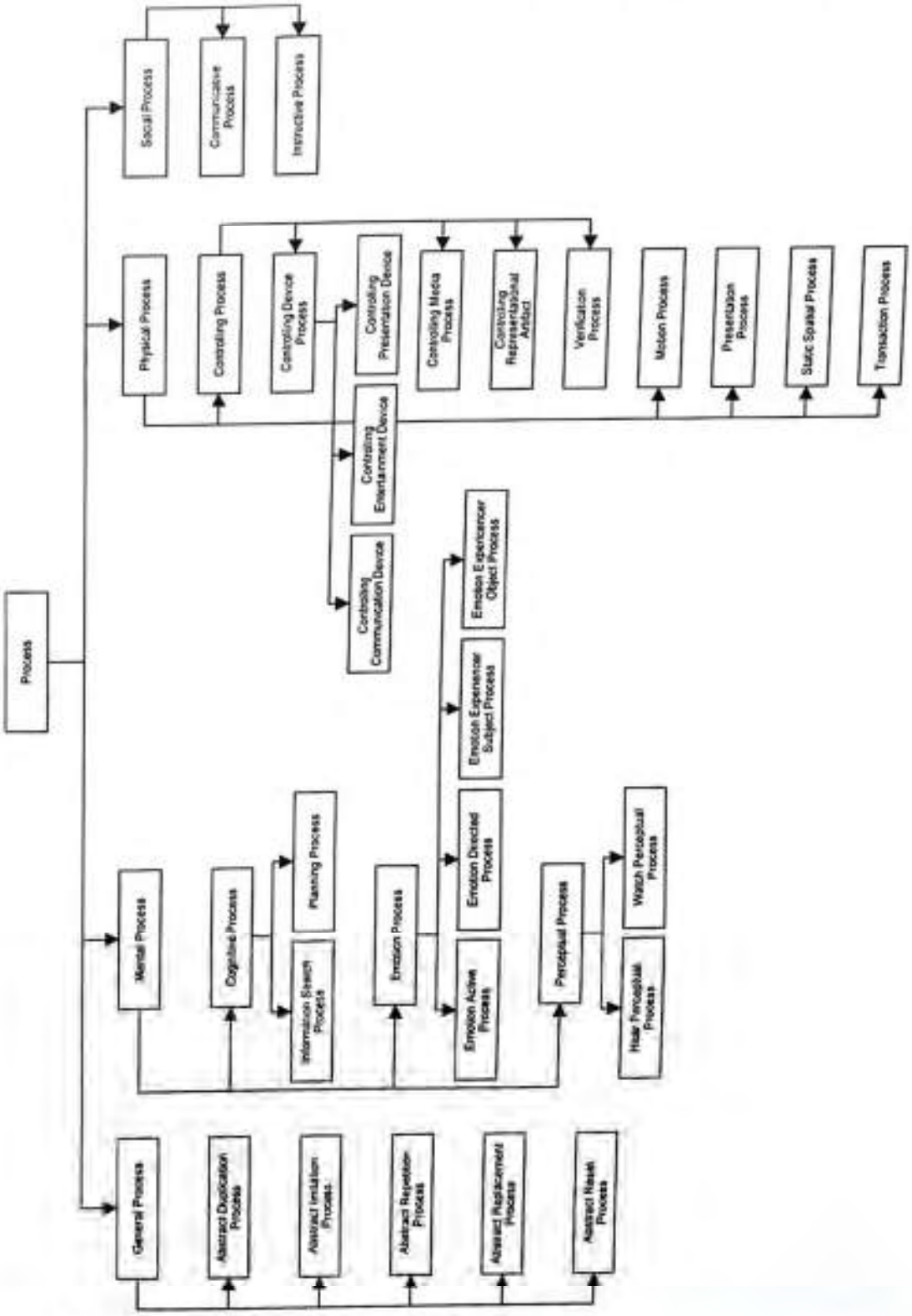

Fig. 2. Process hierarchy

hierarchical trees. The property *theme* in the process of information search is a required *piece-of-information*; in a presentation process it is a *presentable-object*, i.e., the item that is to be presented, etc.

Consider the class *Process*. It has the following slots: *begin-time*, a time expression indicating the starting point; *end-time*, a time expression indicating the time point when the process is complete; *state*, one of the abstract process states. These slots describe properties that are common to all processes, and as such they are inherited by all subclasses of the *Process* class.

An *EmotionExperiencerSubjectProcess* inherits the slots of the *Process* class, among them the slot *theme* that can be filled with any process or object (the basic idea is that any physical entity or the performance of any process can become an object of someone's emotion). It also has several additional properties such as *experiencer* to denote the one who undergoes the process, and *preference* to define the attitude an experiencer has to the object of its emotion.

3.3 Ontology Instances

Consider the definition of the *InformationSearchProcess* in the ontology. It is modeled as a subclass of the *CognitiveProcess*, which is a subclass of the *MentalProcess*. Therefore it inherits the following slot constraints: *begin-time*, a time expression indicating the starting time point; *end-time*, a time expression indicating the time point when the process is complete; *state*, one of the abstract process states, e.g., start, continue, interrupt, etc.; and *cognizer*, filled with a class *Person* including its subclasses.

The *InformationSearchProcess* features one additional slot constraint, *piece-of-information*. The possible slot fillers are a range of domain objects, e.g., *Sight*, *Performance*, or whole *sets* of those, e.g., *TvProgram*, but also processes, e.g., *ControllingTvDeviceProcess*. This way, an utterance such as:[5]

(1)	*I*	*hätte gerne*	*Informationen*	*zum*	*Schloss*
	I	would like	information	about the	castle

can be mapped onto the *InformationSearchProcess*, which has an agent of the type *User*, and a piece of information of the type *Sight*. *Sight* has a name of the type *Castle*. Analogously, the utterance:

(2)	*Wie*	*kann*	*ich*	*den*	*Fernseher*	*steuern*
	How	can	I	the	TV	control

can also be mapped onto the *InformationSearchProcess*, which has an agent of the type *User*, and has a piece of information of the type *ControllingTvDeviceProcess*.

Another example demonstrating how slot structures can be shared between some super- and subclasses: the subclass *AvEntertainment* inherits from its superclass *Entertainment* the following slots: *duration*, *end-time*, and *begin-time*, filled by the

[5] All examples are displayed with the Germano riginal on top and a glossed translation below.

TimeDuration and *TimeExpression*, respectively. The class *AvEntertainment* features two additional slots: *language*, whose filler is an individual *Language*, and *av-medium*, whose filler is a class *AvMedium*. The class *AvEntertainment* has further subclasses — *Broadcast*, which represents an individual entry in a TV program, and *Performance*, which models an entry in a cinema program. Both of them inherit the slots of the superclasses *Entertainment* and *AvEntertainment*, while also featuring their own additional slots, e.g., *channel* and *showview* for the *Broadcast*, *cinema* and *seat* for the *Performance*. This feature can be effectively utilized by a specific dialogue interpretation algorithm called OVERLAY(Pfleger et al., 2002).

4 Applications of the Ontology

There is no agreed methodology for ontology evaluation. In our opinion, the usefulness of an ontology can be evaluated by examining the ways it is employed within the system, allowing us to draw tentative conclusions as for the reusability of the ontology and its portability with respect to new applications and NLP tasks. The ontology described here is used by the complete core of the system, i.e., as a knowledge repository and model for language interpretation, discourse modeling, intention recognition and action planning (Engel, 2002; Löckelt et al., 2002; Alexandersson and Becker, 2003; Gurevych et al., 2003b; Porzel et al., 2003b). In the next sections we describe some examples of the usage within the project.

4.1 Semantic Coherence Scoring

One of the major challenges in making an MMDS reliable enough to be deployed in more complex real world applications is an accurate recognition of the users' input. In many cases both correct and incorrect representations of the users' utterances are contained in the automatic speech recognizer's *n*-best lists. Facing multiple representations of a single utterance poses the question, which of the different hypotheses most likely corresponds to the user's utterance. Different methods have been proposed to solve this problem. Frequently, the scores provided by the recognition system itself are used. More recently, also scores provided by the parsing system have been employed, e.g., Engel (2002). In this application, we propose a new ontology-based method and show that knowledge-based scores can be successfully employed to re-score the speech recognition output.

We introduced the notion of *semantic coherence* as a special measurement that can be applied to estimate how well a given speech recognition hypothesis (SRH) fits with respect to the existing knowledge representation (Gurevych et al., 2003a).

The software for scoring the SRHs and classifying them in terms of their semantic coherence employs the ontology described herein. This means that the ontology crafted as a general knowledge representation for various processing modules of the system is additionally used as the basis for evaluating the semantic coherence of sets of concepts.

The scoring software performs a number of processing steps:

- Converting each SRH into a concept representation. For this purpose, each entry of the system's lexicon was augmented with zero, one, or multiple ontology concepts.
- Converting the domain model, i.e., an ontology, into a directed graph with concepts as nodes and relations as edges.
- Scoring concept representations using a scoring metric based on the shortest path between concepts.

For example, in our data (Gurevych et al., 2002) a user expressed the wish to get more information about a specific church, as:

(3)	*Kann*	*ich*	*bitte*	*Informationen*	*zur*	*Heiliggeistkirche*
	May	I	please	information	about the	Church of Holy Spirit
	bekommen					
	get					

Looking at two SRHs from the ensuing n-best list we found that Example 5 constituted a suitable representation of the utterance, whereas Example 4 constituted a less adequate representation thereof, labeled accordingly by the human annotators:

(4)	*Kann*	*ich*	*Information*	*zur*	*Heiliggeistkirche*	*kommen*
	May	I	information	about the	Church of Holy Spirit	come
(5)	*Kann*	*ich*	*Information*	*zur*	*Heiliggeistkirche*	*bekommen*
	May	I	information	about the	Church of Holy Spirit	get

According to the lexicon entries, the SRHs are transformed into two alternative concept representations:

CR_1:{Person; Information Search Process; Church; Motion Directed Transliterated Process};
CR_2:{Person; Information Search Process; Church; Transaction Process}.

The scores are normalized as numbers on a scale from 0 to 1 with higher scores indicating better semantic coherence. Then, the resulting score assigned to Example 4 is 0.6, and the score of Example 5 is 0.75. The evaluation of the method against the hand-annotated corpus has shown that it successfully classifies 73.2 in a German corpus of 2.284 speech recognition hypotheses as either coherent or incoherent, given a baseline 54.55 from the annotation experiments (the majority class).

Semantic coherence scoring method has been successfully applied to the task of identifying the best SRH in the output of the automatic speech recognizer (Porzel and Gurevych, 2003). Its additional application is the calculation of a semantic coherence score for SRHs taking into account their conceptual context (Porzel et al., 2003a). Also, ontological concepts can be used to perform ontology-based automatic domain recognition and domain change detection.

4.2 Generating Interface Specifications

In this ontology application, we proposed to use the knowledge modeled in the ontology as the basis for defining the semantics and the content of information exchanged between various modules of the system (Gurevych et al., 2003b). In NLP systems, modules typically exchange messages, e.g., a parser might get word lattices as input and produce corresponding semantic representations for later processing modules, such as a discourse manager. The increasing employment of XML-based interfaces for agent-based or other multiblackboard communication systems sets a de facto standard for syntax and expressive capabilities of the information that is exchanged amongst modules. The content and structure of the information to be represented are typically defined in corresponding XML schemata (XMLS) or document type definitions (DTD).

As discussed above, ontologies are a suitable means for knowledge representation, e.g., for the definition of an explicit and detailed model of a system's domains. That way, they provide a shared domain theory, which can be used for communication. Additionally, they can be employed for deductive reasoning and manipulations of models. The meaning of ontology constructs relies on a translation to some logic. This way, the inference implications of statements, e.g., whether a class can be related to another class via a subclass or some other relation, can be determined from the formal specification of the semantics of the ontology language. However, this does not make any claims about the syntactic appearance of the representations exchanged, e.g., an ordering of the properties of a class.

An interface specification framework, such as XMLS or DTD, constitutes a suitable means for defining constraints on the syntax and structure of XML documents. Ideally, the definition of the content communicated between the components of a complex dialogue system should relate both the syntax and the semantics of the XML documents exchanged. Those can then be seen as instances of the ontology represented as XMLS-based XML documents. However, this requires that the knowledge, originally encoded in the ontology, is represented in the XMLS syntax.

The solution proposed states that the knowledge representations to be expressed in XMLS are first modeled in OIL-RDFS or DAML+OIL as *ontology proper*, using the advantages of ontology engineering systems available, and then transformed into a communication interface automatically with the help of the software developed for that purpose.[6]

Employing this approach, XMLS and DTDs are created such that they:

- stay logically consistent
- are easy to manage
- enable a straightforward mapping back to the respective knowledge representation for inference

[6] This is a free software project. The package and respective documentation can be obtained from http://savannah.nongnu.org/projects/oil2xsd.

- allow the handling of a range of NLP tasks immediately on the basis of XMLS, e.g., the discourse module operates on the XML schema obtained via ontology transformation(Alexandersson and Becker, 2003).

The resulting schemata capture the hierarchical structure and a significant part of the semantics of the ontology. We therefore provide a standard mechanism for defining XMLS-based interface specifications, which are *knowledge rich*, and thus can be used as a suitable representation of domain and discourse knowledge by NLP components. Since the software that has been developed completely automates the transformation process, the resulting XMLS are congruent with the XML schema specifications. Furthermore, the ontology can be reused in multiple systems as a single ontology can be used to generate application-specific communication interfaces.

However, the main advantage of our approach is that it combines the power of ontological knowledge representation with the strengths of XMLS as an interface specification framework in a single and consistent representation. Our experience shows, this would not have been possible for a complex dialogue system if XML schemata were defined from scratch or handcrafted, and constitutes a step toward building robust and reusable NLP components.

5 Conclusion

An ontology-based approach to modern MMDS uses a single knowledge representation. We described the major modeling principles and the design choices made. Furthermore, we sketched some examples of the ontology application within the system.

Together these examples suffice to demonstrate the benefits of using a single knowledge representation throughout a dialogue system as opposed to using multiple knowledge representations and formats. An additional advantage of such a homogeneous world model that defines the processing interfaces as well as the system's world knowledge is that no costly mappings between them are no longer necessary. This means that modules receive only messages whose content is congruent to the terminological and structural distinctions defined in the ontology.

Our additional concern while designing the ontology was the reusability of this component within our MMDS as well as other NLP systems. So far, the top-level ontology proved stable. We found the extensions on the lower levels of the ontology to be comparatively cheap. This single knowledge base was successfully tested and applied to multiple NLP problems, e.g., resolving bridging expressions in texts as well as for the resolution of metonymical and polysemous utterances next to defining communication interfaces for NLP components of the system and scoring of speech recognition hypotheses described above. Consequently, the approach to knowledge representation and the ontology itself can be reused in the future design of MMDS.

Acknowledgments

This work was partially funded by the German Federal Ministry of Education, Science, Research, and Technology (BMBF) in the framework of the SMARTKOM project under Grant 01 IL 905 K7 and by the Klaus Tschira Foundation. We would also like to thank Tatjana Medvedeva, Elena Slinko, Lutz Wind, and Iryna Zhmaka for their valuable contributions.

References

J. Alexandersson and T. Becker. The Formal Foundations Underlying Overlay. In: *Proc. 5th Int. Workshop on Computational Semantics (IWCS-5)*, pp. 22–36, Tilburg, The Netherlands, February 2003.

J.F. Allen, B. Miller, E. Ringger, and T. Sikorski. A Robust System for Natural Spoken Dialogue. In: *Proc. 34th Annual Meeting of the Association for Computational Linguistics*, pp. 62–70, Santa Cruz, CA , June 1996.

C.F. Baker, C.J. Fillmore, and J.B. Lowe. The Berkeley FrameNet Project. In: *Proc. COLING-ACL'98*, pp. 86–90, Montreal, Canada, 1998.

R. Engel. SPIN: Language Understanding for Spoken Dialogue Systems Using a Production System Approach. In: *Proc. ICSLP-2002*, pp. 2717–2720, Denver, CO, 2002.

D. Fensel, F. van Harmelen, I. Horrocks, D.L. McGuinness, and P.F. Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

G. Ferguson, J.F. Allen, B. Miller, and E. Ringger. The Desgin and Implementation of the TRAINS-96 System. Technical Report 96-5, University of Rochester, New York, 1996.

N. Guarino and R. Poli. Formal Ontology in Conceptual Analysis and Knowledge Representation. *Special issue of the Int. Journal of Human and Computer Studies*, 43, 1995.

N. Guarino and C. Welty. A Formal Ontology of Properties. In: R. Dieng and O. Corby (eds.), *Proc. EKAW-2000: The 12th Int. Conf. On Knowledge Engineering and Knowledge Management*, vol. 1937, pp. 97–112, Berlin Heidelberg New York, 2000. Springer.

I. Gurevych, R. Malaka, R. Porzel, and H.P. Zorn. Semantic Coherence Scoring Using an Ontology. In: *Proc. Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting 2003*, pp. 88–95, Edmonton, Canada, 2003a.

I. Gurevych, R. Porzel, E. Slinko, N. Pfleger, J. Alexandersson, and S. Merten. Less Is More: Using a Single Knowledge Representation in Dialogue Systems. In: *Proc. HLT-NAACL'03 Workshop on Text Meaning*, pp. 14–21, Edmonton, Canada, May 2003b.

I. Gurevych, R. Porzel, and M. Strube. Annotating the Semantic Consistency of Speech Recognition Hypotheses. In: *Proc. 3rd SIGdial Workshop on Discourse and Dialogue*, pp. 46–49, Philadelphia, PA, July 2002.

G. Herzog, H. Kirchmann, S. Merten, A. Ndiaye, and P. Poller. MULTIPLATFORM Testbed: An Integration Platform for Multimodal Dialog Systems. In: *Proc. HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pp. 76–83, Edmonton, Canada, May 2003.

M. Löckelt, T. Becker, N. Pfleger, and J. Alexandersson. Making Sense of Partial. In: C.M. Johan Bos, Mary Ellen Foster (ed.), *Proc. 6th Workshop on the Semantics and Pragmatics of Dialogue (EDILOG 2002)*, pp. 101–107, Edinburgh, UK, September 2002.

N. Pfleger, J. Alexandersson, and T. Becker. Scoring Functions for Overlay and Their Application in Discourse Processing. In: *Proc. KONVENS 2002*, pp. 139–146, Saarbruecken, Germany, September–October 2002.

R. Porzel and I. Gurevych. Contextual Coherence in Natural Language Processing. In: P. Blackburn, C. Ghidini, R.M. Turner, and F. Giunchiglia (eds.), *Proc. of Modeling and Using Context – 4th International and Interdisciplinary Conference, CONTEXT 2003*, LNCS 2680, pp. 272–285, Berlin Heidelberg New York, 2003. Springer.

R. Porzel, I. Gurevych, and C. Müller. Ontology-Based Contextual Coherence Scoring. In: *Proc. 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan, July 2003a.

R. Porzel, N. Pfleger, S. Merten, M. Loeckelt, I. Gurevych, R. Engel, and J. Alexandersson. More on Less: Further Applications of Ontologies in Multi-Modal Dialogue Systems. In: *Proc. 3rd IJCAI 2003 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Acapulco, Mexico, 2003b.

S.J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

Speech Recognition

André Berton, Alfred Kaltenmeier, Udo Haiber, and Olaf Schreiner

DaimlerChrysler AG, Research and Technology, Ulm, Germany
{andre.berton,alfred.kaltenmeier,udo.haiber,
olaf.schreiner}@daimlerchrysler.com

Summary. The human machine interaction of SMARTKOM is a very complex task, defined by natural, spontaneous language, speaker independence, large vocabularies, and background noises. Speech recognition is an integral part of the multimodal dialogue system. It transforms the acoustic input signal into an orthographic transcription representing the utterance of the speaker. This contribution discusses how to enhance and customize the speech recognizer for the SMARTKOM applications. Significant improvements were achieved by adapting the speech recognizer to the environment, to the speaker, and to the task. Speech recognition confidence measures were investigated to reject unreliable user input and to detect user input containing unknown words, i.e., words that are not contained in the vocabulary of the speech recognizer. Finally, we present new ideas for future work.

1 Introduction

Speech recognition as part of multimodal dialogue systems transforms the acoustic input signal into an orthographic transcription representing the utterance of the speaker. This contribution discusses how to optimize the speech recognizer for the SMARTKOM applications. It starts with a general description of the speech recognition engine, which explains the signal processing, vector quantization, and decoding of the recognizer. The decoding that relies on acoustic Hidden Markov Models (HMMs) and statistical language models is described, particularly the training procedure.

After describing the speech recognition basics, we focus on the four main research interests for speech recognition in multimodal environments, namely how to adapt to the environment, to the speaker, to the task, and how to estimate confidences for the correctness of the words hypothesized by the recognizer. The system adapts itself to the environment by enabling Barge-In using an echo cancellation technique in the acoustic device, by reducing environmental noises using spectral substraction, and by adjusting the input signal level using automatic gain control. Several supervised and unsupervised speaker adaptation techniques are presented: vocal tract normalization, maximum likelihood linear regression, codebook mixture, and speaker-dependent training. The SMARTKOM system adapts itself to the task or

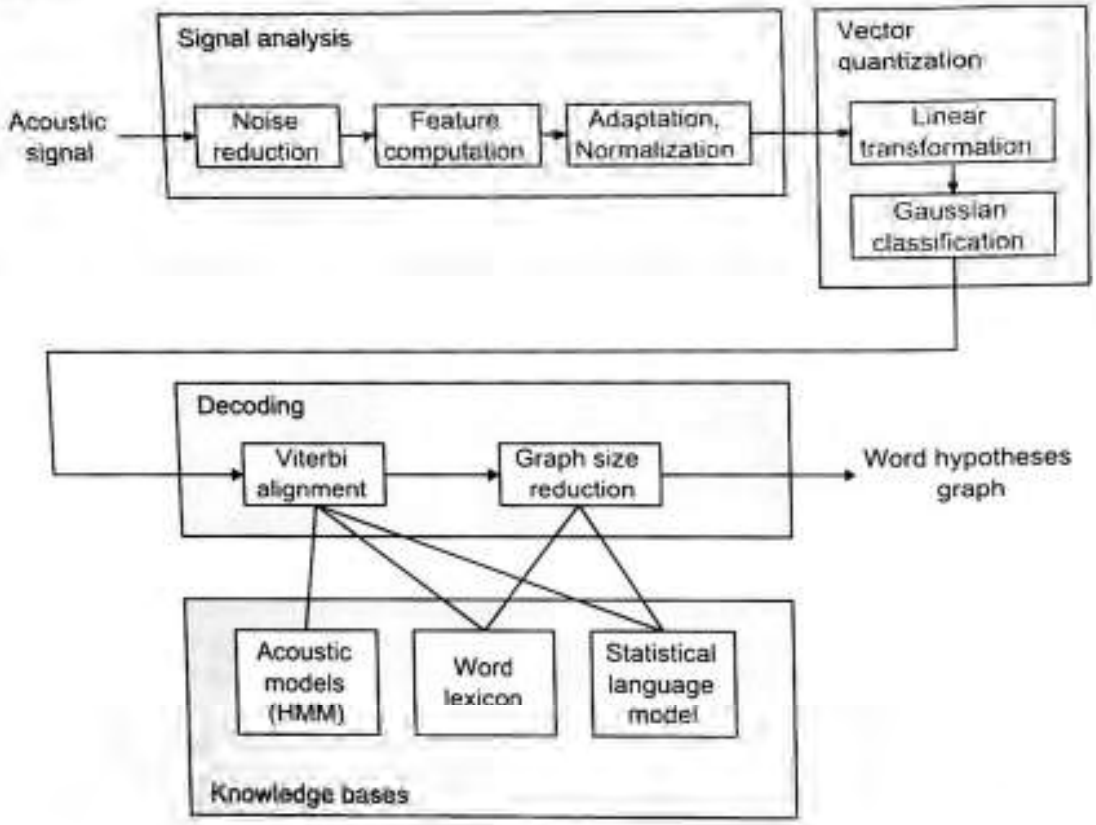

Fig. 1. Setup of the DaimlerChrysler speech recognition engine

application by dynamically updating the vocabulary and the members of the categories in the statistical language model. The last part of the research interests introduces two methods for word confidence estimation: The first method directly calculates the confidence from very large word graphs, while the second method collects features that correlate with the word's correctness and estimates confidences using known classification techniques. Furthermore, we discuss the problem of unknown words. These are words that the speaker may utter but that are not included in the recognizer vocabulary. We introduce an approach how to model such words in order to detect them. Finally, conclusions about the results achieved in the research areas are drawn, and ideas for future work are presented.

2 Speech Recognition Engine

An automatic speech recognizer transforms the acoustic signal into a sequence of hypothesized words in orthographic representation. The speech recognizer implemented at DaimlerChrysler follows the semicontinuous Hidden Markov Model paradigm (SCHMM), which consists of three main processing stages: signal analysis, vector quantization, and decoding (Fig. 1).

Signal processing transforms the speech signal into a vector of features that represent the phonetic contents of the signal. Details about signal analysis can be found in the sections that describe the adaptation to the environment and to the speaker. Vector quantization and decoding are explained in detail in the following sections. The decoding process needs certain knowledge bases to find the best matching word sequences. We distinguish three knowledge bases:

- acoustic models: HMMs
- word lexicon: vocabulary
- language model: statistical trigram model

2.1 Feature Transform and Classification (Vector Quantization)

The second block of the speech recognition engine (Fig. 1) includes feature transformation and classification. After signal analysis 9 adjacent mel-frequency cepstrum (MFC) feature vectors (covering 90 ms of the input speech signal) were combined to form a 117-dimensional super-vector. Linear transforms such as linear discriminant analysis (LDA) or principal component analysis (PCA) can be applied to reduce the vector dimension, from 117 to 32 in our case. Transformed vectors have an inherent component ordering; the first component is the most important one for separation or reconstruction, resp. Transform matrices are computed during forward/backward training of acoustic HMMs. Both transforms yield very similar recognition results with LDA slightly superior to PCA.

Transformed feature vectors are vector-quantized (VQ) with a Gaussian classifier consisting of 1024 classes with a full covariance matrix (full matrix inverse) for each class. Due to the symmetry of covariance matrices and inverses, 33*34/2=561 parameters have to be stored for each class, i.e., 561*1024 parameters for the whole classifier. Taking advantage of component ordering, classification can be performed in a hierarchical structure of classifier levels. Each level takes into account eight components of a transformed input vector: The first level uses the eight most important components and estimates likelihoods for all classes. The second level then adds another eight components and reestimates likelihoods only for classes considered important by the first level. This procedure is iterated for all four levels. Nonhierarchical classification requires 561*1024*100 scalar product operations (multiply/add) per second; our approach reduces computational requirements by more than a factor of 5 ($\approx$10,000,000 operations per second) without *any loss* in recognizer performance.

The classifier output is finally limited to a given likelihood threshold ($\varepsilon = 10^{-3}$) and ten classes and associated likelihoods at maximum. The average number of classes is about 8.0 for our training and test databases.

2.2 Acoustic Models and Training

Recognition is based on statistical models — HMMs — of phonetic units. Many recognition systems described in the literature use generalized triphone models. Our approach uses a more extended set:

- Several models for background noise
- Models for nonverbal utterances (laughter, coughing, clearing one's throat, smacking, breathing, ...)
- Whole-word models for important content words (yes, no, cardinal and ordinal numbers, navigation or entertainment applications, ...)
- 30 models for spelling (a, b, ..., z, ä, ö, ü, and ß)
- Context-dependent models of frequent initial or final consonant clusters of syllables, e.g., |Str...| as in "Straße" or |...nst| as in "Dienst"
- Context-dependent models of diphthongs and vowel–r combinations, e.g., |...aU...| as in "Baum" or |...O6...| as in "dort"
- Context-dependent triphone or generalized triphone models, biphone and monophone models as usual

Model parameters are trained with the well-known forward/backward algorithm in an iterative manner, with four training steps in our case:

1. Silence and background noise models, keep all other models fixed
2. + Whole word, alphabet, nonverbal utterances, monophone models
3. + Initialize triphone and syllable group models with monophone parameters and train the complete model set together
4. Generalized consonant and vowel clusters for out-of-vocabulary words (OOV)

After training, model parameters (weights of classes or emissions) are clipped to a certain threshold ($\varepsilon = 10^{-6}$). Hence our recognizer has the following setup:

- 1750 models with 7300 states
- 715,000 weights (instead of ≈7,400,000 weights for full storage)

Neglecting the ε-weights significantly reduces storage and computational requirements without *any loss* in performance.

The LDA approach for vector quantization requires some class information that is defined by states or state-clusters. Basically, we have 7300 states (or classes) that have to be reduced to 1024 for VQ. Class reduction is performed by bottom-up entropy clustering; most similar states are automatically grouped together in an iterative clustering procedure taking the HMM type into account:

- States of noise models may only be clustered with states of other noise or silence models (noise/silence forms a separate superclass).
- States of alphabet models may be clustered only with states of other alphabet models (alphabets forms another superclass).
- All other model states may be clustered together.

2.3 Decoding and Word Hypotheses Graph Processing

Speech recognition is based on a frame-synchronous Viterbi algorithm and uses following knowledge bases (Fig. 1):

- HMMs as described above
- A lexicon tree constituted from HMMs and the recognizer vocabulary, which may be updated dynamically
- A class-dependent language model (LM) for the active vocabulary, which may be updated dynamically, too

The *nodes* of the tree represent *states* of HMMs; the *arcs* represent *transitions* to other states with associated state-transition probabilities and approximated LM unigram and bigram values.

Decoding is performed on frame-by-frame basis:

- For each frame vector represented by VQ classes and likelihoods, observation probabilities are computed for all states. This requires approximately 7300 * 8.0 * 100 ≈ 6,000,000 operations (multiply/add) per second.
- Using these acoustic probabilities, all active paths through the lexicon tree are extended for one frame. At word ends (leaf nodes of the tree) we not only store the best path and its associated word sequence, but we also store all promising word candidates forming the so-called word hypotheses graph (WHG). The nodes of this graph represent time stamps, and the arcs represent words with associated acoustic probabilities.
- If words end at a time frame, the lexicon tree is "restarted" again.

The WHG is finally reduced in the last block of the decoder module; i.e., words and paths may be eliminated from the graph. Reduction is performed on a node-by-node basis. At each "anchor" node, WHG size is reduced considering acoustic and LM probabilties. Anchor nodes are normally constituted by speech pauses or well-recognized words, i.e., words with no competing alternatives.

For the SMARTKOM lexicon with about 5000 words, our recognizer requires about *0.4-times* real-time on a IBM T30 laptop computer. Recognition was evaluated with three test sets. Table 1 gives word error rates (WER) and sentence error rates (SER).

Table 1. Recognition results for the final SMARTKOM recognizer and three test sets "Public", "Mobil", "EML"

Test set name	No. of sentences	No. of words	WER	SER
Public	1144	7411	11.2%	36.1%
Mobil	1136	4874	7.9%	19.3%
EML	1508	8788	21.1%	49.2%

3 Adaptation to the Environment

The SMARTKOM system must be able to adapt itself to the continuously changing environmental conditions, like noises and overlapping speech. A great variety of

instationary noises make the complex task of speech recognition even harder. This section presents three techniques, namely echo cancellation, noise reduction, and automatic gain control, which are applied in the SMARTKOM system in order to achieve better system behavior due to environmental adaptation.

3.1 Echo Cancellation and Barge-In

One of the common expectations that the majority of state-of-the-art dialogue systems do not meet is that the user can interrupt the system anytime, even while the system speaks to the user. This functionality that allows the user to speak his request while the system is speaking is called *Barge-In*. The user expects an appropriate system response to the request on interpretation level. This kind of interaction relates to all modules of the dialogue system. Barge-In can be applied in the following scenarios:

1. Shortcut: e.g., early selection of a list item (before the end of the list)
2. Cancel: e.g., interruption due to misunderstandings or irrelevant system output
3. Back-channel: maintainance of the communication by the user during long system outputs (e.g., user utterance of "Uhm")
4. Off-talk: utterances not directed to the SMARTKOM system

The following technical requirements can be defined for handling Barge-In intelligently:

1. Two input channels (system output and user input)
2. Speech recognition in parallel with speech synthesis output
3. Echo cancellation of the synthesis prompt
4. Back-channel models (user signals attention, but does not intend to interrupt): uhm, yeah, right, okay, I see
5. Semantic constructions for back-channel models
6. Contextual interpretation
7. Dialogue control models for different types of Barge-In
8. Time alignment of synthesis, generation, and audio output (duration estimation for utterances and segments of utterances)
9. Optimization of recognition control by empirical tuning

The most important component that needs to be added to the dialogue system to allow Barge-In is *echo cancellation*. The microphone of the dialogue system needs to be open almost anytime, so that the system responses from the loudspeakers will be included in the signal received by the microphone additionally to the utterances of the user. The sound of the speakers reflected by the walls of the room arrives at slightly different times and volumes at the microphone. Larger rooms lead to longer signal response times. The speech recognizer requires a speech signal that is distorted as little as possible to achieve high recognition accuracy. Therefore, the echo cancellation needs to substract the system output from the input signal on the microphone in order to ensure that the input signal of the recognizer contains only the

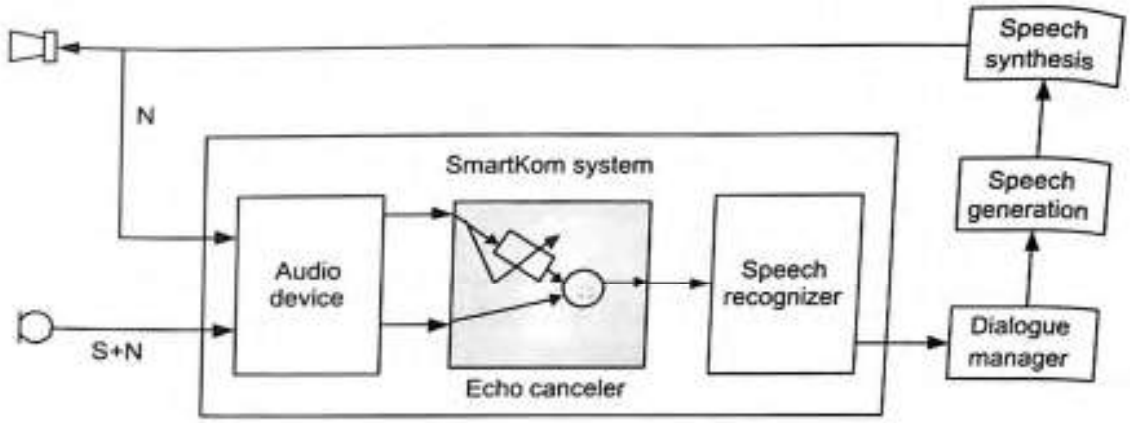

Fig. 2. Technical setup of the echo cancellation in the SMARTKOM system

speech uttered by the user. The noise signal N (system response) is simultaneously recorded with the input signal S from the microphone on a different channel of the audio device (Fig. 2), since only synchronous computation allows for canceling the system response.

The following three tasks of an echo canceler can be distinguished:

1. Estimation of the signal impulse response of the system output (variation of the signal between loudspeaker and microphone)
2. Convolution of the speaker output signal with the system impulse response
3. Substraction of the microphone signal after convolution

The impulse response consists of the room impulse response and the impulse response of the loudspeaker and microphone. It is iteratively estimated by the language model score (LMS) algorithm.

The echo cancellation has been implemented in cooperation with MediaInterface Dresden. DaimlerChrysler provided the echo cancellation algorithm, and MediaInterface integrated it in the audio device and optimized it for the particular environments.

The echo cancellation algorithm adapts itself to environmental changes and is a pure software solution. Experiments have shown that the recognizer's word error rate decreases from 100% to about 30% for Barge-In and continuous utterances if echo cancellation is applied. That means that Barge-In can only be allowed if a good echo canceler is at hand. The echo canceler has still to be optimized for every new room because of varying sound characteristics (length of filter stages and adaptation coefficients). The cancellation leads to an average signal gain of about 40 dB at very low computational costs and very little delay caused by the incremental processing. However, the estimation of the impulse responses needs a certain onset time, because it follows the principle of iterative error minimization. It is still a scientific challenge to shorten the onset time.

3.2 Noise Reduction

Another aspect of environmental adaptation is a more precise modeling of speech and noises to be able to separate one from another. Noises can either be substracted from the speech signal or additionally modeled.

The technique commonly applied to reduce stationary noises is *spectral substraction* (SPS). An adaptive version of that method has been developed at DaimlerChrysler. This technique reduces noises in the spectral representation of the speech signal. The stationary noises that will be substracted from the signal can be estimated in speech pauses. The SPS requires a well-working voice activity detector, which segments the signal. The automatic gain control that has been implemented to improve voice activity detection will be described in Sect. 3.3.

We also investigated a new technique for robust speech recognition, namely the *missing data method*. This method assumes that not all frequency bands are distorted by noise. The aim of our investigations was to recognize speech only in the nondistorted frequency bands. The selection of nondistorted bands proved to be very difficult. The method performed better on noisy speech than a speech recognizer that was trained only on clean speech, but worse than a recognizer also trained on noisy speech.

So far, we described methods to reduce noise in the signal. The following methods, however, will try to model noise as precisely as possible and to separate it from speech. That allows us to hypothesize noises on the HMM level and to reject them if needed. We trained various HMMs for technical and human noises, so that such speech segments will not be mixed up with words.

We also considered an approach called *parallel model combination* (PMC) to additively model noises by a particular mean estimation and parallel noise models. This approach led to recognition improvements only for recognizers trained on clean speech. Since the DaimlerChrysler recognition engine is usually trained on noisy speech, there has been no need to apply PMC so far.

Finally, we investigated a promising new hybrid method: *multistream recognition*, since such an approach combines the advantages of neural networks and HMMs. That method employs a neural network (multilayer perceptron) to estimate the posterior probabilities of each HMM state for a given speech signal on frame basis. These posterior probabilities are considered as emission probabilities in the HMMs and can be used directly in Viterbi decoding. Various acoustic feature streams and their combinations have been investigated in the acoustic front end:

1. MFC: mel-frequency cepstrum (the classical short-term feature stream)
2. PLP: perceptual linear prediction (oriented toward human perception)
3. MSG: modulation spectogram (covers long-term signal characteristics)

The Performance of the new method, which includes the different feature streams, was superior to our standard recognizer on a speech database covering digit sequences with artificially added noises (Aurora-2). In particular, the combination of PLP and MSG feature streams led to a significant improvement in recognition accuracy of up to 60%. That improvement can be explained by the complementary

properties of the feature streams: the PLP features cover the short-term properties of the signal (about 20 ms), while the MSG features represent the long-term characteristics (about 2 s), like reverberation. However, the new methods did not prove successful for large vocabulary continuous speech recognition, because the estimation of posterior probabilities was not discriminative enough for large sets of HMMs. Therefore, the multistream approach has not been further considered for the SMARTKOM project.

3.3 Automatic Gain Control

Adapting the speech recognizer to the environment also means coping with various signal levels. Some users speak louder than others, and there are different microphone setups and various distances between speaker and microphone. The speech recognizer however performs best if the signal level is well-adjusted to about half the maximum level. Such an adjustment is performed by an *automatic gain control* (AGC), improving the speech-pause segmentation of the voice activity detector and thus the recognition accuracy. In contrast to the standard approach, the method developed at DaimlerChrysler adjusts the signal only in speech segments, while non-speech segments remain unchanged. This has a very positive effect on segmentation accuracy, particularly at low signal-to-noise ratios. The AGC makes sure the input signal is not overamplified, and thus distorted, by converting the signal for the speech recognizer to the range of 60% of the maximum amplitude. Figure 3 shows an example of the effect of AGC by amplifying the low signal in the upper part to a level well-suited for speech recognition.

4 Adaptation to the Speaker

Speaker adaptation plays a great role in SMARTKOM, since the multimodal dialogue system should be able to handle the requests of any user. Different utterances of each speaker vary greatly, but utterances of different speakers vary even more. Speaker adaptation is intended to improve recognition accuracy, so that it contributes largely to user acceptance. Adaptation can be realized by supervised and unsupervised methods. Unsupervised adaptation is performed while the user interacts with the system without precisely knowing the transcription of the user utterance. The transcription of user utterance is known in supervised adaptation, where the user is requested to speak some predefined sentences (typically 10 to 100) in a training phase. In this section we present three unsupervised methods: vocal tract normalization (VTN), maximum likelihood linear regression (MLLR), and codebook mixture (CBM), and one supervised adaptation method: speaker dependent training (SDT).

A particular motivation for performing speaker adaptation is to better cope with problem speakers. Problem speakers are characterized by one or more of the following phenomena: dialect, unclear articulation or strong coarticulation, and syntactical mistakes. These phenomena decrease recognition accuracy significantly, so that each of the following methods should contribute to improving adaptation.

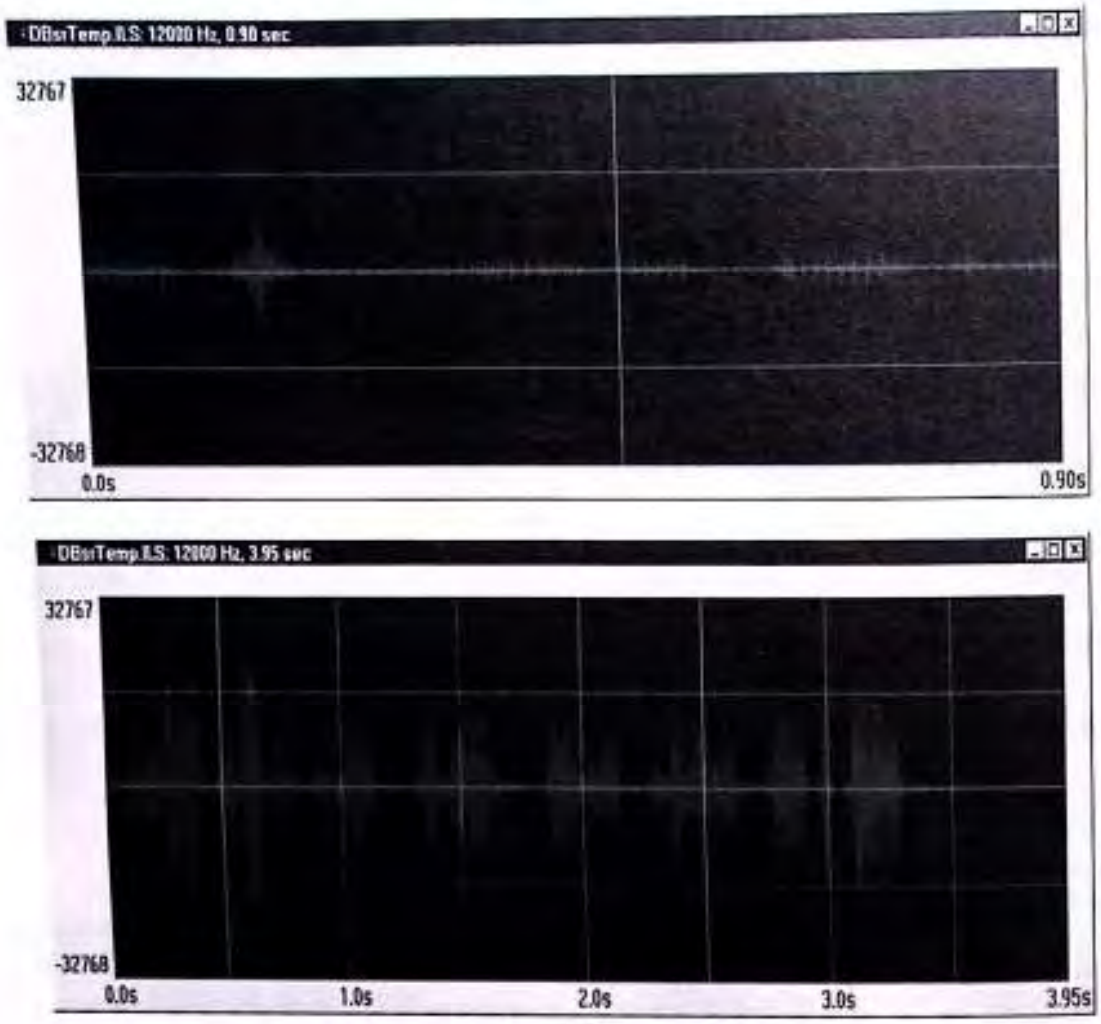

Fig. 3. The effect of automatic gain control on a lowlevel speech signal

4.1 Vocal Tract Normalization and Maximum Likelihood Linear Regression

The well-known speaker adaptation methods: Vocal tract normalization and maximum likelihood linear regression had already been integrated in the DaimlerChrysler speech recognizer before the beginning of the SMARTKOM project. Both methods were optimized for the SMARTKOM applications. VTN tries to normalize the vocal tract length of all speakers using a pitch estimation and a spectral adjustment (shift). The dynamic estimation of the shift factor was optimized. Normalization on phrase level proved superior to normalization on frame level.

Speaker adaptation by MLLR is based on shifting the classes of the vector quantization codebook in order to better represent the speaker properties. The HMM state sequence that is needed for adaptation can be directly extracted from Viterbi decoding or in an additional forced alignment of the hypothesized words. Codebook classes may also be marked invariant in order to prevent them from transformation.

Some HMM states, like pauses and noises, are excluded from MLLR, so that they do not contribute to the estimation of the transformation.

Each of the two adaptation methods, VTN as well as MLLR, leads to an average improvement in recognition accuracy of between 1% and 2% absolute. Both methods are applied in the SMARTKOM system since the improvements can be provided at low computational cost.

4.2 Speaker-Dependent Training

The SDT method explicitely adapts the vector quantization codebooks to a particular speaker. Therefore, the speaker has to utter 20 to 100 sentences that are given by the system. This supervised training method should be considered only if the voice of the speaker is recognized with low accuracy and if the speaker frequently uses the system. SDT can be applied in cars, but not while driving since such a training procedure would distract the driver significantly. The Baum–Welch algorithm is incorporated for SDT, because it is easier to implement than the maximum a posteriori (MAP) method, and also allows us to configure the recognizer user-specifically simultaneously with the speaker getting used to the system itself. Experiments performed on test sets of city names, spelling units, and digits sequences showed a maximum relative improvement in recognition accuracy of 33% for digit sequences and an average improvement of about 20% for the entire test set (see Table 2). The SDT method proved to be particularly suitable for problem speakers since for these speakers it achieved an average improvement in recognition accuracy of almost 25%. Empirical studies indicated that the more sentences the speaker enters for SDT, the better the adaptation result. However, uttering 100 sentences is quite a burdensome task and should be considered the maximum amount for training.

Table 2. Recognition accuracy and its relative improvement for speaker-dependent training for all speakers and the problem speakers using 20 to 100 utterances

	All speakers	Problem speakers
Baseline	22.5%	28.2%
SDT-20	20.2%	24.3%
SDT-50	19.0%	22.5%
SDT-100	18.0%	21.2%
Improvement (rel.)	20.0%	24.8%

4.3 Codebook Mixture (CBM)

The mixture of the vector quantization codebooks can be considered as an extension of the SDT method. SDT is incorporated to compute a vector quantization codebook for every single speaker given all his or her utterances in the training set. Principal component analysis is then used to reduce the number of codebooks to 10 to 100 prototypic mixtures. These given prototype codebooks can be linearly combined to give

a whole new range of codebook means. The prototype codebooks are loaded during the initialization phase of the recognizer, while the weights for the linear combination are estimated online, so the method belongs to the category of unsupervised adaptations. The experimental results in Table 3 prove that CBM technique outperforms the MLLR and VTN techniques, particularly for small adaptation sets. It achieves about 60% of the improvement of the SDT method, but has the clear advantage of an unsupervised method. That is, the user does not need to undergo a special training procedure. Therefore, CBM is used in SMARTKOM, while SDT is not.

Table 3. Recognition accuracy and its relative improvement for codebook mixture training for all speakers and the problem speakers using 20 to 100 utterances

	All speakers	Problem speakers
Baseline	22.5%	28.2%
CBM-10	20.7%	25.7%
CBM-100	19.6%	23.6%
Improvement (rel.)	12.5%	16.2%

5 Confidence Measures

Confidence measures have become common practice in state-of-the-art speech recognition systems. In this contribution we focus on confidence estimations on the word level, i.e., the confidence of each word is estimated as the posterior probability for the word being correct. The theoretical framework for confidence estimation is introduced in Young (1994) and Eide et al. (1995).

Confidence estimation methods strongly depend on the decoding algorithm of the recognizer. N-Best list decoders (Weintraub, 1995), word graph decoders (Schaaf and Kemp, 1997), and hybrid HMM/ANN (Artificial Neural Networks) decoders (Williams and Renals, 1997) all require different confidence estimators. This report compares two different estimation methods for word graph decoders. The first method computes the confidence from a large word graph (Wessel et al., 1998), and the second collects a set of features that correlate with the correctness of a word (Eide et al., 1995). Both methods are modified and extended in this contribution.

Our aim is to find the best-suited confidence estimation method for in-car recognition tasks. So far, command speech is the common way to operate phone, radio, CD player, and navigation systems in a car. The Linguatronic II system, which is available as a supplementary feature in Mercedes-Benz cars, is one such advanced product. Since customers find remembering a set of predefined commands a burdensome task, they demand more natural ways to input speech. SMARTKOM is a first step toward natural speech interaction covering complex navigation, tourist information, and parking place information tasks.

Many applications of confidence measures have already been presented: rejection, word error minimization (Fetter et al., 1996), speaker adaptation (Pitz et al., 2000), and language identification (Metze et al., 2000). In car applications, we use confidence measures in the framework of dialogue systems, where the dialogue manager decides what action to take next based on the given confidence level. A low confidence is interpreted as rejection and leads to a repetition prompt. A medium confidence triggers a confirmation step, and a highly confident recognition result causes the user input to be executed.

The following sections describe two confidence estimation methods explored at DaimlerChrysler. We also present our evaluation framework for confidence estimation, the experiments and results, and finally the conclusions which can be drawn.

5.1 Confidence Estimation Methods

This section presents two different methods for confidence estimation. The first method computes confidences directly from a large word graph, whereas the second method is based on a typical two-stage classification framework. That is, features that correlate with the word's correctness are collected and fed into a classifier which estimates posterior probabilities for the word being correct.

5.2 Direct Estimation from Word Graphs

In this method the large hypothesis graph, which results from the first decoding stage and which has been pruned, acts as the search space and is the sole input for confidence estimation. The probability mass lost due to pruning may be neglected if the graph is large enough.

How can confidences be estimated from the word graph only? Wessel et al. (1998) present a method that computes for each edge i in the word graph the probability $\gamma(i)$ that it is passed. The calculation is based on the well-known forward/backward algorithm, where acoustic probabilities replace the emission probabilities and language model probabilities replace the transition probabilities. Given the forward variable $\alpha(i)$ and the backward variable $\beta(i)$, the probability $\gamma(i)$ can be computed as follows:

$$\gamma(i) = \frac{\alpha(i)\beta(i)}{\sum_{j:e(j)=T} \alpha_T(j)} \tag{1}$$

Since the nodes are the connection points between successive edges, any efficient algorithm must consider them. T is the final node in the word graph. The sum in the denominator in Eq. (1) is the total probability that the utterance is contained in the word graph.

Since word graphs usually contain overlapping instances of the same word in competing paths, the probabilities of these instances should be combined in some

way as they represent the same word. This can be achieved by a homogeneity measure HAP(i) (*homogeneity of all paths*), which is calculated on a frame-by-frame basis:

$$\mathrm{HAP}(i) = \frac{\sum_{t=t_a(i)}^{t_e(i)} \sum_{j:\, t_a(j) \le t \le t_e(j)} \gamma(j)\, \delta(w_i, w_j)}{\sum_{t=t_a(i)}^{t_e(i)} \sum_{j:\, t_a(j) \le t \le t_e(j)} \gamma(j)} \tag{2}$$

where $\delta(w_i, w_j)$ is the Kronecker function.

5.3 Feature Collection and Classification

The second method for confidence estimation is based on statistical decision theory (Eide et al., 1995). A set of confidence features m, each correlating with the correctness of the hypothesized word, is mapped into a binary class k with values 0 for "erroneous" and 1 for "correct" word hypothesis. The posterior probability can then be calculated as follows:

$$P(k=1|m) = \frac{P(m,k=1)}{P(m,k=1)+P(m,k=0)} \tag{3}$$

A set of utterances must be excluded from recognizer training in order to train the confidence classifier. The target vector for classification training is computed by a recognition run on the heldout set and an alignment of the word hypothesis on the reference transcription. Correctly recognized words receive the label 1, and recognition errors receive the label 0 in the target vector. This is because errors should ideally lead to a very low posterior probability and correct results to a high probability. Because we used Levenshtein alignment for the experiments in this study, only insertions and substitutions, but not deletions, are covered. A schematic diagram for this two-stage method can be found in Fig. 4.

Which of the many possible features should be used must be decided by a human expert. Features were selected from all recognition stages mainly heuristically. The most important features are listed below:

1. Signal analysis:
   a) SNR: signal-to-noise ratio (Eide et al., 1995)
   b) Fo: fundamental frequency
2. Vector quantization:
   a) VQMaxProb: maximum probability per frame
   b) VQNumClass: number of competing codebook classes
3. Decoding:
   a) NormScore: length-normalized acoustic score (Schaaf and Kemp, 1997)
   b) LMScore: average language model score (Chase, 1997)

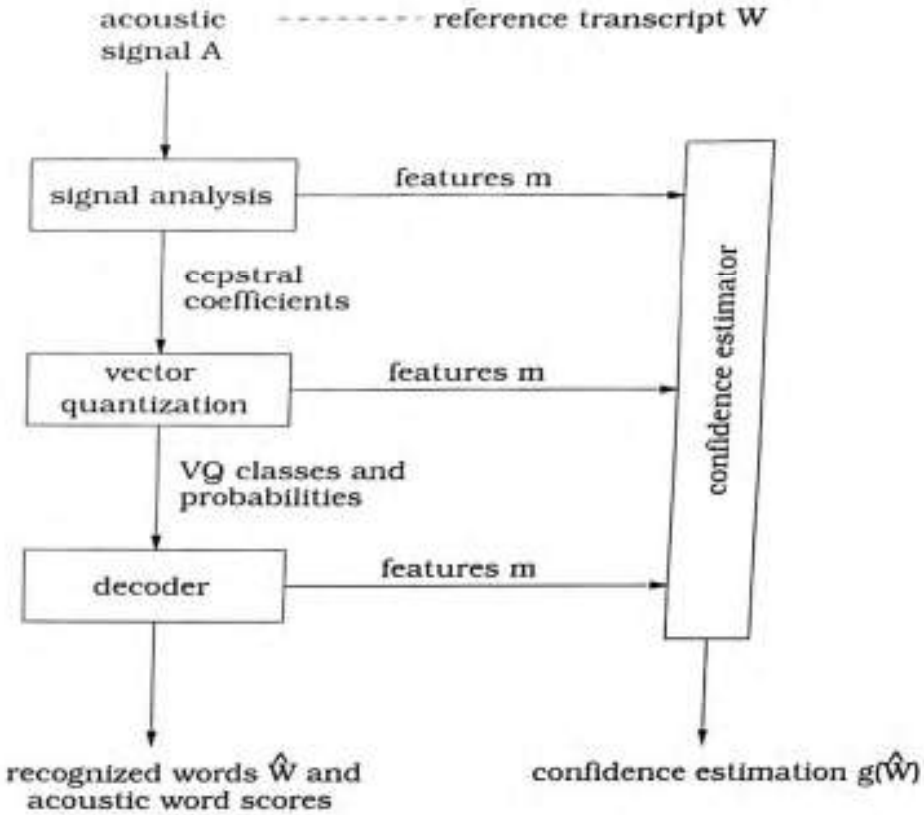

Fig. 4. Schematic diagram of two-stage confidence classification

c) LMScoreBestPath: language model score on the best path
d) LRWordPhonFree: likelihood ratio of scores of word decoding to scores of free phoneme decoding (Young, 1994)
e) LREmiBestAvgFree: likelihood ratio of best and average emission probabilities per frame (for active model states)
f) GraphHomo: graph homogeneity HAP in Sect. 5.2 (a local confusability measure)
g) Astabil: acoustic stability (Finke et al., 1996)
h) ActWordEnds: number of competing word ends
i) SpeechRate: simple speech rate estimation as frames per state
j) AvgLossProbFrm: average probability loss per frame

4. Word-specific properties from recognizer training:
   a) WordCorRate: word-correct rate of each word in the training data (Schaaf and Kemp, 1997)
   b) WordFreqTrain: frequency of the word in the training data (Chase, 1997)

A large number of additional features have also been evaluated, but they are neglected here since they do not improve classification performance.

Many different classification techniques can be employed for confidence estimation. We decided to investigate four different approaches, each with its own particular advantages:

1. Gaussian mixture models (GMM) — easy and fast to train; most features fulfill the normal distribution assumption
2. Polynomial classifiers (PC) — better functional approximation than GMM; practical solution by a system of linear equations
3. Multilayer perceptron (MPL) — complex and time-consuming training procedure; robust against outliers; suggested if no knowledge about feature distributions exists
4. Support vector machine (SVM) — prevents overadaptation by minimizing the empirical risk; focuses on modeling the class boundaries

5.4 Evaluation Metrics

The confidence measures in this study are applied to word rejection. This means a binary output of "accept" or "reject" is sufficient. Since the introduced confidence measures are estimations of posterior probabilities, a threshold operation allows the continuous values to be transformed into binary values. In order to compare the performance of different confidence estimators, we are looking for a scalar measure that represents the relative amount of errors. One such well-known measure is the classification error rate (CER), the ratio of the number of erroneous classifications to the overall number of classifications. Classification errors are rejections of correctly recognized words (type I errors) and acceptances of recognition errors (type II errors).

Type I errors increase with decreasing type II errors, and vice versa. For some applications it makes sense to allow more rejections in order to be very sure about the accepted samples. The optimal operating point for each application can be found by investigating the detection error tradeoff (DET) curves that plots the two types of errors against each other. The DET diagrams also help us gain insight into the performance of the estimators by comparing curves.

5.5 Experimental Setup

A semicontinuous HMM-based speech recognition system (Class et al., 1993) was used for our experiments. This system was described in detail in Sect. 2. For our studies two different recognizers are compared. The first one was trained on the command speech database CARCOMM of DaimlerChrysler, which contains commands spoken in the noisy car environment. The second one was trained on the spontaneous speech database Verbmobil II, which consists of human-to-human travel-planning dialogues. Word transitions were modeled by a restricted command grammar for CARCOMM, and by a statistical language model for Verbmobil. Spectral subtraction (SPS) is used in signal preprocessing for CARCOMM only. Table 4 summarizes some important statistical data of these databases.

Table 4. Training and test databases for word recognition and confidence classification

Database	No. of HMMs	No. of words in train. set	Size of vocabulary	WER (%)
CARCOMM	1500	24,336	406	9.7
Verbmobil II	1500	35,943	10,166	37.6

5.6 Results and Discussion

This section shows the performance of the direct confidence estimation method (HAP). Results are presented in terms of relative reduction in CER over the baseline system. The baseline CER assumes a naive classifier that always chooses the most frequent class, in our case the class "correct". It is basically WER excluding deletion rate.

Table 5 shows results for CARCOMM and Verbmobil II. CER can be significantly decreased (about 30%) for both databases.

Table 5. Results for direct estimation of acoustic confidences (HAP)

Database	No. of words in rec. set	Size of vocabulary	WER (%)	CER %
CARCOMM	8825	406	9.7	30.5
Verbmobil II	10,914	10,166	37.6	30.9

The classifier for the two-stage method is initialized with just the best-performing feature. Then all other features are added one by one in the order of their performance. Tables 6 and 7 show the results using GMM classifiers for command and spontaneous speech respectively.

Table 6. Results for the GMM classifier on command speech

Database	Feature	CER (%)
CARCOMM	WrdCorRate	47.3
	+ GraphHomo	55.0
	+ LMProbBestPath	57.9
	+ Astabil	58.2
	+ ActWrdEnds	60.4

The *WrdCorRate* feature strongly influences the confidence performance in command speech due to the smaller sized vocabulary. *GraphHomo* also plays an important role in both kinds of speech. It proved to be the best feature for spontaneous speech. Some other decoder features reduced CER further, but these features differ for command and spontaneous speech.

The best GMM classifier for command speech performs almost twice as well as the best GMM classifier for spontaneous speech. That can be explained by the

Table 7. Results for the GMM classifier on spontaneous speech

Database	Feature	CER (%)
Verbmobil II	GraphHomo	30.9
	+ WrdCorRate	34.0
	+ SpeechRate	34.1
	+ LREmiBestAvgFree	34.2
	+ ActWrdEnds	34.7

fact that command speech is more restricted, so that the distributions of confidence features are better separated.

The assumption made by the GMM approach that feature distributions can be approximated by a mixture of Gaussians does not always hold. Therefore, we examined some other classification techniques. These techniques lead to further CER reductions (Table 8), particularly for the SVM classifier. Precise modeling of class boundaries and minimizing empirical risk instead of classification error on the training database pays off here.

Table 8. Comparison of classification methods

Database	GMM	PC	MLP	SVM
CARCOMM	60.4	63.0	64.2	66.1
Verbmobil II	34.7	36.7	34.8	36.2

Finally, the estimation methods for both speech databases are compared in a DET curve (Fig. 5). The two-stage method clearly outperforms the direct estimation method for command speech (two lower curves). The performance difference between the two methods is not so significant for spontaneous speech (two upper curves).

6 Modeling Unknown Words

In natural speech input the user is not restricted in the way how to put his or her words. Thus, naturally, a very large vocabulary is required to cover all the utterances a user might enter into the system. Now, at this point, two aspects have to be regarded: first, the recognition performance of a speech recognition system degrades with growing vocabulary size due to decreasing separability. Second, the gain in recognition performance obtained by adding more ever-less-frequent words to the vocabulary will constantly shrink and approach zero. Hence it makes sense to limit the vocabulary size, which is inevitable in real-world applications, anyway.

This leaves us with a certain (unknown) number of words that the recognizer cannot detect. Since these are the least frequent words, we can assume that they will not bear any meaning critical to the application. Thus, in a given utterance, we

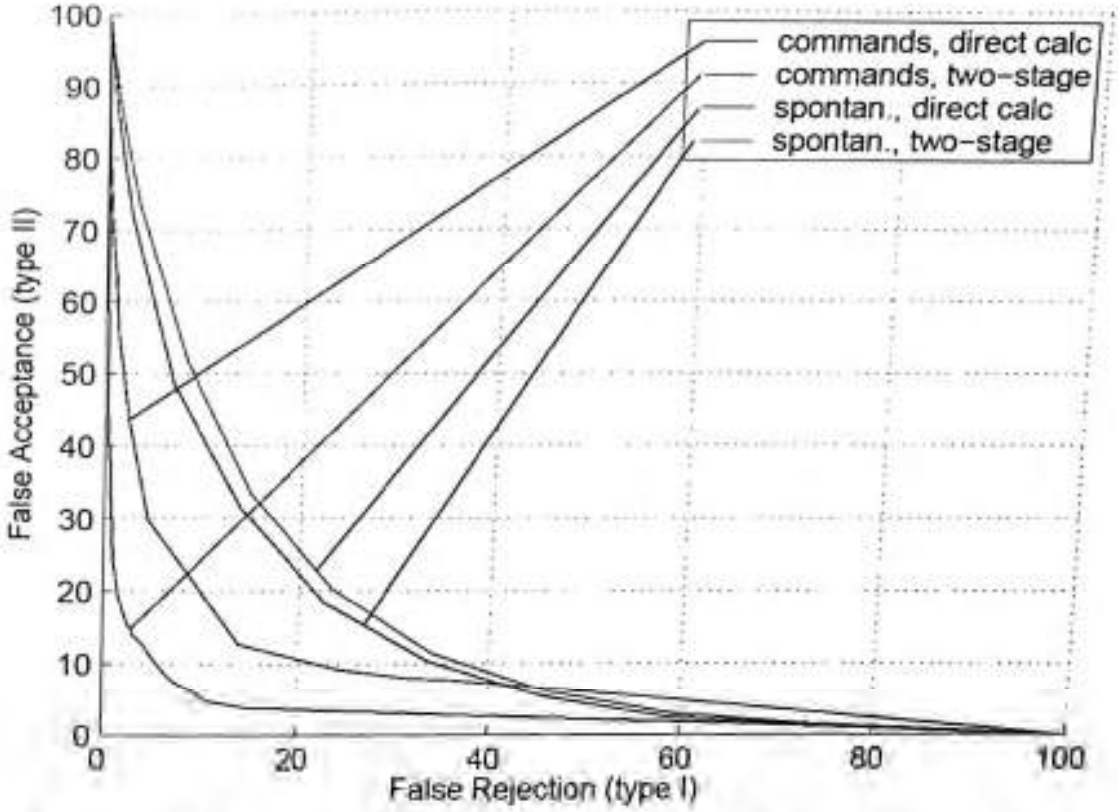

Fig. 5. DET curves for direct estimation and two-stage approaches on command and spontaneous speech

will be happy to ignore these words, but we want the rest of the utterance perfectly preserved. Therefore, the word boundaries need to be preserved, whereas a typical recognizer would do anything, including shifting the word boundaries, in order to match the utterance with the unknown words against words that it does have in its vocabulary.

To fill the holes left by unknown words, acoustic models are needed that match better against the unknown word than a wrong alignment of vocabulary words would, including the fact that a wrong alignment might produce lower scores for wrong surrounding words. Such acoustic out-of-vocabulary (OOV) models have been suggested as one single garbage model (Class et al., 1993) and as a set of 15 length-dependent models (Fetter, 1998). In the case of a single model either all nonapplication words or all words of a dictionary are trained to that model. It is quite obvious that a single model cannot be a reasonable acoustic match to many acoustically different words. Therefore, Fetter divided the training vocabulary into 15 sets of words by number of phonemes (SAMPA characters) and trained them to HMMs with 15 topologies, respectively.

In the present approach, a further step is made toward capturing the acoustic properties of unknown words more closely. Every training word is regarded as consisting of syllables. Every syllable in turn consists of a number of start consonants,

a core of vowels, and a number of end consonants. A certain number of phoneme classes are introduced for each, the start consonants, vowels, and end consonants, respectively. To obtain these classes, gradually phonemes with the greatest similarity (see Sect. 2.2) are merged together to form a class. Each class is trained to a separate HMM. In the experiments, five sets of phoneme classes with different stages of granularity are regarded (see Tabel 9).

Table 9. OOV sets/HMMs and lexicon entries

Set No.	Start cons. classes	Vowel classes	End cons. classes	No. of HMMs	Lexicon entries
1	1	1	1	3	33
2	2	1	2	5	399
3	2	2	2	6	567
4	2	2	3	7	4509
5	3	2	3	8	4936

For each set, the phonemes of each training word (60,000 German city names) are mapped to the generalized phoneme classes, and the resulting different generalized words are introduced into the recognizer vocabulary as OOV words. The 33 lexicon entries for set 1 roughly correspond to Fetter's length-dependent models.

In the test setup, a statistical trigram language model with backoff is used, where all OOV words were in a single garbage word class. For the test purpose, the OOV word rate in the test material was artificially raised to 23.2% by choosing the 1000 sentences of the LM training material that contained the most unknown words. In a first recognition run the overall performance was evaluated (see Tabel 10). After the first run, the timestamps of the recognizer output were used to cut out the signal parts that were marked OOV. These signal parts should bear the unknown words that were really uttered. In a second recognition run these signal portions were recognized with a dictionary that contained those words instead of the OOVs with a single word recognizer. This second run measures how well the word boundaries are preserved by the OOV models.

Table 10. OOV/recognition results

	1st Run			2nd Run
Set No.	WER (%)	RCL (%)	PRC (%)	WER (%)
1	33.9	86.1	78.2	83.6
2	31.0	89.7	76.3	77.3
3	28.9	92.7	76.3	69.7
4	18.8	97.1	80.4	40.0
5	18.7	97.0	80.9	39.2

The recognition results clearly show a great increase in performance regarding word error rate (WER), OOV recall rate (RCL) and OOV precision (PRC). All figures of the first run improve significantly until set 4. Sets 4 and 5 are almost equal. Thus, set 4 is proposed for use in the application. The second run, which tested if the signal portions marked OOV by the first run really contain the uttered word, shows dramatic improvement by increasing the number of OOV words. This means that the preservation of word boundaries could also be greatly improved. Again, in this test, sets 4 and 5 are very close.

In a natural speech input environment the recognizer dictionary can never hold all words that may possibly be uttered. Acoustic OOV models are a means to fill in the holes left by unknown words. Literature shows that length-dependent models outperform the single model approach. Experiments prove that we were able to improve the system performance substantially by replacing the length-dependent approach by a greater number of generalized word models.

6.1 Conclusion

Significant improvements in speech recognition have been achieved in all four main research areas investigated in SMARTKOM. The research demonstrators for the different SMARTKOM scenarios are able to cope with natural, spontaneous language input, speaker independence, large vocabularies, and background noises. A speech recognizer based on semi continuous Hidden Markov Models and statistical language models was trained for the SMARTKOM application on a large set of training data.

The system adapts itself to the environment by enabling Barge-In, reducing environmental noises, and adjusting the input signal level using automatic gain control. Barge-In allows for user input while the system speaks to the user. This functionality is enabled by an echo cancellation technique integrated in the acoustic device. Several noise reduction and noise modeling methods were investigated, but only spectral substraction proved to be useful in the SMARTKOM applications.

A number of speaker adaptation methods have been presented in this contribution, each improving recognition accuracy by approximately 10% to 20% relative. The SMARTKOM system incorporates several unsupervised speaker adaptation techniques: Vocal tract normalization is mandatory, while maximum likelihood linear regression and codebook mixture can be used optionally.

The SMARTKOM system adapts itself to the task or application by dynamically updating the vocabulary and the members of the categories in the statistical language model. Vocabulary sizes range from 3000 words to 12,000 words, depending on the application. Typical language model categories are city names, street names, movie titles, actor names, genres, numbers, etc. Adaptation to the task has a major influence on recognition accuracy.

Two methods for word confidence estimation were introduced. The first method calculates word confidences from very large word graphs directly, while the second method collects features that correlate with the word's correctness and estimates confidences using known classification techniques. The direct calculation and the two-stage methods both lead to significant improvement of confidence estimations. A

36.7% relative reduction in confidence error rate is achieved for spontaneous speech, and a 66.1% reduction is obtained for command speech. Confidence estimation is more reliable on command speech, since vocabulary size and perplexity are lower than in spontaneous speech. The feature-based two-stage approach is superior to the direct calculation method, but also requires more computation time and memory.

For SMARTKOM, however, two-stage calculation has some drawbacks:

1. Two different recognition algorithms (for German and English, resp.). Hence a *feature-based* confidence classifier strongly depends on specific algorithms implemented.
2. Any changes in the recognizer setup (e.g., signal analysis, vector quantization, HMM modeling, vocabulary size, and even parameters to speed up computation) may degrade classifier performance. In order to overcome these disadvantages of two-stage calculation, classifiers should be strictly adapted to a given setup.

Considering these drawbacks, we decided for direct estimation of confidences. This is because we have a well-defined interface between the Viterbi decoding module and the confidence estimation module: *the word hypotheses graph*.

Finally, we investigated how to model words that are not covered by the vocabulary of the speech recognizer. Using an approach of generalized word models we were able to detect a large number of unknown words and improve the system performance substantially.

References

L.L. Chase. *Error-Responsive Feedback Mechanisms for Speech Recognizers*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

F. Class, A. Kaltenmeier, and P. Regel-Brietzmann. Optimization of an HMM-based Continuous Speech Recognizer. In: *Proc. EUROSPEECH-93*, pp. 803–806, Berlin, Germany, 1993.

E. Eide, H. Gish, P. Jeanrenaud, and A. Mielke. Understanding and Improving Speech Recognition Performance Through the Use of Diagnostic Tools. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-95)*, pp. 221–224, Detroit, MI, 1995.

P. Fetter. *Detection and Transcription of Out-Of-Vocabulary Words in Continuous Speech Recognition*. PhD thesis, Technical University of Berlin, 1998.

P. Fetter, F. Dandurand, and P. Regel-Brietzmann. Word Graph Rescoring Using Confidence Measures. In: *Proc. ICSLP-96*, pp. 10–13, Philadelphia, PA, 1996.

M. Finke, T. Zeppenfeld, M. Maier, L. Mayfield, K. Ries, P. Zhan, and A. Waibel. Switchboard April 1996. Technical report, DARPA, 1996.

F. Metze, T. Kemp, T. Schultz, and H. Soltau. Confidence Measure Based Language Identification. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-2000)*, Istanbul, Turkey, 2000.

M. Pitz, F. Wessel, and H. Ney. Improved MLLR Speaker Adaptation Using Confidence Measures for Conversational Speech Recognition. In: *Proc. ICSLP-2000*, Beijing, China, 2000.

T. Schaaf and T. Kemp. Confidence Measure for Spontaneous Speech Recognition. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-97)*, pp. 875–878, Munich, Germany, 1997.

M. Weintraub. LVCSR Log-Likelihood Ratio Scoring for Keyword Spotting. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-95)*, pp. 887–890, Detroit, MI, 1995.

F. Wessel, K. Macherey, and R. Schlüter. Using Word Probabilities as Confidence Measures. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-98)*, pp. 225–228, Budapest, Hungary, 1998.

G. Williams and S. Renals. Confidence Measures for Hybrid HMM/ANN Speech Recognition. In: *Proc. EUROSPEECH-97*, pp. 1955–1958, Rhodes, Greece, 1997.

S. Young. Detection of Misrecognitions and Out-Of-Vocabulary Words. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-94)*, pp. 21–24, Adelaide, Australia, 1994.

Class-Based Language Model Adaptation

Martin C. Emele, Zica Valsan, Yin Hay Lam, and Silke Goronzy

Sony International (Europe] GmbH
Sony Corporate Laboratories Europe, Advanced Software Laboratory, Stuttgart, Germany
{emele,valsan,lam}@sony.de, silke@goronzy-privat.net

Summary. In this paper we introduce and evaluate two class-based language model adaptation techniques for adapting general n-gram–based background language models to a specific spoken dialogue task. The required background language models are derived from available newspaper corpora and Internet newsgroup collections. We followed a standard mixture-based approach for language model adaptation by generating several clusters of topic-specific language models and combined them into a specific target language model using different weights depending on the chosen application domain. In addition, we developed a novel word n-gram pruning technique for domain adaptation and proposed a new approach for thematic text clustering. This method relies on a new discriminative n-gram–based key term selection process for document clustering. These key terms are then used to automatically cluster the whole document collection. By selecting only relevant text clusters for language model training, we addressed the problem of generating task-specific language models. Different key term selection methods are investigated using perplexity as the evaluation measure. Automatically computed clusters are compared with manually labeled genre clusters, and the results provide a significant performance improvement depending on the chosen key term selection method.

1 Introduction

Statistical language modeling has been successfully applied to many domains such as automatic speech recognition (ASR (Jelinek, 1997; Niesler and Woodland, 1997), spoken language understanding (Wang, 2001), information retrieval (Miller et al., 1999) and handwriting recognition (Srihari and Baltus, 1992)). In particular, trigram language models (LMs) have been demonstrated to be highly effective for these domains. The need to go beyond trigram models is advocated in Jelinek and Lafferty (1991). Long-distance models, which try to capture statistical dependencies that go beyond trigrams, include models such as caching (Clarkson and Robinson, 1997; Kuhn and De Mori, 1990), higher order n-grams (Goodman, 2001), sentence mixture (Iyer and Ostendorf, 1999), and class-based models (Kneser and Ney, 1993; Bellegarda et al., 1996), etc.

An important issue for any ASR system is the adaptivity to a specific domain or task for which it is intended. Models used in such systems should be specific to the

style of language used in a particular domain. As such, when confronted with utterances from another domain, performance will degrade substantially. Therefore, the adaptation of the language model to new domains is required. Techniques such as caching or mixture-based language models are commonly used to tackle this problem. For the latter approach, which we have followed here, one needs to generate several LMs specific to a particular topic or style of language (Iyer and Ostendorf, 1999; Dempster et al., 1977; Seymore and Rosenfeld, 1997).

Language modeling attempts to capture, represent, and exploit the regularities and characteristics of natural language. In large-vocabulary continuous speech recognition (LVCSR), the language model provides a prior estimation of the probability of a spoken word sequence. It helps to disambiguate acoustically ambiguous utterances. It also facilitates the reduction of the search space to word sequences that occur more frequently in natural language. State-of-the-art statistical n-gram modeling (Bahl et al., 1983) models the frequency of the occurrence of word sequences. It assumes that the occurrence of a word depends only on the preceding n-1 words. Neither words with a distance more than n nor the semantic contexts are taken into consideration. The probability of a word sequence $W = w_1, w_2, w_3, \ldots, w_N$ is estimated as:

$$P(W) = \prod_{i=1}^{N} P(w_i | w_{i-n+1}, \ldots, w_{i-1}). \tag{1}$$

All the probability components on the right-hand side of Eq. (1) are usually estimated by taking statistics on a large collection of written texts such as newspaper articles.

In application-specific spoken dialogue systems, in order to recognize spontaneous task-specific utterances correctly, domain-specific language models, which are able to capture spontaneous speech phenomena, are essential for the recognition engine. Nevertheless, n-gram language models trained on newspaper articles do not really reflect the characteristics of the spontaneous speech as used in real dialogue systems. Therefore labor- and cost-intensive Wizard-of-Oz (WOZ) experiments (Schiel and Türk, 2006) have to be conducted to collect tas-specific dialogue data. Compared to the size of written texts, the manually collected data is usually small and hence is not a reliable basis for constructing adequate language models. On the other hand, Internet resources such as newsgroup posts, mails etc. provide abundant amount of texts with characteristics and styles similar to spontaneous speech.

In Sect. 2, we first describe a newsgroup-based text corpus development and preparation system, which automatically collects newsgroups, filters junk material, and converts them to a high-quality general corpus that can be used for language model training. In Sect. 3, we describe the generalization of standard n-gram LMs to class-based LMs where the contents of classes can be dynamically changed at runtime. In Sect. 4, we introduce two complementary language model adaptation techniques, which adapt the general corpora to application- and task-specific language models. In Sect. 5, we discuss the clustering performance of three different term selection methods for the thematic text-clustering approach and evaluate the adapted LM in terms of word error rates using different test sets from the SMARTKOM multimodal dialogue collections. Finally, we present our conclusions in Sect. 6.

2 Copora Preparation

Since we are dealing with German, we had to collect and prepare suitable training corpora for the language model training. Besides using newspaper text from the Süddeutsche Zeitung (SZ), we set up an automatic collection of domain-specific texts from newsgroups and collected dialogue turns which were recorded during the WOZ experiments. Since the corpora are not directly in a format that can be used for language model training, a set of filtering and formatting tools was developed. The module consists of two main components:

1. **Removal of junk content.** For the corpora, the filtering script removes the junk material, e.g., the binary streams such as audio and video files in the newsgroups. Then, it also removes all non-German news articles by using a simple language guesser based on a wide-coverage dictionary for German, i.e., CELEX (Baayen et al., 1993), containing 359,611 words. It also removes nonalphabetic words, like emoticons etc.
2. **Text reformatting.** This component does tokenization and detection of sentence boundaries by trying to disambiguate interpunctuations, like period, whether it is part of an abbreviation or whether it marks the end of a sentence. Since many news articles make use of a special abbreviation language to speed up communication, e.g., "cu" expands to "see you," etc., we try to expand those abbreviations by using a dictionary of common abbreviations. Next, we try to clean up the text by normalizing German umlauts from ASCII notation into their corresponding ISO 8859-1 encoding scheme. Finally, we run a simple spell checker across the text by checking the spelling of words against the CELEX dictionary.

After the preprocessing phase of the SMARTKOM corpora, we yield 3,297,970 sentences for training. The SZ corpus mainly serves as the background corpus to capture the general characteristics of German utterances. The dialogue turns that partly result from the WOZ experiments and partly from manual construction based on the analysis of the particular tasks are supposed to reflect the specific scenarios for which SMARTKOM is intended. The collection of newsgroup texts in addition to the dialogue data is necessary, since the amount of dialogue-specific data is not sufficient to construct a reliable LM. Furthermore the nature of writing in newsgroups is somehow closer to spontaneous speech. All these corpora are used as the basis for the language model adaptation techniques, which are described in Sect. 4.

3 Class-Based Language Models

The LM used in this project is a statistical, class-based, 3-gram LM. The LM is tightly coupled to the lexicon (Gorony et al., 2006) and also has to reflect the dynamic nature of the vocabulary. To be able to do so, frequently changing words, such as actor names, movie titles, streets in different cities, etc., are categorized and put into word classes. The 3-grams are then based on these classes, rather than on the

words themselves. The advantage of the class-based LM is that it is no longer necessary to have all words in all contexts in the training data. For all words in one class the histories of all the single words in that class can automatically be associated with all words in that class. For example, consider two sample sentences: "I liked James Bond very much," and "I hated Star Trek". Using the class MOVIETITLE the sentences will be mapped to "I liked MOVIETITLE very much," and "I hated MOVIETITLE". The resulting LM then covers the original sentences as well as all possible combinations, e.g., "I liked Star Trek very much" etc., since both "Star Trek" and "James Bond" are in the MOVIETITLE class.

In SMARTKOM the content of the word classes can dynamically change. New words can be added/removed to/from a class. We just need to recompute the in-class probability for all words in the classes, which is trivial if all words in a class are equally weighted. However, it would be possible to increase the weight of those words in a class that are currently displayed on the screen.This would account for the fact, that if, e.g., a list of 20 movie titles is the result of a user request and only 10 can be displayed on the screen (to view the others you would have to scroll down), then the user is more likely to say one of those titles that he can currently see on the screen. Therefore their probability could be increased.

By updating the contents of the classes, we can achieve the adaptation of the LM to a specific domain to a certain extent, without the need to completely change and reload the LM as is done by many other dialogue systems, e.g., those based on VoiceXML. Such an approach is very time-consuming and often not feasible in dialogue systems of this coverage. Furthermore, by not restricting the LM to subparts for certain kinds of expected dialogue turns (which is an often used strategy), the capability of being able to speak words from all subdomains, no matter if they are within the range of expected utterances at this stage of the dialogue, remains.

Currently, 31 word classes are defined. However, only the following classes will be dynamically updated, because their content changes frequently depending on the current dialogue state: ACTOR, CITY, GARAGE, GENRE, HOTEL, LOCATION, MOVIETHEATER, and MOVIETITLE.

4 Language Model Adaptation

Within the SMARTKOM project, two different approaches are adopted to tackle the problem of LM adaptation: language model adaptation using word *n*-gram pruning (Lam and Emele, 2001) and language model adaptation using thematic text clustering (Valsan and Emele, 2003).

4.1 LM Adaptation Using Word *N*-Gram Pruning

The vocabulary size of the background corpora, i.e., SZ and newsgroups, is large. On the other hand, for spoken dialogue systems, it is a common practice to define the vocabulary beforehand based on the underlying tasks, e.g., as those in the SMARTKOM

system. Such a task-specific system usually employs a small or medium size vocabulary. Therefore, the probability distribution of n-grams generated from such general corpora usually does not reflect those that occur in the spoken dialogue system due to the presence of large amount of out-of-vocabulary (OOV) words. Therefore, in the n-gram pruning approach described below, n-grams with OOV words are pruned.

N-gram pruning using count cutoff has been used in the literature to reduce the size of language models. Any word n-grams with counts below a predefined threshold are pruned. Instead of using such a count cutoff, a novel word n-gram pruning technique based on OOVs is introduced to adapt a language model on the n-gram statistical level. It exploits the basic assumption of n-gram language models that the probability of a given word in a word sequence is only influenced by its n-1 preceding words. In word n-gram pruning, all word n-gram frequency counts are computed. Then, any word n-grams that contain OOVs are pruned. N-gram probabilities are directly estimated from the pruned word n-gram frequency counts. As OOVs are completely eliminated in the pruning step, the probability of unknowns or the likelihood of seeing an OOV in the test set can be set to values that are appropriate for the application if an open vocabulary is required. This approach also allows rapid adaptation to new vocabularies as the new vocabulary is only required during the pruning stage and recounting of word n-grams from the corpora is not necessary. The pruning step can also be interpreted as a corpus adaptation technique in which the adapted corpora are formed of sentence segments, which consist of at least n continuous non-OOVs. In addition to word n-gram pruning, which is an adaptation on the level of n-grams, we developed another adaptation technique on the level of selecting only relevant documents from a general background corpus by applying an automatic text-clustering approach.

4.2 Language Model Adaptation Using Thematic Text Clustering

In general, an application domain can be characterized by the subtasks that comprise the overall dialogue flow. Each task can be characterized by a topic or a set of topics to which the sample dialogues belong. Similarly, a given document collection like newspaper texts can be categorized into specific text clusters according to a given set of topics. Usually, the newspaper/newsgroup articles are already manually classified into different genres like news, sports, movies, etc. Together with the title of the article or words that appear in the beginning of the documents, this input can be used to derive the main topic of the documents. Based on this information, topic-specific text clusters can be derived and further, for each cluster, a topic-specific LM can be generated. The probability estimates from these component LMs are interpolated to produce an overall probability, where the interpolation weights are chosen to reflect the topic or style of language currently being recognized.

But, in general, such a naive approach, which uses only manually assigned topic labels and limited context information like title words and abstracts, is not sufficient to cover the dynamic topic shifts that occur even within one single document. The result might be a false labeling of documents, leading finally to a cluster that covers

many different topics. Hence, the topic-generated LM is actually a more general LM rather than a specific one.

Therefore, in SMARTKOM, we proposed another automatic text-clustering approach for topic assignment based on three different key term selections that are used to define the topics of the documents. These three term selections are compared with the naive baseline reference, which rely on manually clustered documents according to the genre information.

The process of term selection is based on standard information retrieval (IR) techniques (Rasmussen, 1992), but in our approach the importance of a term is evaluated by averaging over the whole document collection. The accuracy of the classifier is therefore subject to the selection of key terms. The semantic classification of the document context is done by using a combination of standard techniques like text preprocessing, random mapping projection (RMP, see Kohonen et al. (2000)) to reduce the size of the term vectors, and vector quantization (VQ) techniques like the split LBG algorithm (Linde et al., 1980) for identifying the clusters in the reduced vector space.

4.2.1 Term Selection and Document Representation

Text-clustering algorithms partition a set of documents into groups or clusters. In this respect a document has to be transformed into a representation which is suitable for the classification task. We used the vector space model (VSM) approach (Salton and McGill, 1983) for coding the documents, where the documents are represented as an m-dimensional stream of indexing *terms*. In general, terms can be n-grams, instead of simple words ($n > 1$):

$$d = (w^{(1)}_{n-gram}, w^{(2)}_{n-gram}, w^{(3)}_{n-gram}, \ldots, w^{(m)}_{n-gram}).$$

Defining and selecting the most representative terms in a corpus using this technique is based on the assumption that the terms in a document are conditionally independent of each other (Robertson and Spärck, 1976). Due to this independence assumption some information about the document is lost. Additional information about the context in terms of n-grams ($n > 1$) can be an effective way to alleviate the limitation of the simple bag-of-words model. On the other hand, using, for instance, all bigrams or trigrams, the size of the coded document is increased now to around m^2 or to around m^3, respectively. Therefore, in order to not increase the dimension of the term-category matrix too much, one can apply the term selection process only to those n-grams where **all** n-constituent words belong to a predefined vocabulary. This selection is efficient for most practical dialogue systems, where a fixed medium-sized vocabulary is already defined by the domain. The advantage of using such an approach is twofold: First, it decreases the computational cost by discarding all irrelevant documents (those whose corresponding n-grams are missing). Second, it improves the classifier performances by providing clusters, which cover documents with the same or at least related topics.

After removing words that occur in the stop word list, and after applying a standard stemming algorithm, the most relevant terms in the collection need to be se-

lected. This is done by defining a *term-weighting* function which supplies a weight for each term (w_i) given a document d_j in the collection.

Since we are interested in defining the terms that are the most representative for the entire collection and not in a particular document (as is the case for standard IR systems) the formula below was used:

$$weight(w_{n-gram}) = \left(\frac{1}{N_{n-gram}} \sum_{k=1}^{N_{n-gram}} tf_{n-gram,k} \right) \cdot \log \frac{N}{N_{n-gram}} \tag{2}$$

where each term is an n-gram where $n \geq 1$, N_{n-gram} is the number of documents that contain the term w_{n-gram}; $tf_{n-gram,k}$ is the number of concurrencies of term w_{n-gram} in the document k.

Thus, the n-gram terms weights are computed according to their "average" importance in the corpus. To select a subset of m terms in all documents, which represent the key terms, the m n-grams with the highest weight exceeding an empirical threshold are chosen. The value of the vector elements $w^{(i)}_{n-gram}$ for a document is a function of the frequency counts of the n-grams in the corresponding document. The documents are coded based on these values, represented by vectors in the vector space, and then normalized to the unit vector. In this way the direction of the vector will reflect the content of the document.

4.2.2 Methods for Dimensionality Reduction

The main problem of the vector-space model is its large vocabulary for any sizable collection of free-text documents. Such term count vectors are typically very sparse and very high dimensional; therefore it is almost impossible to use them as such. A computationally simple method of dimensionality reduction that does not introduce a significant distortion in the data is thus desirable. Random Mapping Projection is a solution that provides a computationally feasible method for reducing the dimensionality of the data so that the mutual similarities between the data vectors are approximately preserved (Kaski, 1998).

4.2.3 Estimation and LM generation

Once the documents have been encoded and mapped to a lower dimension, so that similarity of the texts is reflected in the similarity of the respective document vectors, the remaining task is to create text clusters out of these documents.

We resort to a VQ technique such as split LBG (Linde et al., 1980), which uses for comparison purposes 32 dimensions (classes) for the code vectors, exactly the same number as the manually generated topics in the corpus. In the next section we evaluate various term selection variants in terms of perplexity and present the word error evaluation results using the optimised LM.

5 Experimental Results

In this work 100,117 documents were used for training. The whole document collection was manually categorized into 32 clusters, using genre information, with each cluster representing a single topic. This categorization serves only as a reference for evaluating the automatic clustering algorithm described above. Furthermore, we will refer to these clusters as genre-clusters. For each cluster a 3-gram–based LM was generated using the CMU SLM toolkit (Rosenfeld and Clarkson, 1997).

5.1 Comparative Study

Three sets of comparative experiments for the key term selection process were performed, which are defined as follows:

1. **Word Frequency Counts** (WFC): Select as key terms in the corpus all words with counts higher than a predefined threshold (as one of the standard procedures).
2. **Word Score Counts** (WSC): Select as key terms in the corpus all words whose weight computed by Eq. (2) exceeds a predefined threshold.
3. **Bigram Score Counts** (BSC): Select as key terms in the corpus all bigrams whose weight computed by Eq. (2) exceeds a predefined threshold.

Some preprocessing steps are required for each of the three above approaches in order to find a "good" set of features. A combination of several methods is used as follows: All nontextual information, all numbers, and all words in the stop list are removed; words are converted to their base form using a stemmer; and finally, infrequent and high-frequency words are pruned (see Valsan and Emele (2003) for details).

The resulting set of features defines the vector of key terms, and all documents in the corpus are coded according to this vector. Because the dimensionality of this document vector is huge (around 180,000 for the first two approaches and 450,000 for the third one), we used the RMP method to reduce the size of each of the document vectors to a reasonable value. The mapped size must satisfy two contradictory requests. On one hand, it must be large enough to not distort the mutual similarities of the data vectors too much, and on the other hand it must be small enough to be feasible by the classifier. Therefore, we have chosen a value of 100, which is a good trade-off between precision and feasibility.

The reduced documents matrix was used as input data for the split LBG algorithm. The number of codevectors was 32, the same as the number of manually derived clusters for the corpus. After applying the split LBG algorithm, each document was assigned to a codevector. All documents corresponding to the same codevector were grouped together and define a text cluster. There are 32 such clusters, which in the following will be referred to as smart-clusters. The next section describes the evaluation of the clustering performance based on perplexity results.

5.1.1 Clustering Evaluation Based on Perplexity Results

Several experiments were conducted to investigate how well the automatic clustering algorithm picked out particular topics. Therefore, we have evaluated the performances of the manual clustering against the automatic clustering for all three approaches. Two steps are required for this evaluation:

Step A: manually clustering. Thirty-two Genre-clusters were produced out of all training documents in the corpus using the a priori labeling. Accordingly, 32 LMs were generated for each topic, and we refer to them as LM_{G-i}, i=1,...,32. By using some additional test documents from the same corpus, which have been correctly assigned to their topics, another 32 test sets were produced (each test set corresponds to a topic; the topics are the same as those used for genre-clusters). Perplexity of each of the LM_{G-i} was evaluated on each of these test sets. Each time when a LM_{G-i} computes the smallest perplexity value on the corresponding test set i, we assumed that the cluster contains documents related to the topic of the test set i. In this case, we counted the manually assigned topic label as a correct one and we called such a cluster a topic-winner.

Step B: automatic clustering. One LM was also generated for each of the 32 smart-clusters in order to evaluate the separability between topics and at the same time to give the smart-clusters a proper label. For convenience we refer to them as LM_{S-i}. Perplexities of all LM_{S-i} were evaluated on the same 32 test sets as used in step A. The outcome is a matrix of 32×32 perplexity values. For each test set, whenever any of the 32 LM_{S-i}, which covers an unknown topic, outputs a perplexity value smaller than the smallest perplexity value corresponding to a LM_{G-i}, i=1,...,32, the smart-cluster is considered as a smart-cluster winner. In this way we considered the smart-cluster winner as covering the topic of the test set and it was labeled accordingly. The results for all experiments are summarized in Table 1.

Table 1. Clustering performance evaluation

Clustering approach	Manual	WFC	WSC	BSC
No. of winner clusters/32 (%)	12/32 (37.5%)	19/32 (59.37%)	26/32 (81.25%)	29/32 (90.62%)
Average perplexity	70	55	38	25

The results proved that the automatic text clustering based on any of these approaches outperforms the manual clustering. In case of the WFC-based term selection, the discriminative power of terms is measured by their number of occurrences. The clustering performance is about 59.37%. Using instead the proposed new selection criterion according to Eq. (2) and/or additional context information by using bigrams as terms, a better separability of different topic areas is achieved. In case of WSC a clustering performance of 81.25% is achieved and, in case of BSC the clustering performance is 90.62%, leading to an improvement of 31.25% compared to the standard WFC baseline approach. Because the ultimate goal of text clustering is language model adaptation, we found perplexity to be a more appropriate metric

to measure the performance of the classifier for clustering the documents. Moreover, we have used perplexity to assign a label to each cluster, whenever we considered it as being representative for a certain topic. A huge improvement in term of perplexity is achieved whenever any of three methods (WFC, WSC, and BSC) is used to provide text clusters instead of manually clustering.

5.1.2 Language Model Evaluation using WER

We also evaluated the system in terms of word error rate (WER) using the SMARTKOM recognizer that was developed by DaimlerChrysler (see Berton et al., this volume). We did so by using different speech corpora that were recorded by different partners. The results are listed in Table 2. "Pub" again reflects the public scenario (recorded by DaimlerChrysler) and "MobI" are dialogues from the mobile scenario (car navigation, recorded by DaimlerChrysler). "MobII" are also dialogues from the mobile scenario (pedestrian navigation, recorded by EML); "MTI" are dialogues for the home scenario that were recorded during the MTI exhibition in October 2001; SKI and SKII were dialogues recorded in the SMARTKOM WOZ recordings.

The results for the WOZ recordings are rather bad. The reason is that these recordings include huge amounts of off-talk that was completely out of domain (e.g., users starting to read movie descriptions that are displayed in the home scenario when the task was to select a movie, etc.). For the more scenario-specific dialogues, which did not include that much off-talk, the results are much better, ranging from 8.8% WER to 29.9%. Remember that these experiments included the dynamic update of the word classes and the automatic generation of pronunciations for these words. Please also note that SMARTKOM is a multimodal system. Thus by combining speech with gestures a lot of sentences can be interpreted correctly, even if the speech recognition rate is not perfect.

Table 2. Speech recognition evaluation using different corpora

Corpus	Pub	MobI	MobII	MTI	SKI	SKII
No. speakers	11	11	29	2	3	50
No. sent	100	100	1507	23	26	1234
Perplexity	7.2	14.6	12.6	16.7	19.7	42.2
WER	10.7	8.8	27.6	25.5	29.9	65.4

6 Conclusion

In this paper, we have shown that newspaper and Internet resources, such as newsgroups, are a valuable resource for language model adaptation. Based on these background corpora we have compared and evaluated two LM adaptation techniques and applied them to dialogue-specific language model adaptation. Besides developing a

novel word n-gram pruning for domain adaptation, we proposed a new approach for thematic text clustering. This method relies on a new discriminative n-gram–based term selection process, which reduces the influence of the corpus inhomogeneity, and outputs only semantically focused n-grams as being the most representative key terms in the corpus. These key terms are then used to automatically cluster the whole document collection. By selecting only relevant text clusters for LM training, we addressed the problem of generating task-specific LMs. Different key term selection methods are investigated using perplexity as the evaluation measure. Automatically computed clusters are compared with manually labeled genre clusters. Compared to the manual clustering, a significant performance improvement between 21.87% and 53.12% is observed, depending on the chosen key term selection method.

We also described the class-based language model that allows dynamic updates of classes during runtime, depending on the current dialogue state. Furthermore, we evaluated the adapted language model within the SMARTKOM dialogue system in terms of word error rates using the SMARTKOM recognizer developed by DaimlerChrysler. For the scenario-specific dialogues the word error rates ranged from 8.8% to 29.9%. Compared to the much more restricted first version of the dialogue system, the results using the adapted class-based LMs show a big improvement because the final system demonstrator covers a huge variety of natural interaction dialogues ranging from device control, movie selection, document sending, and route planning to seat reservations, etc.

Acknowledgments

We would like to thank Ralf Kompe for the helpful discussions at the early stage of the project. Many thanks also to Philipp Cimiano and Marion Freese, who worked on the tools for the corpus preparations and helped in evaluating the language models. We would also like to thank our SMARTKOM colleagues from DaimlerChrysler, Andre Berton, Alfred Kaltenmeier, Klaus Mecklenburg, and Peter Regel-Brietzmann, for the very good cooperation and for providing us test data and relevant parameter tunings for their speech recognizer.

References

H. Baayen, R. Piepenbrock, and H. van Rijn. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, 1993.

L.R. Bahl, F. Jelinek, and R.L. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 5(2), 1983.

J.R. Bellegarda, J. Butzberger, W. Chow, N. Coccarao, and D. Naik. A Novel Word Clustering Algorithm Based on Latent Semantic Analysis. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-96)*, vol. 1, pp. 172–175, Atlanta, GA, 1996.

P.R. Clarkson and A.J. Robinson. Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-97)*, Munich, Germany, 1997.

A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood From Incomplete Data Using the EM Algorithm. In: *Annals of the Royal Statistical Society*, vol. 39, pp. 1–38, London, UK, 1977. Royal Statistical Society.

T. Goodman. A Bit of Progress in Language Modeling. *Computer Speech and Language*, 15(403-434), 2001.

S. Goronzy, S. Rapp, and M. Emele. The Dynamic Lexicon, 2006. In this volume.

R. Iyer and M. Ostendorf. Modeling Long Distance Dependence in Language: Topic Mixtures Versus Dynamic Cache Models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39, January 1999.

F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, 1997.

F. Jelinek and D. Lafferty. Computation of the Probability of Initial Substring Generation by Stochastic Context Free Grammars. *Computational Linguistics*, 17(3): 315–323, 1991.

S. Kaski. Dimensionality Reduction by Random Mapping Fast Similarity Computation for Clustering. In: *Proc. IJCNN'98*, vol. 1, pp. 413–418, Piscataway, NJ, 1998.

R. Kneser and H. Ney. Improved Clustering Techniques for Class-Based Statistical Language Modeling. In: *Proc. EUROSPEECH-93*, pp. 973–976, Berlin, Germany, 1993.

T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, 2000.

R. Kuhn and R. De Mori. A Cache-Based Natural Language Model for Speech Reproduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(6):570–583, 1990.

Y.H. Lam and M.C. Emele. Application-Specific Language Model Adaptation Using Internet Resources. In: *Proc. 11th Sony Research Forum '01*, 2001.

Y. Linde, A. Buzo, and R.M. Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, pp. 702–710, 1980.

D.R.H. Miller, T. Leek, and R.M. Schwartz. A Hidden Markov Model Information Retrieval System. In: *Proc. 22nd Int. Conf. on Research and Development in Information Retrieval*, pp. 214–221, Berkley, CA, 1999.

T.R. Niesler and P.C. Woodland. Modeling Word-Pair Relations in a Category-Based Language Model. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-97)*, Munich, Germany, 1997.

E. Rasmussen. Clustering Algorithms. In: W.B. Frakes and R. Baeza-Yates (eds.), *Information Retrieval: Data Structures and Algorithms*, pp. 419–442, Englewood Cliffs, NJ, 1992. Prentice Hall.

S.E. Robertson and J.K. Spärck. Relevance Weighting of Search Terms. *Journal of American Society for Information Science*, 27:129–146, 1976.

R. Rosenfeld and P. Clarkson. Statistical Language Modeling Using the CMU-Cambridge Toolkit. In: *Proc. EUROSPEECH-97*, Rhodes, Greece, 1997.

G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

F. Schiel and U. Türk. Wizard-of-Oz Recordings, 2006. In this volume.

K. Seymore and R. Rosenfeld. Large-Scale Topic Detection and Language Model Adaptation. Technical report, School of Computer Science, Carnegie Mellon University, June 1997.

R. Srihari and C. Baltus. Combining Statistical and Syntactic Methods in Recognizing Handwritten Sentences. In: *Proc. AAAI Symposium: Probabilistic Approaches to Natural Language*, pp. 121–127, Cambridge, MA, 1992.

Z. Valsan and M. Emele. Thematic Text Clustering for Domain Specific Language Model Adaptation. In: *Proc. IEEE Automatic Speech Recognition and Understanding ASRU 2003*, St. Thomas Island, USA, 2003.

Y.Y. Wang. Robust Spoken Language Understanding. In: *Proc. EUROSPEECH-01*, Aalborg, Denmark, 2001.

The Dynamic Lexicon

Silke Goronzy, Stefan Rapp, and Martin Emele

Sony International (Europe) GmbH
Sony Corporate Laboratories Europe, Advanced Software Laboratory, Stuttgart, Germany
silke@goronzy-privat.net, {rapp,emele}@sony.de

Summary. The dynamic lexicon is one of the central knowledge sources in SMARTKOM that provides the whole system with the capabability to dynamically update the vocabulary. The corresponding multilingual pronunciations, which are needed by all speech-related components, are automatically generated.

Furthermore, a novel approach for generating nonnative pronunciation variants and therefore for adapting the speech recognizer to nonnative accents is presented. The capability to deal with nonnative accents is crucial for dialogue systems dealing with multimedia and therefore often with multilingual content.

1 Introduction

This chapter describes the dynamic lexicon. This module is distinguished from comparable components in other dialogue systems by its capability to dynamically update its vocabulary with new words and to automatically generate multilingual pronunciations for these words using grapheme-to-phoneme conversion. It therefore serves as a central knowledge source for several SMARTKOM modules that deal with the recognition, analysis and synthesis of speech. For dialogue systems like SMARTKOM that are characterised by their highly variable application content, this is an essential feature. In applications such as city information systems, cinema or TV program guides, or multimedia databases, it is not possible to foresee the vocabulary for the speech-related modules at the time of the design of the system as they are determined by accessing dynamically changing content (Rapp et al., 2000). Online updates of the vocabulary that are consistently propagated to all modules involved are therefore absolutely necessary to guarantee a smoothly running system. We describe such a lexicon along with all involved technologies as well as design issues that are of concern for the whole dialogue system.

It is also part of this chapter to describe a novel technique to automatically generate nonnative pronunciation variants. Such pronunciations are necessary in systems where the user is using words or sentences in a language that is different from his/her mother tongue. In these cases the words will be pronounced in an accented

way, which causes some difficulties for speech recognisers that are usually trained on native speech only. Generating alternative pronunciations that reflect accented pronunciations and keeping them in the dictionary can help alleviate this problem to a certain extent.

The remainder of the chapter is organised as follows: Sect. 2 illustrates the need for a dynamic lexicon in SMARTKOM before Sect. 2.1 explains how the lexicon is integrated in the SMARTKOM architecture. The updateable content is described in Sect. 2.2 before Sect. 2.3 explains how dynamic changes of the vocabulary take place. The automatic generation of multilingual pronunciations via grapheme-to-phoneme conversion is described in Sect. 3. Finally, Sect. 4 outlines the novel approach for the automatic generation of accented pronunciation variants.

2 The Dynamic, Multilingual Lexicon

Information systems like SMARTKOM often need to access databases, whose content might change frequently. In all three scenarios — SMARTKOM-Home (Portele et al., 2006), SMARTKOM-Public (Horndasch et al., 2006) and SMARTKOM-Mobile (Berton et al., 2006a; Malaka et al., 2006) — this is the case for the content of the information the user might want to access.

One example is a TV program that is accessed via an Internet-based electronic program guide (EPG) in SMARTKOM-Home. In this case it is obvious that the content changes daily, and sometimes even more often. One might argue that these EPGs are often available several days in advance, but generating a static dictionary every five days and loading it to the system is too expensive and cumbersome. In addition, sudden program changes might occur that need to be reflected in the system.

When retrieving the TV program for a certain time, channel etc. specified by the user, the results will typically be a list of TV shows that are then adequately presented to the user via the presentation manager (Poller and Tschernomas, 2006). Even though SMARTKOM offers the possibility of using pointing gestures for selecting icons, etc., on the screen, the system should not be limited to this modality for the selection of show titles. Speech input should also be possible, which in this case means that the user can speak the TV show titles resulting from the user query. Currently dialogue systems usually cannot automatically handle this kind of dynamic content. As a result, they need to keep as many words as possible in the system dictionary right from the beginning, which increases the dictionary size and thus the speech recognition error rates. It also causes the subsequent components (e.g., speech analysis) to become more complex. Furthermore, it is often not possible to determine the necessary words in advance (like in the TV program example), and thus such systems risk working with an incomplete dictionary that restricts the user to use only a certain set of words.

As a result, a system like SMARTKOM requires dynamic updates of the recogniser dictionary to reflect the actual titles. Thus, there is the need for a lexicon module that is able to dynamically update the vocabulary of the system and inform all other modules that might depend on this information. In the case of the lexicon this means

that new words can be added (and removed) to (from) the dictionary and the pronunciation of these new words will be provided automatically. Especially in the context of multimedia content and information such as movies or music titles, the original language of the content needs to be taken into account when the pronunciation is generated.

While it seems to be an obvious step to allow words to be added to the lexicon, it is not so clear how exactly this can happen. There are many open questions that need to be answered. First of all, an identification of words that do actually change frequently is necessary. Simply allowing everything for the update-able content is not suggested since this might lead to unnecessary lexicon updates. Second, it is not a trivial point to determine which module in the SMARTKOM architecture is best aware about the current status of the other modules and can thus deliver the information to the lexicon, which words are needed for the next dialogue steps and which are to be added to the lexicon. It then needs to be clarified what is the appropriate timing for lexicon updates and how this can be synchronised with other modules to allow for a more responsive system. Last but not least, it needs to be clarified which kind of pronunciations are to be generated, taking into account the requirements of the different scenarios.

All these questions played an important role for the design of the dynamic lexicon. How these problems were solved is described in the following sections.

2.1 The Lexicon as a Central Knowledge Source

In the above EPG example, the speech recogniser was named one of the components that needs to access dynamic vocabulary information. However, besides the speech recogniser (Berton et al., 2006b), other modules also need to do so. More concretely, the prosody recogniser (Zeißler et al., 2006) also needs to be able to recognise new words. Furthermore the speech analysis (Engel, 2006) needs to be able to interpret a user's input, possibly referring to the new content. Finally the speech synthesis (Schweitzer et al., 2006) needs to be able to generate output that might also contain new words.

Thus the lexicon can be considered as one of the central knowledge sources of SMARTKOM. As such, all dynamic information is managed in only one place so that it can be ensured that all above-mentioned modules use the same information at any time. This increases consistency, which is mandatory to show consistent system behaviour during the interaction with the user. The placement of the lexicon in the SMARTKOM system architecture is shown in Fig. 1. This figure does not show the SMARTKOM system in detail but focuses on those components that are relevant for describing the functionality of the lexicon. The normal arrows indicate the direct links between the modules, while the bold arrows are meant to show those modules in particular that depend on the lexicon.

2.2 "Updateable" Content

It is to be expected that not all words that are used in a dialogue system change frequently. There will always be a certain set of words that the system should always

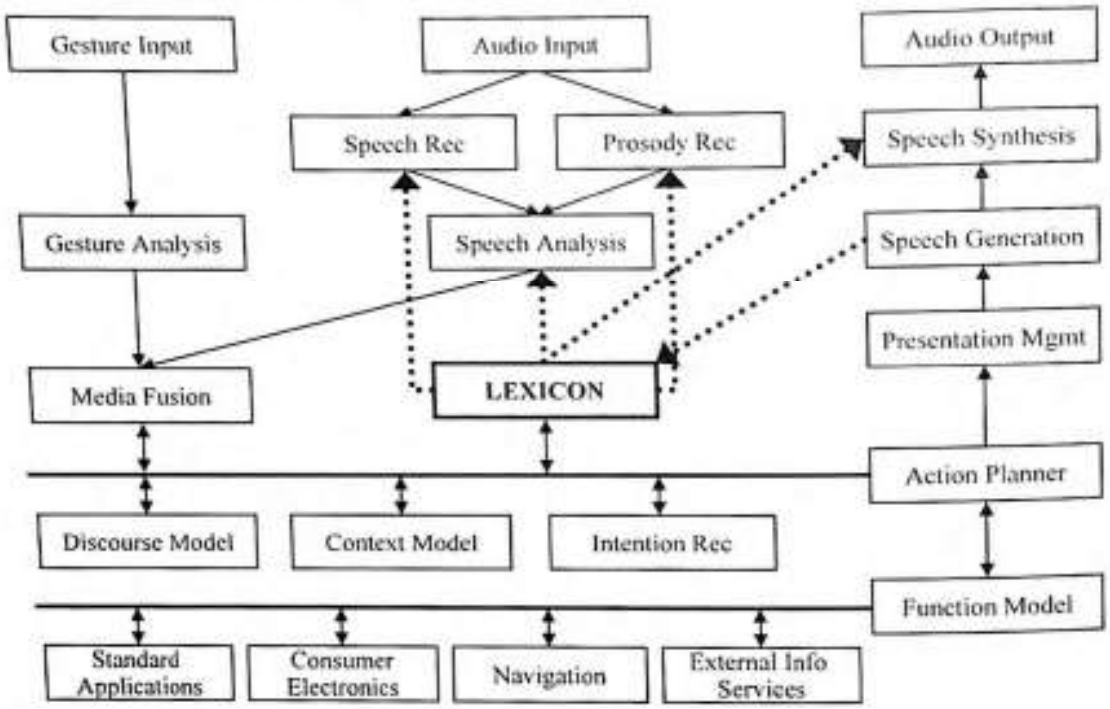

Fig. 1. SMARTKOM architecture (part of the system)

be able to recognise and process, whereas there are others that are only important in certain dialogue turns. "General" words like "please, show, me, ...", etc., are contained in a base lexicon whereas, e.g., movie titles, which are only important when the user is talking about movies, are dynamically loaded. We thus defined a set of certain word classes that can be updated dynamically. In the EPG example such a class would be called MOVIETITLE. Other word classes that were used are ACTOR, CITY, GARAGE, GENRE, HOTEL, LOCATION and MOVIETHEATER. These are closely coupled with the word classes that are used by the language model as described in Goronzy et al. (2003). A detailed explanation can be found in Emele et al. (2006).

2.3 Dynamic Updates

2.3.1 Adding Words to the Lexicon

The dynamic updates of the lexicon are initiated by the action planner (Löckelt, 2006). The action planner is the module that also initiates requests to external services such as the EPG. It is thus aware that a request for "movies on TV for tonight" will result in a list of movie/TV show titles. Whenever the query result is received by the action planner, it forwards the relevant information to the lexicon. Since the SMARTKOM system is using the pool architecture as described in Herzog and Reithinger (2006), the lexicon extracts all relevant words after having found a vocabulary update in the output pool of the action planner. It then tests whether these words are

already contained in the current lexicon. If so, it is checked whether a pronunciation in the requested language, specified by a language tag in the update request, exists. If everything is there already, no action is taken. If the word is not yet contained in the lexicon, an entry for it is added. Additionally, the pronunciation for the word is generated by automatic grapheme-to-phoneme conversion (G2P), as described in Sect. 3. An update is written to the output pool as described in Sect. 2.4.

2.3.2 Replacing Words in the Lexicon

A situation that often occurs when a user is searching for some content is the following: he/she will ask the system for something and will receive an answer. However, often a refinement of the user's request to reach the final decision will happen. In the EPG example, users might ask for "movies that are on TV tonight". They might get a lot of results, so they refine this query to "only news shows for tonight." Such multiple requests will result in repeated lexicon updates of the same word classes (in this case MOVIETITLE). But since it is not necessary to always keep all words in the lexicon, even the results from the very first request, the contents of the word classes are replaced by the result of the new request. This is the default behaviour of the lexicon, but other behaviour like adding/removing words can always be requested explicitely.

2.3.3 Removing Words from the Lexicon

Sometimes a user request might result in a very long result list, e.g., when the request is not very specific. All these words are then added to the lexicon. Also, when the system is used for a longer time words might be repeatedly added to the lexicon. This can increase the lexicon size such that it might negatively affect the speech recogniser performance. It thus makes sense to remove words again that are no longer needed. This can, e.g., be done if the topic of the interaction changed. Once the user selected a movie to be recorded at home on his VCR and then requests some information about Heidelberg, all the previously loaded movie titles are considered to be no longer needed and are thus removed from the lexicon.

2.4 Update Mechanisms

After the update request was processed as described above, the new information needs to be propagated to the modules that need the lexical information. For this two different possibilities were implemented, the full update and the incremental update. At the startup of the lexicon it can be specified which of the two updates should be performed, as it depends on the other module's capabilities which updates can best be handled. In both cases the updates are written to a pool, called modelled lexicon, that the speech recogniser, the speech analyser and the prosody recogniser have subscribed to. The speech synthesis is treated a little differently in this respect, as will be explained in Sect. 2.4.3.

2.4.1 Full Update

In this mode, whenever there is a change in lexicon initiated by the action planner, the full lexicon is written to the corresponding pool and read by all modules that subscribed to it. This solution is rather impractical for frequent dynamic updates and was more intended to serve as a "lexicon synchronisation" help during system testing. In real system use, this update is of minor importance.

2.4.2 Incremental Update

The update mode that is most frequently used in SMARTKOM is the incremental update. In this case the lexicon writes only the differential lexicon to the pool, meaning those parts that were changed. That means if a list of words is added to the lexicon, only these words together with the generated pronunciations are written to the pool and read by the other components.

2.4.3 Special Requests

In Sect. 2.1 the speech synthesis module was also named one of the modules that needs to have access to the lexicon. However, this module is treated differently from the others. While the other modules always need the complete lexicon for their tasks, they need to be informed about all updates. The synthesis, on the other hand, needs the pronunciations for the words it is supposed to synthesize. As a consequence, there is a dedicated pool that can be used to explicitly request pronunciations for certain words and sentences. These are looked up or, if necessary, generated and then used to synthesize a system answer. In SMARTKOM the speech generation generates the sentence that is to be synthesized as the system response to the user. Thus the speech generation sends a request to the pool modelled.lexicon.request containing the words to be synthesized. The lexicon then adds the pronunciation for each word if it is contained in the dictionary; if not, the word is added as usual to the lexicon and the pronunciation is automatically generated. The words together with all pronunciations are then written to the pool modelled.lexicon.response, where they are read by the synthesis module and used for generating the speech output.

2.5 Intelligent Handling of Movie Titles

In the Home scenario the problem arose that, e.g., the movie titles that are contained in the EPG database are often rather long and furthermore often also contain subtitles, like "James Bond, 007: *Tomorrow never dies*". If this title were added to the lexicon as described above, users would have to speak the complete title if they want to select this movie. In reality, however, users tend to say "record James Bond" or "record 007", rather than the complete title. This is accounted for by the lexicon by generating multiple pronunciations that are assigned to the complete title. The complete title needs to be kept as the orthography for the shortened pronunciations to ensure consistency for further database requests concerning this title. The various

pronunciations then reflect the shortened versions of the titles as the users usually say them. The shortened title versions are determined by splitting the titles at special symbols such as "-", "!", ",", etc., that we found to often separate titles and subtitles. This feature allows a more natural use of SMARTKOM while interacting with the EPG application.

3 Automatic Generation of Pronunciations

After discussing the need for a dynamic lexicon and how it is integrated into the SMARTKOM system, we now describe how we actually derive the pronunciations for new words. Automatic grapheme-to-phoneme conversion (G2P), the task of mapping a written word (that is, a sequence of letters) to its pronunciation (that is, a sequence of phonemes) can be achieved in various ways. First, in the knowledge-based approach, a set of manually written rules can be deployed that define how a given letter or group of letters is pronounced in a given graphemic, phonemic, morphological or other context (such as a specific foreign language origin). Such rules can frequently be found printed at the beginning or end of pronunciation dictionaries to advise speakers on the pronunciation of words not found in the dictionary, that is, for a task very similar to the one the dynamic lexicon has in the SMARTKOM system. In practice, however, manually written rules for G2P are often not very precise. For instance, we implemented a subset of the rules from a standard German dictionary (those that were reasonably easy to implement) and found that our rules achieve about 15% errors on the phoneme level, so that only less than every second word is correct.

Second, it was proposed to use corpus-based (or data-driven) methods to learn the mapping by some machine learning (ML) program such as neural networks (Sejnowski and Rosenberg, 1987). Besides neural networks, various other ML techniques can be used. In our work, we rely on the induction of decision trees (Black et al., 1998; Rapp and Raddino, 1999). Obviously, whereas manually written rules must be implemented again for every new language, the corpus-based approach has the advantage that it is largely independent of the language in question and only presupposes the existence of a rather large pronunciation dictionary as a training corpus. In our work, we found that the automatically trained methods achieve much higher precision than our manually written rules (Rapp and Raddino, 1999).

The contribution of our G2P work is twofold. First, we propose a hybrid two-stage architecture for multilingual G2P. In the first stage, phonemic pronunciations are generated using a corpus-based approach, and in the second stage, manually written rules map from phonemic to phonetic pronunciations, including pronunciation variants. Second, we introduce boosting, a meta-learning technique to the decision tree induction of the first stage and find that this considerably improves accuracy at the expense of extra memory. We examine the influence of training parameters on size and accuracy of decision tree classifiers to optimise space in the first stage. The parameters we vary are the amount of pruning applied during construction of the classifiers, the number of classifiers and the size of training set.

3.1 Architecture

We propose a two-stage process for G2P, with the orthography as input, an abstract phonemic representation as an intermediate step and the phonetic representation as output. The intermediate phonemic representation is also applicable for automatic speech recognition, but performance is expected to be better for the phonetic representations.

The two-stage architecture offers flexibility in creating lexicons. The phonemic representation output from the first stage gives a canonical ("standard") pronunciation. The second stage can be altered to derive pronunciation variants, such as those needed to represent different dialects, foreign accents or recognizer phone sets. For SMARTKOM, the two-stage approach simplifies development and contributes to the recognition performance.

For one, we could take advantage of the rather large coverage German pronunciation lexicon CELEX (Baayen et al., 1993) to achieve good prediction accuracy. CELEX has listed about 360,000 pronunciation entries, much more than what would have been available from the SMARTKOM and Verbmobil data collections. On the other hand, the pronunciations found in CELEX are more phonemic (or canonical) oriented than what is typically deployed in state-of-the-art speech recognizers. This difference is accounted for in the second stage. By comparing the output of the first stage's classifier with the transcription on which the recogniser is trained, rules were developed by phonetic experts that map from the learned phonemic representation to the more phonetic phone set of the recogniser. In this process either a single or multiple pronunciations can be generated. Thus, the recogniser is supplied with acoustically reasonable pronunciations while the corpus-based learning can be kept simpler, as the data for the learning task is more consistent. Finally, by exchanging the standard rules of the second stage by speaker-dependent ones—either by modifying the rules or by using a data-driven approach as described in Sect. 4, the system can be adapted to dialectal or nonnative variations of speech.

For the first step, the prediction of the phonemic representation, we deploy decision tree classifiers that are induced from language-specific training data. The induction itself is language-independent. As described in Rapp and Raddino (1999), we use the C5.0 tool,[1] which first builds tree classifiers and then optionally transforms them into more memory-efficient rule sets of comparable accuracy. Preparation of the data for C5.0 requires alignment of the dictionary to relate each grapheme symbol to one or more phoneme symbols, and vice versa. One-to-one relations between graphemes and phonemes are established by inserting additional "null" graphemes or phonemes using Viterbi decoding: If there was a 1:1 relation between grapheme and phoneme, the inserted null phoneme will correspond to the inserted null grapheme. Otherwise, either a grapheme or a phoneme will correspond to a null symbol, see Rapp and Raddino (1999).

Pronunciations are predicted moving grapheme by grapheme from right to left, mapping each grapheme symbol to a phoneme symbol. For every grapheme, two contexts are considered: (1) three (non-null) graphemes to the right and to the left, and

[1] http://www.rulequest.com

(2) the already predicted phonemes corresponding to the three (non-null) graphemes to the right. To allow for prediction of word stress and syllable boundaries as well, the number of stressed syllables to the right of the current syllable and the number of consonants up to the next vowel to the right are also taken into account.

For the second step, the generation of one or more phonetic representations, we use rules written by phonetic experts. Currently, we have implemented such rules for American English and German. For American English, they generate pronunciation variants. For German, they currently only constitute some postlexical rules (Sect. 4).

3.1.1 Building Classifiers

Like most machine learning algorithms, C5.0 can be adapted to different learning tasks by setting some training parameters. These parameters have an influence on the degree of generalisation of the resulting classifiers. There has been speculation that generalisation is often harmful in natural language tasks and that the best results are achieved by highly specialised classifiers (Daelemans et al., 1999).

The parameter with the strongest influence on generalisation in C5.0 is the prepruning parameter m, which specifies the minimum number of cases that must be covered by each branch in the tree. Small values for m yield highly specialised, large trees, while large values give smaller classifiers that generalise more strongly.

Boosting is a machine learning technique that is known to significantly improve prediction accuracy for many tasks by combining several classifiers (Bauer and Kohavi, 1999). During training, each classifier emphasizes the cases misclassified by the previous one. For prediction, each classifier is applied, and the class predicted by most classifiers will be chosen. Boosting requires more memory since several trees have to be stored. An interesting question is whether the accuracy improvements of boosting can be maintained even if the individual trees are pruned strongly to save space.

3.1.2 Experiments

We investigated the relation between size and word accuracy rate of the tree classifiers for English and German. We varied the prepruning parameter m (2 to 256) and the number of trees used for boosting (1, 5, 10, 15) on training data of 20K and 100K words sampled from Baayen et al. (1993). The classifiers were evaluated ignoring errors in the assignment of word stress and syllable boundaries.

Figure 2 shows the results for German. We are comparing the size (number of nodes in the trees, in logarithmic scale, on the x-axis) and the word accuracy rate (in %, on the y-axis) of the classifiers.

The best word accuracy rates can be achieved for larger sets of training data when using boosting. However, both factors result in larger trees, as seen in Fig. 2. Results for the 100K training set are marked by gray symbols, and results for the 20K training set are indicated by white symbols. In general, the gray 100K series in the diagram show better accuracies with larger numbers of nodes than the white 20K series. We used boosting with 5, 10 and 15 classifiers, represented by the triangle,

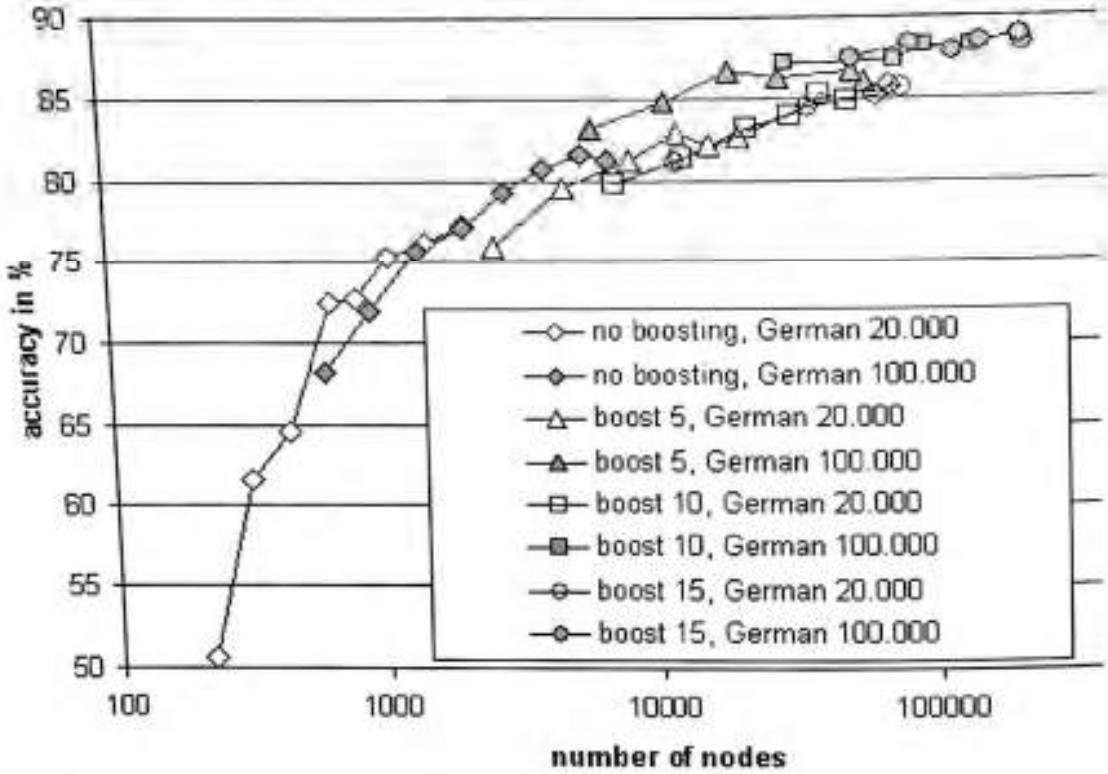

Fig. 2. Word accuracy rates for classifiers trained on german data

square and circle series, respectively. Boosting gave higher rates than did the series without boosting, as indicated by the diamond symbols, again at the expense of larger classifiers.

An interesting comparison is between the strongly pruned trees trained on the 100K training set and the less strongly pruned trees built from the smaller 20K set. The trees were comparable in size, but the more specialised 20K trees outperformed the 100k trees (cf. the series with the white and gray diamond symbols in the diagram). This means that the generalisation originating from stronger pruning is more harmful than the one originating from a smaller training set.

As for the question whether the high accuracy rates achieved for boosting can still be maintained if the individual trees are strongly pruned, the answer is yes. Moreover, when reducing the pruning factor to apply less pruning, word accuracy rates only improve down to a pruning factor of $m = 32$, then stagnate or deteriorate for $m < 32$, as the trees get bigger and bigger. (For British English, the threshold was found to be $m = 16$). Also, boosting with 15 classifiers did not yield significantly better accuracy rates (the circle series and the square series overlap in Fig. 2).

For British English, the same tendencies can be observed. Word accuracy rates figure at about 9% below the ones for German. The best result for British English, 80.4%, was achieved using boosting with ten classifiers and a pruning factor of

$m = 16$ on training data from a 100K dictionary, with a total of approx. 195K nodes. This accuracy compares favourably to the results of others. The same parameters on 20K training data yielded 69.5% accuracy. For American English, first experiments on training data of 53K entries show lower accuracies, namely 63%. The difference between American and British English is attributed to the different lexical resources that were deployed. On the other hand, the considerably improved accuracy of German over English is consistent with the findings of others and can be explained with the conservative writing system of English.

3.2 Discussion of G2P results

We presented a language-independent procedure to obtain classifiers for G2P and evaluated it on German, British English and American English data. It can be said that the optimal training parameters are heavily dependent on the language, the size of the training data and the desired size of the trees. Optimal word accuracy rates can be reached with large amounts of training data using boosting in combination with careful pruning of the individual classifiers. However, optimal classifiers come at the cost of huge memory consumption. For many applications, a reasonable compromise between size and accuracy has to be found, as it is not always possible to afford arbitrary memory for every system component. In its totality we could not confirm the position of Dealeman et al. that generalization is harmful for the G2P task. If also memory consumption is considered as an additional dimension to optimize for, it becomes obvious that generalization is to be pursued. In our data we also can see clearly the advantage of using boosting for the G2P task. For the 20K lexicon, boosting improves word accuracy by about 9%, and for the 100K lexicon by about 7%.

3.3 Multilingual Pronunciations

For the base lexicon a default language is defined in the M3L document containing all words of the base vocabulary. If new words are added to the lexicon, they are generated in this default language unless it is explicitly requested to generate them in another language. The languages supported by the lexicon are German and British English. These two languages can be provided by using two decision trees for G2P, one trained on German, the other trained on English. Again, movie titles are treated differently from other words. Especially for movie and music titles, it is difficult to determine a default, since they are often mixed. Sometimes English subtitles are translated to German, while the original title remains unchanged, sometimes everything is translated, sometimes the title is completely English. Since we do not know what was translated to German and what was not, both an English and a German pronunciation is always generated for movie and music titles to allow the greatest flexibility in terms of pronunciation for the user. Additionally, there is the problem of a user speaking, e.g., an English title, with a German-accented pronunciation. This case needs special treatment for the generation of appropriate pronunciations, and an approach for this is described in Sect. 4.

4 Pronunciation Adaptation

The methods implemented for lexicon updates allow to add multiple pronunciations for words. In Sect. 3.3 we already discovered the problem of nonnative pronunciations. If we want to reflect such pronunciations to increase recognizer robustness w.r.t. nonnative speech, the question is how to derive such accented or nonnative pronunciations that users typically use for speaking, e.g., English titles with a German accent. A novel method was developed within SMARTKOM and will be outlined in this section. A detailed description can be found in Goronzy (2002) and Goronzy et al. (2004). There are several approaches to pronunciation modeling. They can be categorized into two broad classes, which are knowledge-based and data-driven approaches.

Knowledge-based approaches make use of existing knowledge that was derived by experts (e.g., dictionaries). Such knowledge is often used to generate rules that are able to generate typical pronunciation variants from canonical pronunciations or from the orthography of a word. Data-driven approaches try to derive the pronunciation variants directly from the speech signal, e.g., using manual transcriptions or automatic transcription using phoneme recognisers. In the latter case, in order to obtain a certain generalization capability to new words, the generated variants are often not used as they are. Rules can be generated therefrom, or decision trees can be grown that are able to predict the pronunciation variants.

In general, the pronunciation variants that are generated can then be added to the dictionary. A detailed overview on pronunciation modeling can be found in Strik and Cucchiarini (1999).

While knowledge-based approaches are completely task-independent, they tend to overgenerate. In contrast, data-driven approaches are rather database dependent. For our problem we would need to explicitly record accented speech databases, which is a cumbersome and expensive task.

4.1 Automatic Generation of Nonnative Pronunciation Variants

Since we are interested in pronunciations for various accents and languages, both approaches are too expensive. We need an approach that can flexibly be applied to various combinations of source (the native language of a speaker) and target languages (the language they are trying to speak).

The automatic method that was developed exploits the fact that many native databases exist. It is a data-driven approach but solely uses native speech to derive nonnative pronunciations. This is achieved by simulating one of the many different possibilities of humans to learn a new language, namely to listen to native speakers of the target language and try to reproduce what was just heard. If we assume that the learner is completely new to the target language two things are likely to happen:

- He/she will use phonemes known from the source language.
- He/she will try to use phonotactic constraints inherent to the mother tongue.

The first item simply means that phonemes that exist in both languages will be pronounced as in the source language rather than as in the target language. This can be handled easily by speaker adaptation techniques that adapt the Hidden Markov Models (HMMs) to the current speaker. The second point means that speakers will tend to reproduce certain phonotactic structures characteristic to their source language when speaking the target language, thus resulting in wrong phoneme sequences. The consonant–vowel structure of Japanese might serve as an example here. Beacause Japanese speakers often experience difficulty in pronouncing subsequent consonants, they often insert vowels, like in "table": /t eI b l/ becomes /t eI b u r u/. In addition to the pronunciation of the /l/ as an /r/, two phonemes (/u/) are inserted before and after the /r/. Standard acoustic speaker adaptation algorithms are often used to improve performance for nonnative speakers. While taking into account that certain phonemes are pronounced differently, they totally neglect potentially different phoneme sequences. In the above example, acoustic adaptation would try to adapt the HMM for /l/ to /u r u/ since it uses the native canonical pronunciation as a reference for adapting the HMMs. However, in reality three instead of one phonemes have been produced, which would cause a misadaptation that might cause the overall recognition rates to decrease.

In order to generate these accented pronunciations, we simulate the learning process by using a phoneme recognizer that is trained on the source language and uses a source language phoneme-bigram to reflect the phonotactic constraints of the source language. This source language phoneme recognizer simulates the learner trying to reproduce a word. We thus let this recognizer recognize native speech of the target language. For example, we use a British English phoneme recognizer to recognize German speech, and the result is an English-accented pronunciation for German words. Since phoneme recognition usually is rather error-prone, some kind of generalisation is necessary to filter out pronunciations that contain a lot of errors. We achieve this by training a decision tree that learns a mapping from the German canonical to an English-accented pronunciation. Adding these generated variants to the lexicon, recognition rates on nonnative speech could be improved by up to 5.2%, as described in Goronzy et al. (2004). The method is particularly effective when combined with speaker adaptation techniques, as was also shown in Goronzy et al. (2004). Then recognition rates can be improved by up to 18%. The processes of generating the variants (1) and then employing these later in the recognition phase (2) are shown in Fig. 3.

5 Conclusion

In this chapter the dynamic lexicon was described in detail. A module like this allows dialogue systems to flexibly react to user requests that refer to dynamic content such as music databases or electronic program guides for TV. State-of-the-art dialogue systems usually are not able to handle such dynamic content. As a consequence, users cannot speak, e.g., movie titles, which renders the dialogue very inflexible. In SMARTKOM a very flexible behaviour concerning this aspect was achieved by using

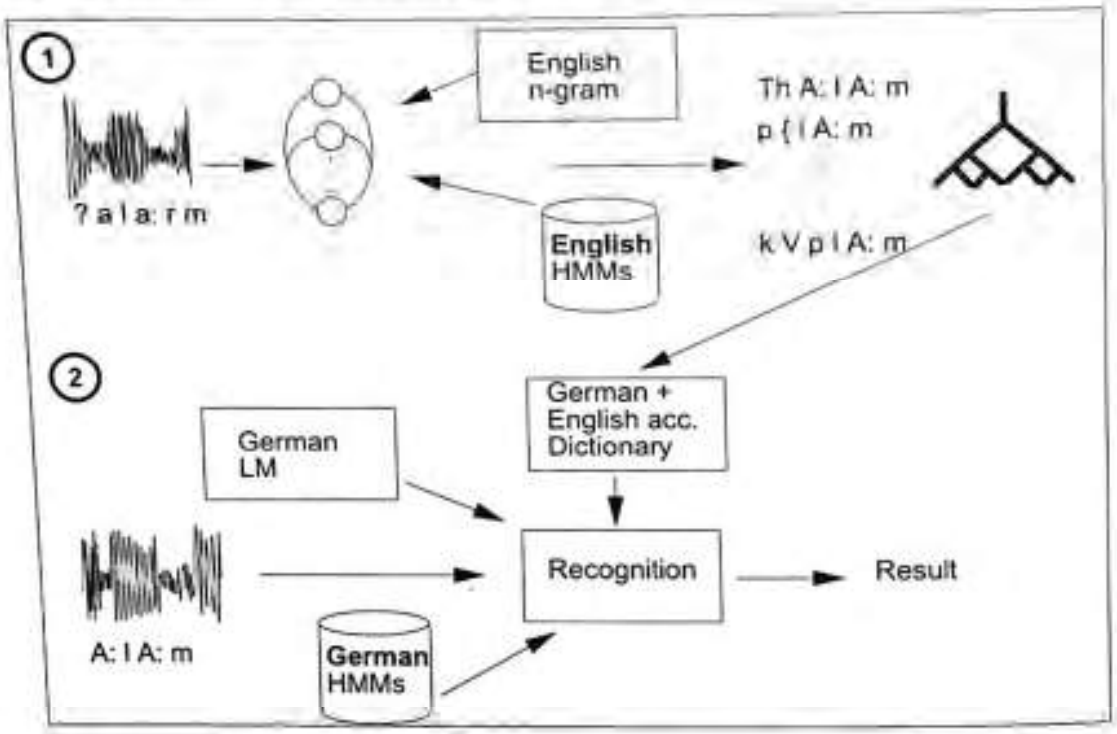

Fig. 3. *1*: Method to generate normative pronunciations. *2*: Final recognition system with extended dictionary

a lexicon that can accept such content for inclusion and that is well synchronized with all necessary dialogue modules. We described the integration of the lexicon into the SMARTKOM architecture and the methods to perform the dynamic updates.

The automatic generation of multilingual pronunciations using a grapheme-to-phoneme conversion approach was also explained. Furthermore, a novel approach to automatically generate accented pronunciations for inclusion into the lexicon that is very flexible w.r.t. different accents and does not need any accented speech data was outlined.

Acknowledgments

We would like to thank Ralf Kompe for the helpful discussions, which helped give not only the lexicon but also SMARTKOM its direction. Our colleague Jürgen Schimanowski was an enormous help in the implementation and tuning phase — thanks! We are also very grateful to Antje Schweitzer, who made a crucial contribution to the works concerning G2P. Anton Batliner shared his expertise on all phonetic issues that arose and manually checked a vast number of base lexicon pronunciations. Last but not least, we want to thank all other SMARTKOM colleagues involved in the lexicon activities for the very good cooperation.

References

H. Baayen, R. Piepenbrock, and H. van Rijn. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, 1993.

E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36(1–2):105–139, 1999.

A. Berton, D. Bühler, and W. Minker. SmartKom-Mobile Car: User Interaction With Mobile Services in a Car Environment, 2006a. In this volume.

A. Berton, A. Kaltenmeier, U. Haiber, and O. Schreiner. Speech Recognition, 2006b. In this volume.

A. Black, K. Lenzo, and V. Pagel. Issues in Building General Letter to Sound Rules. In: *Proc. 3rd ESCA Workshop on Speech Synthesis*, pp. 77–80, Jenolan Caves, Australia, 1998.

W. Daelemans, A. van den Bosch, and J. Zavrel. Forgetting Exceptions Is Harmful in Language Learning. *Machine Learning*, 34:11–41, 1999.

M. Emele, Z. Valsan, Y.H. Lam, and S. Goronzy. Class-Based Language Model Adaptation, 2006. In this volume.

R. Engel. Natural Language Understanding, 2006. In this volume.

S. Goronzy. *Robust Adaptation to Non-Native Accents in Automatic Speech Recognition*. LNAI 2560. Springer, Berlin Heidelberg New York, 2002.

S. Goronzy, S. Rapp, and R. Kompe. Generating Non-Native Pronunciation Variants for Lexicon Adaptation. *Speech Communication*, 42(1):109–123, 2004.

S. Goronzy, Z. Valsan, J. Schmanowski, and M. Emele. The Dynamic, Multi-Lingual Lexicon in SmartKom. In: *Proc. EUROSPEECH-03*, vol. 3, pp. 1937–1940, Geneva, Switzerland, 2003.

G. Herzog and N. Reithinger. The SmartKom Architecture: A Framework for Multimodal Dialogue Systems, 2006. In this volume.

A. Horndasch, H. Rapp, and H. Röttger. SmartKom-Public, 2006. In this volume.

M. Löckelt. Plan-Based Dialogue Management for Multiple Cooperating Applications, 2006. In this volume.

R. Malaka, J. Häußler, H. Aras, M. Merdes, D. Pfisterer, M. Jöst, and R. Porzel. SmartKom-Mobile: Intelligent Interaction With a Mobile System, 2006. In this volume.

P. Poller and V. Tschernomas. Multimodal Fission and Media Design, 2006. In this volume.

T. Portele, S. Goronzy, M. Emele, A. Kellner, S. Torge, and J. te Vrugt. SmartKom-Home: The Interface to Home Entertainment, 2006. In this volume.

S. Rapp and D. Raddino. Trained Automatic Grapheme to Phoneme Conversion. In: *Proc. 9th Sony Research Forum (SRF'99)*, Tokyo, Japan, 1999. Sony Corporation Corporate Technology Dept.

S. Rapp, S. Torge, S. Goronzy, and R. Kompe. Dynamic Speech Interfaces. In: *Proc. Workshop Artificial Intelligence in Mobile Systems at the 14th ECAI (AIMS 2000)*, Berlin, Germany, 2000.

A. Schweitzer, N. Braunschweiler, G. Dogil, T. Klankert, B. Möbius, G. Möhler, E. Morais, B. Säuberlich, and M. Thomae. Multimodal Speech Synthesis, 2006. In this volume.

T. Sejnowski and C. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1:145–168, 1987.

H. Strik and C. Cucchiarini. Modeling Pronunciation Variation for ASR: A Survey of the Literature. *Speech Communication*, 29:225–246, 1999.

V. Zeißler, J. Adelhardt, A. Batliner, C. Frank, E. Nöth, R.P. Shi, and H. Niemann. The Prosody Module, 2006. In this volume.

The Prosody Module

Viktor Zeißler, Johann Adelhardt, Anton Batliner, Carmen Frank, Elmar Nöth, Rui Ping Shi and Heinrich Niemann

Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
{zeissler,adelhardt,batliner,frank,noeth,shi,
niemann}@informatik.uni-erlangen.de,

Summary. In multimodal dialogue systems, several input and output modalities are used for user interaction. The most important modality for *human computer interaction* is speech. Similar to human human interaction, it is necessary in the human computer interaction that the machine recognizes the spoken word chain in the user's utterance. For better communication with the user it is advantageous to recognize his internal emotional state because it is then possible to adapt the dialogue strategy to the situation in order to reduce, for example, anger or uncertainty of the user.

In the following sections we describe first the state of the art in emotion and user state recognition with the help of prosody. The next section describes the *prosody module*. After that we present the experiments and results for recognition of user states. We summarize our results in the last section.

1 The State of the Art

Prosody refers to the segments of speech larger than phonemes, e.g., syllables, words, phrases, and whole utterances. These segments are characterized with properties like pitch, loudness, duration, speaking rate, and pauses. The machine can analyze these properties and detect prosodic events, such as accents and phrase boundaries, as well as decide the mood or emotion in which a human expresses a certain utterance (Adelhardt et al., 2003). With the help of prosody it is consequently possible to get more knowledge about the user who is "talking" with the system, as has been, for instance, shown in some studies (Dellaert et al., 1996; Amir and Ron, 1998; Li and Zhao, 1998; Petrushin, 2000).

User states are an extension of the well-known term of emotion with some internal states of a human like, e.g., *"hesitant"*, that are important in the context of human computer interaction (HCI). This extension of emotion refers to the interaction of users with the system, for instance, if the user shows hesitance or uncertaincy because he does not know how the machine can help him. For details, see also Streit et al. (2006).

One problem with the recognition of user states is the difficulty of data collection. In most cases actors "create" emotions according to some certain scenario, but

recognizers trained with these *actor data* are not applicable for emotion detection with naive speakers. An alternative method to collect data for training an emotional recognizer is the so-called Wizard-of-Oz experiment (WOZ, see Schiel and Türk (2006)).

In the research of emotion recognition through prosody, generally three base features are used: fundamental frequency (F0 or pitch), duration, and energy. Furthermore, these base features can be combined with several other features. In Dellaert et al. (1996) actor data of five speakers are used to detect four emotions — joy, sadness, rage, and fear. The authors use several F0 features, speaking rate and statistics about the whole utterance. In Amir and Ron (1998) actor data of more than 20 persons are used to detect joy, anger, grief, fear, and disgust. The authors use, e.g., F0, energy and derived features based mainly on the whole utterance. In Li and Zhao (1998) joy, anger, fear, surprise, sadness, and neutral are detected with actor data of five speakers. The authors use, e.g., formants, F0, energy, and derived short-term features as well as several derived long-term features. In Petrushin (2000) data of 30 persons showing 5 emotions, joy, rage, sorrow, fear, and neutral, in their speech are classified. The authors use features like F0, energy, speaking rate, formants, and bandwidth of formants. In addition, minima, maxima, average, and regression of the features above are used.

Different classification techniques can be applied to emotion classification. In Dellaert et al. (1996) maximum likelihood Bayes classification, kernel regression, and k-nearest neighbor are used for classification. In Amir and Ron (1998) the authors use two methods for classification. The first method computes wordwise emotion scores averaged over the utterance, which are compared against each other to determine the emotion. The second method suggests framewise classification followed by the final decision based on majority voting of the frames in each emotional class. In Li and Zhao (1998) the choice of the features for classification is based on principal component analysis (PCA), while classification results from vector quantization, Gaussian mixture models, and *artificial neural networks* (ANN) are also used in Petrushin (2000).

Another contribution that is very important to our work stems from Huber (2002). The author uses wordwise as well as turnwise prosodic features and linguistic information for the classification of emotion (for the features, see Sect. 2.4 and Nöth et al. (2000); Kießling (1997)).

There are some other methods for emotion detection based on evaluation of linguistic information. One possibility is keyword spotting, where the utterance is checked against certain words (Lee et al., 2002; Arunachalam et al., 2001). Another method is the use of *part of speech* features (POS) introduced in Batliner et al. (1999).

In the SMARTKOM project, speech, gesture, and facial expressions are used for emotion recognition. In the further context of emotion recognition, there are several studies in the area of the term "affective computing," which has been established mainly by R. Picard. Affective computing combines several information channels to get the emotion of the user. An interesting introduction to this field is given in

Picard (1997). It covers, e.g., emotion in speech, facial expression, and physiological signals.

2 Module Description

The prosody module used in the SMARTKOM demonstrator is based on the Verbmobil prosody module described in detail in Batliner et al. (2000a) and Nöth et al. (2000). Compared to the Verbmobil version, several major changes have been made concerning both implementation and classification models. Since the SMARTKOM system provides a different approach for module communication (Herzog and Ndiaye, 2006), the module interface has been fully reimplemented. The classification core remains essentially the same, except for some minor changes, which increase the stability and performance of the module. All existing classification models for the recognition of prominent words, phrase boundaries, and questions have been retrained on the actual SMARTKOM WOZ dataset (Schiel and Türk, 2006). This makes it possible to achieve much better recognition results than those obtained with the old models on the current dataset (Sect. 3.1). Additionally, the user state classifier has been trained and integrated into the module.

In the following sections we first give a brief overview of the overall module structure and the data flow in the module (Sect. 2.1). The issues concerning the execution of the module in the SMARTKOM system and synchronization of input data streams are covered in Sect. 2.2 and 2.3, respectively. Afterwards the features used for classification are presented in Sect. 2.4, followed by the description of the prosodic classifiers in Sect. 2.5.

2.1 Architecture

The goal of the prosody module in SMARTKOM is to analyze the speech as a modality of the user input in order to detect the prosodic events as well as the most likely emotional state of the user (Streit et al., 2006). As shown in Fig. 1, the module has two main inputs: the speech signals from the *audio module* and the word lattices (*word hypothesis graphs*, WHGs) from the *speech recognizer*. After running through feature extraction and classification steps, the detected prosodic events are added to the original input WHG and the user state lattice is generated. In more detail, the subsequent processing steps are described below:

- *XML parser:* According to the SMARTKOM communication paradigm all data exchanges are done in the XML format. The incoming XML packets have to be parsed and filtered. Thus, we can check the data consistency, drop the irrelevant information, and convert the useful data to a compact internal representation.
- *Stream synchronization:* This component compares the time stamps of incoming packets to find the correct mapping between the WHGs and the speech data. It also ensures the amount of data is enough to trigger the feature extraction for a given WHG. For the detailed description see Sect. 2.3.

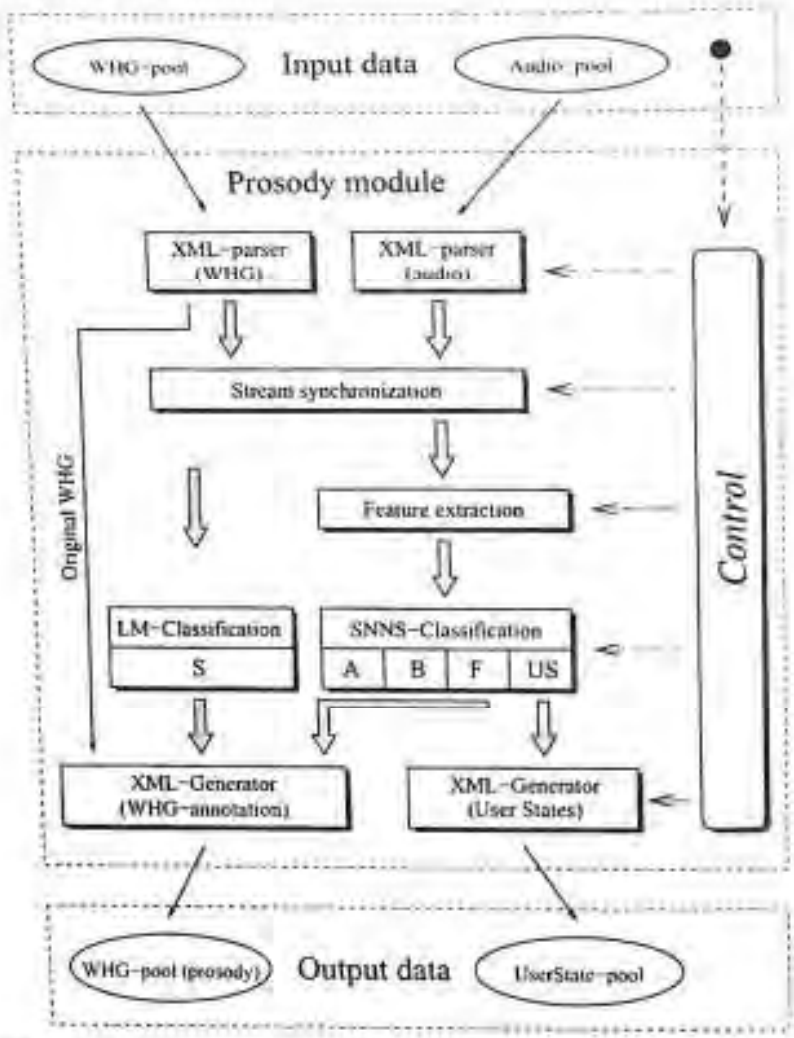

Fig. 1. Architecture of the prosody module

- *Feature extraction:* After the synchronization of the input data, a speech signal segment corresponding to each word in the WHG is located. The various F0, energy, and durational features of one segment are combined into a single feature vector to be used at the classification step.
- *ANN classification:* The ANN classification block consists of four independent classifiers used for the detection of:
  - phrase *A*ccents (A-labels),
  - phonetic phrase *B*oundaries (B-labels),
  - rising intonation for *Q*ueries (Q-labels),
  - *U*ser *S*tates (US-labels).

  For each word and label the classifiers generate a likelihood score representing a confidence of the recognition. The scores of each classifier are normalized to resemble the real probabilities (summed up to 1), though their distribution might be quite different.

- *LM classification:* The recognized words taken from the WHG are used here to detect Syntactic phrase boundaries (S-labels) with a *Language Model classifier* described in Batliner et al. (2000a); Schukat-Talamazzini (1995).
- *XML generators:* The output of classifiers is used to annotate the original WHG stored at the parsing step. Technically, we generate the XML structure only for the annotation part and paste it to the original lattice. This improves the performance of the module because disassembling of the complex WHG structure and reassembling of a new WHG from scratch can be avoided. Additionally, the user state labels are used to generate the user state lattice for the current turn. After generation, both lattices are sent to the output pools.
- *Control:* The control component determines the order of execution of all other parts and also handles the data transfer.

Apart from the main input pools shown in Fig. 1 the prosody module also gets data from the lexicon and the configuration pool. The processing of lexicon data makes it possible to update the internal static lexicon with new entries. The configuration data are used to set the internal module parameters from the GUI.

2.2 Execution

After being started by the SMARTKOM system the prosody module goes through several initialization steps including reading the configuration files, setting up the classifiers and loading statistical models. To be able to interact with the rest of the system, the module then subscribes the needed communication resources and installs the pool handlers.

The SMARTKOM communication system is based on the so-called *Pool Communication Architecture* (PCA) described in Herzog and Ndiaye (2006). There are several I/O FIFOs called communication pools which run in an asynchronous manner. If any module puts data into a pool, all modules that have subscribed this pool will be notified and can access the data. There are two possibilities to get notification events. The module can wait for these events by calling a special function or it can install handlers to be called when the notification event arrives. The prosody module handles the controlling events, such as exit or restart commands in the former and the pool notification events in the latter way.

When there are new data in one of the input pools subscribed by the module, the installed pool handler is called and the control component becomes active. First of all, the appropriate XML parser is called to get the data from the pool. When a new WHG arrives, the control component tries then to find the matching speech signal region (synchronization component). If it succeeds, the module registers itself as *processing* to the SMARTKOM system and proceeds to the further steps: feature extraction, classification and the data output.

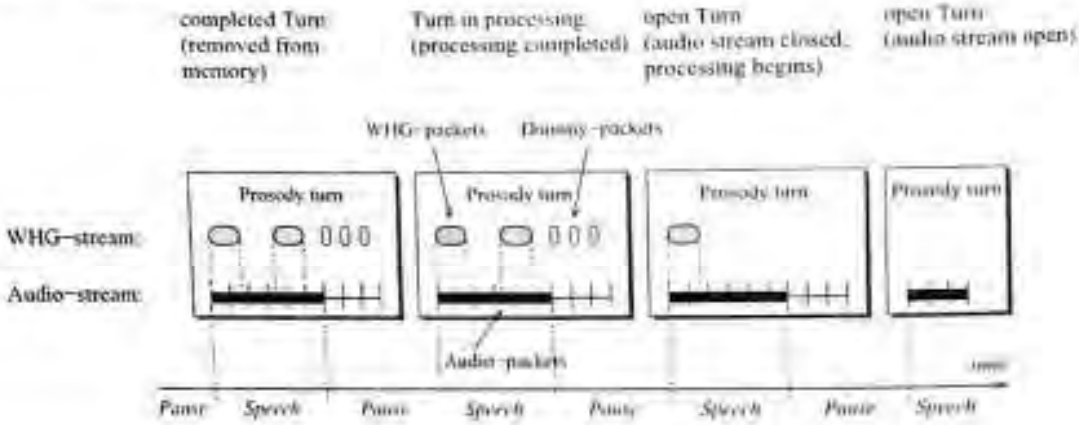

Fig. 2. Synchronization of audio and WHG streams in the prosody module

2.3 Data Synchronization

The essential part of the module managing the internal data flow is the stream synchronization component. Several problems must be solved to keep the data processing running smoothly from the technical point of view:

- memory allocation for the subsequently arriving data packets
- building the data structures needed for the following steps
- memory disallocation for the already processed data

Furthermore, some special features of the modules providing the input data should be considered. It concerns, for instance, the *silence detection* scheme. The audio module records the speech until silence is detected. The first and the last packet of the audio sequence between silences are labeled, so that such sequences can easily be identified. During silence dummy audio packets are generated in regular intervals. They are handled by the speech recognizer and can be ignored by the prosody module. The silence detection from the audio module regards only the signal energy and therefore only a robust detection of long pauses is possible. The speech recognizer has its own silence detection that also works for shorter and less distinct pauses. For every speech segment between such *short silences*, the speech recognizer generates a WHG and sends it to the output pool. Thus, there can be more than one WHG for a single audio sequence that needs to be properly aligned. Another factor to be considered is the processing of the dummy packets. The dummy packets from the audio module are transformed by the speech recognizer to dummy WHGs and sent to the WHG pool. These packets are needed for the robust detection of the user turns and dialogue management of SMARTKOM, therefore they should be passed through prosody module to the output pool "as is."

To reflect the SMARTKOM dialogue turns, the stream synchronization component works also with turn-based structures. Correspondingly, the *prosodic turn* is defined as a continuous speech sequence between the silences detected by the audio module. The speech data are collected from the incoming audio packets and stored in an array

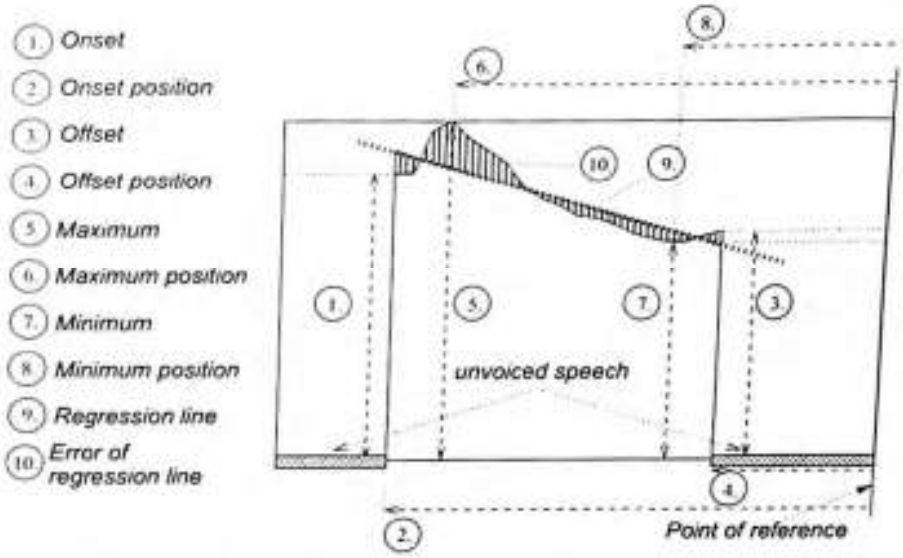

Fig. 3. Example of features describing the pitch contour (Batliner et al., 2000a)

in the turn structure. With the begin of a new audio sequence, the next turn structure is created to collect the incoming audio and WHG streams. When a new WHG packet arrives, the time stamps are compared to decide which turn the WHG belongs to. The processing scheme is illustrated in Fig. 2. Although this makes it necessary to hold open and manage several turns at a time, it has obviously the following advantages:

- The speech data belonging to one turn are stored in a single array and do not need to be pasted together from different packets. It also means smaller overhead at the feature extraction step.
- The memory management is done in the following manner: After the last WHG in a turn is processed, the complete turn, including all WHG, speech, and all intermediate data, is deleted and there is no need for costly garbage collection.

2.4 Feature Extraction

To recognize the prosodic characteristics of a speech signal we use the features describing such prosodic cues as energy, pitch, and duration, but also some linguistic information, in particularly POS features. The feature extraction is performed in two steps. First, we compute two *basic prosodic features*: integrated short time energy and pitch values framewise with a step size of 10 ms (Kießling, 1997). At the second step we take the WHG to determine the word boundaries and compute for each word a set of *structured prosodic features* (Nöth et al., 2000; Kießling, 1997). Every feature of the set has the following three configuration parameters:

- *Extraction interval:* To incorporate the word context information, the features can be computed on the subsequent words within a five-word interval beginning two words before and ending two words after the actual word. For example, we

can take a two-word interval beginning one after the word in focus and ending two words after it. We use no more than a two-word range for such intervals. The larger intervals in our experiments brought no further improvement.

- *Basic prosodic cue:* Every feature can be computed using one of the following information cues: energy and pitch (computed at the first step), duration (taken from the WHG according to the chosen extraction interval), and linguistic information (taken from the dictionary).
- *Feature type:* For every prosodic cue there are a number of different features that can be subdivided into two main groups: *contour describing* and *integrated* features. The features of the first group including onset/offset/maximum/minimum values and their positions, regression coefficient, and regression error are shown in Fig. 3. They can be applied to the pitch and some of them to the energy contour. The second group includes mean and normalized values of the energy and duration cues. To eliminate the influence of microprosodic and global factors, such as the intrinsic word duration and the speech rate (when applied to duration cue), we use twofold normalization, described in Zeißler et al. (2002).

To sum up, we compute a total set of 91 prosodic and 30 POS features, described in detail in Batliner et al. (2000b). We use different subsets for classification of different prosodic events. The total set of 121 features is used for the detection of phrase accents and phrase boundaries. With a subset of 25 F0 features we classify the rising intonation (queries). For the classification of the user states, we use only 91 prosodic features.

2.5 Classification

For the prosodic classification we use a *multilayer perceptron*, a special kind of artificial neural network (ANN). We normalize all the input features to the mean of zero and the variance of one. The normalization coefficients are estimated in advance on the whole training dataset. To find an optimal training configuration, we need to know the following parameters: network topology, training weight for the RPROP training algorithm (Riedmiller and Braun, 1993), and random seed for the initialization. In preliminary tests we found out that complex topologies with two or more hidden layers do not improve the results for our dataset in comparison to a simple perceptron with one hidden layer. Hence, we restrict the number of hidden layers to one and look only for the optimal number of nodes in the hidden layer. We then evaluate different combinations of these parameters and choose the configuration with the best result on the validation set.

As primary classification method we use the wordwise classification. For each word ω_i we compute a probability $P(s \mid \omega_i)$ to belong to one of the given user states s. The probability maximum then determines the classification result. Further, we use these probabilities to classify the whole utterance, assuming the conditional independence between word classification events (Huber et al., 1998). The utterance probabilities are computed according to the following equation:

Table 1. The classwise averaged (Cl) and total (Rr) recognition rates (in %) yielded for a two-class problem on phrase accents, phrase boundaries, and rising intonation

Classifiers	SmartKom				Verbmobil			
Tested on	SK-train		SK-test		SK-test		Verbmobil	
	CL	RR	CL	RR	CL	RR	CL	RR
Prominent words	81.2	81.0	77.1	77.0	72.8	72.8	79.2	78.4
Phrase boundaries	90.7	89.8	88.5	88.6	82.2	82.7	81.5	85.8
Rising intonation	75.6	72.0	79.2	66.4	57.4	81.1	60.1	89.7

$$P(s \mid \omega_1, \omega_2, \ldots, \omega_n) \approx \prod_{i=1}^{n} P(s \mid \omega_i) \,. \tag{1}$$

3 Experiments

3.1 Detection of Prosodic Events

Like in Verbmobil, in SmartKom we have three different prosodic events to detect, phrase accents, phrase boundaries, and rising intonation. We use the WOZ dataset, since it contains the labeling we need (Schiel and Türk, 2006). For each event we use a two-class mapping of the existing labels described in Batliner et al. (2000a): one class for the strong occurrence of the event and the other for its absence.

In our experiments we use a part of the WOZ dataset consisting of 85 public wizard sessions. It includes 67 minutes of speech data (the total session length is about 2.5 hours) collected from 45 speakers (20 m/25 w). We divide the data into training and test sets built from 770 and 265 sentences, respectively. The test set contains only speakers unseen in the training set. The recognition rates on both sets are given in columns 1 to 4 of Table 1. To ensure the newly trained networks have a better performance than those of the old Verbmobil classifiers, we also tested the Verbmobil networks on the test data. The results are given in columns 5 and 6 of Table 1 (For comparison, the results of the Verbmobil networks on the Verbmobil dataset are given in columns 7 and 8).

Comparing the results on test and training sets we see small (but significant) differences. Nonetheless, we conclude that the trained classifier has rather good generalization capability for the recognition of prosodic events on the SmartKom data. On the other hand, we observe rather big differences if comparing it with the results of Verbmobil classifiers, especially in the detection of rising intonation. It illustrates the necessity to retrain the old classifiers on the actual dataset.

3.2 Detection of User States on the WOZ Data

For the detection of user states on the WOZ data we use the same subset of 85 sessions as described above. There was no special *prosodic* labeling of user states, thus we used a so-called *holistic* labeling based on the overall impression of the labeler,

Table 2. Original labels, used class mappings and the number of words in each class

<table>
<tr><th colspan="2" rowspan="2">Original Labels</th><th colspan="5">Mappings</th></tr>
<tr><th>7 Classes</th><th>5 Classes</th><th>4 Classes</th><th>3 Classes</th><th>2 Classes</th></tr>
<tr><td>Joyful (strong)</td><td>113</td><td>113</td><td rowspan="2">805</td><td rowspan="3">884</td><td rowspan="6" colspan="2">11,491</td></tr>
<tr><td>Joyful (weak)</td><td>692</td><td>692</td></tr>
<tr><td>Surprised</td><td>79</td><td colspan="2">79</td></tr>
<tr><td colspan="2">Neutral</td><td colspan="3">9236</td></tr>
<tr><td>Pondering</td><td>848</td><td rowspan="2">1371</td><td colspan="2" rowspan="2">1371</td><td rowspan="4">2030</td></tr>
<tr><td>Hesitant</td><td>523</td></tr>
<tr><td>Angry (weak)</td><td>483</td><td>483</td><td colspan="2" rowspan="2">659</td><td rowspan="2">659</td></tr>
<tr><td>Angry (strong)</td><td>176</td><td>176</td></tr>
</table>

taking both speech and facial expression into account: During the annotation the labeler could observe the test person and hear the recorded utterances simultaneously. Then he was supposed to detect the changes in the user state of the test person and assign the corresponding label to the actual time interval (Schiel and Türk, 2006). The disadvantage of this method is obvious: If the observed user state change is judged only after the user's facial expression, gesture, or/and the situational context, there will be no prosodic equivalent of the assigned label, and this fact will in turn have negative influence on the classifier training and recognition results.

After the preprocessing of the annotations, we have one of the different user state labels assigned to each word. Thus, we apply only a word-based classifier as described in Sect. 2.5. In our experiments we used different mappings to get 2-, 3-, 4-, 5-, and 7-class problems. The original labels, the used class mappings, and the number of words in each class are shown in Table 2. The main problem for the automatic classification is a strong unequal distribution of the different classes. In case we want to train a classifier for a two-class problem, we have 659 words labeled as angry vs. 11,491 words with other labels. To get stable recognition results under such unfavorable conditions we conduct only *leave-one-speaker-out* (LOSO) tests on the whole dataset with neural networks (ANN) and *leave-one-out* (LOO) tests with *linear discriminant analysis* (LDA) from the SPSS software (Norusis, 1998). The results are shown in two last columns of Table 3.

Above we pointed out that there was only a holistic labeling available, taking into account both speech and facial expression at the same time: *u*ser *s*tates, *h*olistic (USH). In a second pass, other annotators labeled all nonneutral user states based purely on facial expressions: *u*ser *s*tates *f*acial (USF); details can be found in Batliner et al. (2003). For further classification experiments, we divided the whole database into two disjunct subsets, based on the agreement of USH and USF for the four "basic" user states *positive*, *neutral*, *hesitant*, and *angry*: *agreeing* (USH = USF) and *not agreeing* (USH $\neq$ USF). LOO classification was this time done with LDA and decision trees (J48) (Witten and Frank, 2000), and all 91 prosodic and 30 POS features. Recognition rates for *not agreeing* cases were lower, and for *agreeing* cases higher than for all cases taken together. For example, for seven classes, LDA yields a

Table 3. Classwise averaged recognition rates (in %) yielded in Lo(S)O tests on the WOZ data for five different class mappings. The results of two different classifiers, ANN and LDA, are given in last two columns

No. of classes	User states							Results	
								ANN	LDA
7	Joyful (strong)	Joyful (weak)	Surprised	Neutral	Hesitant	Angry (strong)	Angry (weak)	30.8	26.8
5	Joyful		Surprised	Neutral	Hesitant	Angry		36.3	34.2
4	Positive			Neutral	Hesitant	Angry		34.5	39.1
3	Positive			Neutral	Problem			42.7	45.5
2	Not angry					Angry		66.8	61.8

classwise averaged recognition rate of 29.7%, J48 47.5; for the two classes *not angry* vs. *angry*, the figures are 68.9% for LDA and 76.5% for J48. These results indicate a mutual reinforcement of the two modalities, which, in turn, leads to a more stable annotation if holistic and facial impressions converge.

3.3 Detection of user states on the Multimodal Emogram (MMEG) data

In experiments described below we use the data from the MMEG collection introduced in Adelhardt et al. (2006). In this dataset we have sentences labeled with one of four user states: *joyful*, *neutral*, *helpless*, and *angry* (Streit et al., 2006). From all collected speech data with good quality we choose randomly 4292 sentences for the training set and 556 for the test set (*test1*). Notice that we have here approximately 60% of sentences with the same syntactic structure in both training and test sets that were definitely dependent on the used user state. For example, all sentences built after the pattern "*I don't like <some TV genre>*" belong to the user state *angry*. To ensure that we really recognize user states and not the different syntactic structures of the sentences, we additionally select 1817 sentences for another test set (*test2*). The second test set contains only sentences with a syntactic structure independent of the labeled user state, for instance, sentences consisting of an isolated name of a TV genre or special expressions (Adelhardt et al., 2006). Thus, for this set, the syntactic structure of the sentence could not be used as a key to a proper recognition.

To train the classifier we had first to find out the optimal feature set. We tried different subset combinations of F0-based features, all prosodic features (91 set), and linguistic POS features in both context-dependent and independent form. In context-independent feature sets we used only the features computed for the word in question. For all configurations we trained the neural networks and tested them on the test sets (see the results of *test1* vs. *test2* in Table 4). The classwise averaged recognition rates for the four class problems (in percent) are shown in Table 4. We computed both word based and sentence-based recognition rates as indicated in the second column.

In Table 4 we notice that the POS features yield remarkable improvement only on the *test1* set; the results on the *test2* set get worse (see columns 3 and 5). That means they reflect to a great extent the sentence structure and therefore cannot be properly

Table 4. Classwise averaged recognition rates (in %) for four user states classified with ANN using five different feature sets. For each test set the wordwise and the sentencewise recognition rates are given

Test set	Type	Without context			With context	
		F0 feat. 12 feat.	All pros. 29 feat.	Pros.+POS 35 feat.	All pros. 91 feat.	Pros.+POS 121 feat.
Test1	Word	44.8	61.0	65.7	**72.1**	86.6
	Sentence	54.0	64.5	72.1	**75.4**	81.4
Test2	Word	36.9	46.8	46.5	**54.5**	52.7
	Sentence	39.8	47.5	48.1	**55.1**	54.2

Table 5. Confusion matrix of user state recognition with the ANN on the best feature set (91 features) using LOSO. Both the wordwise and sentencewise results are computed (in %)

Reference	Wordwise				Sentencewise			
User state	Neutral	Joyful	Angry	Hesitant	Neutral	Joyful	Angry	Hesitant
Neutral	**68.3**	12.5	12.6	6.6	**67.7**	12.0	16.5	3.8
Joyful	13.8	**65.8**	10.6	9.8	14.3	**66.4**	13.8	5.5
Angry	14.5	11.3	**64.7**	9.5	13.9	9.2	**70.8**	6.1
Hesitant	10.0	10.8	9.9	**69.3**	10.0	6.5	15.3	**68.2**

applied for the user state recognition in our case due to the construction of the corpus. The best results were achieved with the 91 prosodic feature set (75.4% *test1*, 55.1% *test2*, sentencewise). To verify these results with the speaker-independent tests, we additionally conducted a LOSO test using the 91 feature set. Here we achieved an average recognition rate of 67.0% wordwise and 68.3% sentencewise. The confusion matrix of this test is given in Table 5.

4 Conclusion

The prosody module used in the SMARTKOM demonstrator is based on the Verbmobil prosody module described in detail in Batliner et al. (2000a) and is extended with several new features. The module detects phrase accents, phrase boundaries, and rising intonation (prosodic marked queries) and includes new features concerning module communication, data synchronization, and a classifier for user state recognition. User state classification is done in two steps. In the first step we use word-based classification to compute a probability to assign one of several user states to each word. In the second step we process the probability of the whole utterance to decide one of the several classes.

For classification of prosodic events with the test set, we obtain a classwise averaged recognition rate of 77.1% for phrase accents, 88.5% for phrase boundaries, and 79.2% for rising intonation. For user state classification we collected our own data, due to the lack of training samples in WOZ data. Regarding the recognition of the user states, we noticed that POS features yield a remarkable improvement only

for the *test1* set containing sentences with the same syntactic structure as the training set. For the *test2* set, which contains only one-word sentences and special expressions, POS features have worsened rather than improved the results. Because of the construction of the MMEG database, the POS features reflect to a great extent the structure of sentence and thus cannot be properly applied to the user state recognition in this case. In a speaker-independent test for user state classification without POS features we achieved an average recognition rate of 67.0% wordwise and 68.3% sentencewise.

References

J. Adelhardt, C. Frank, E. Nöth, R.P. Shi, V. Zeißler, and H. Niemann. Multimodal Emogram, Data Collection and Presentation, 2006. In this volume.

J. Adelhardt, R.P. Shi, C. Frank, V. Zeißler, A. Batliner, E. Nöth, and H. Niemann. Multimodal User State Recognition in a Modern Dialogue System. In: *Proc. 26th German Conference on Artificial Intelligence (KI 03)*, pp. 591–605, Berlin Heidelberg New York, 2003. Springer.

N. Amir and S. Ron. Towards an Automatic Classification of Emotions in Speech. In: *Proc. ICSLP-98*, vol. 3, pp. 555–558, Sydney, Australia, 1998.

S. Arunachalam, D. Gould, E. Andersen, D. Byrd, and S. Narayanan. Politeness and Frustration Language in Child-Machine Interactions. In: *Proc. EUROSPEECH-01*, pp. 2675–2678, Aalborg, Denmark, September 2001.

A. Batliner, A. Buckow, H. Niemann, E. Nöth, and V. Warnke. The Prosody Module. In: W. Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 106–121, Berlin Heidelberg New York, 2000a. Springer.

A. Batliner, R. Huber, H. Niemann, E. Nöth, J. Spilker, and K. Fischer. The Recognition of Emotion. In: W. Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 122–130, Berlin Heidelberg New York, 2000b. Springer.

A. Batliner, M. Nutt, V. Warnke, E. Nöth, J. Buckow, R. Huber, and H. Niemann. Automatic Annotation and Classification of Phrase Accents in Spontaneous Speech. In: *Proc. EUROSPEECH-99*, vol. 1, pp. 519–522, Budapest, Hungary, 1999.

A. Batliner, V. Zeißler, C. Frank, J. Adelhardt, R.P. Shi, E. Nöth, and H. Niemann. We Are Not Amused – But How Do You Know? User States in a Multi-Modal Dialogue System. In: *Proc. EUROSPEECH-03*, vol. 1, pp. 733–736, Geneva, Switzerland, 2003.

F. Dellaert, T. Polzin, and A. Waibel. Recognizing Emotion in Speech. In: *Proc. ICSLP-96*, vol. 3, pp. 1970–1973, Philadelphia, PA, 1996.

G. Herzog and A. Ndiaye. Building Multimodal Dialogue Applications: System Integration in SmartKom, 2006. In this volume.

R. Huber. *Prosodisch-linguistische Klassifikation von Emotion*, vol. 8 of *Studien zur Mustererkennung*. Logos, Berlin, Germany, 2002.

R. Huber, E. Nöth, A. Batliner, A. Buckow, V. Warnke, and H. Niemann. You BEEP Machine – Emotion in Automatic Speech Understanding Systems. In: *TSD98*, pp. 223–228, Brno, Czech Republic, 1998.

A. Kießling. *Extraktion und Klassifikation prosodischer Merkmale in der automatischen Sprachverarbeitung*. Shaker, Aachen, Germany, 1997.

C.M. Lee, S.S. Narayanan, and R. Pieraccini. Combining Acoustic And Language Information For Emotion Recognition. In: *Proc. ICSLP-2002*, pp. 873–876, Denver, CO, 2002.

Y. Li and Y. Zhao. Recognizing Emotions in Speech Using Short-Term and Long-Term Features. In: *Proc. ICSLP-98*, vol. 6, pp. 2255–2258, Sydney, Australia, 1998.

M.J. Norusis. *SPSS 8.0 Guide to Data Analysis*. Prentice Hall, Upper Saddle River, NJ, 1998.

E. Nöth, A. Batliner, A. Kießling, R. Kompe, and H. Niemann. Verbmobil: The Use of Prosody in the Linguistic Components of a Speech Understanding System. *IEEE Transactions on Speech and Audio Processing*, 8(5):519–532, 2000.

V.A. Petrushin. Emotion Recognition in Speech Signal: Experimental Study, Development, and Application. In: *Proc. ICSLP-2000*, vol. IV, pp. 222–225, Beijing, China, 2000.

R.W. Picard (ed.). *Affective Computing*. MIT Press, Cambridge, MA, 1997.

M. Riedmiller and H. Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: *Proc. IEEE Intl. Conf. on Neural Networks*, pp. 586–591, San Francisco, CA, 1993.

F. Schiel and U. Türk. Wizard-of-Oz Recordings, 2006. In this volume.

E.G. Schukat-Talamazzini. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Braunschweig, Germany, 1995.

M. Streit, A. Batliner, and T. Portele. Emotion Analysis and Emotion Handling Subdialogs, 2006. In this volume.

I.H. Witten and E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques With Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000.

V. Zeißler, E. Nöth, and G. Stemmer. Parametrische Modellierung von Dauer und Energie prosodischer Einheiten. In: *Proc. KONVENS 2002*, pp. 177–183, Saarbruecken, Germany, 2002.

The Sense of Vision: Gestures and Real Objects

Jens Racky, Michael Lützeler, and Hans Röttger

Siemens AG, Human Computer Interaction, München, Germany
{Jens.Racky,Michael.Luetzeler,Hans.Roettger}@siemens.com

Summary. Natural human communication is based on visual and acoustical signals. Thus a technical system needs the same senses to allow intuitive interaction. The acoustical interface has been described in earlier chapters. In this chapter we concentrate on the perception of conscious visual utterances, for example, hand gestures and handling of dedicated (real) objects, e.g., paper documents.

1 Gesture Recognition

The gesture recognition modules supply the SMARTKOM system with video-based interaction capabilities. The users pointing gestures are captured by a downward looking video camera. Based on the well-known Siemens SiViT technology (Maggioni and Röttger, 1999) the position of the user's fingertips is determined and transformed to screen coordinates. In a second step the individual measurements are aggregated to gesture strokes that provide deeper insight into the user's intentions.

1.1 Introduction

Gestures play an important role in face-to-face interaction between humans. In conjunction with speech it forms the major part of information transfer. Gestures are an easy to use and natural way of interaction. Additionally, pointing gestures and speech disambiguate each other. A speech phrase "I want this" and an appropriate pointing gesture provide a very efficient, multimodal approach to information transfer.

Capturing pointing gestures by a video-based interaction system can provide an intuitive and toolsfree input modality. To meet this goal several requirements have to be met:

- **Real time:** The evaluation of the video input must be performed in real time to give the user direct feedback of the recognized intention.
- **Robust:** Recognition has to be be robust, especially under video projection, e.g., when the graphical user interface is projected by a beamer.

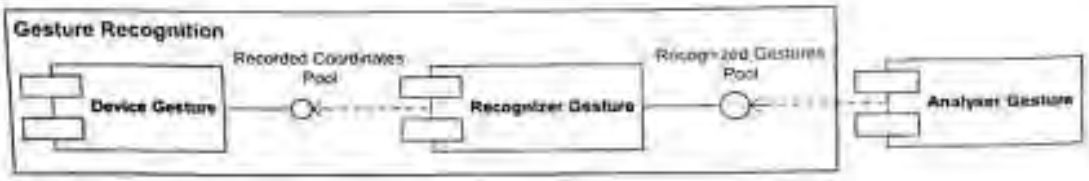

Fig. 1. Gesture recognition modules

- **Intuitive:** The system has to be adequate to the natural gesture vocabulary of the users, e.g., the users should be able to use their natural hand position and not be required to learn any specific hand position.

1.2 System Integration

A module for capturing pointing gestures can be seen as an abstract pointing device, similar to a computer mouse, a pen or a touch screen. These devices supply streams of single-coordinate readings. Individual measurements are then aggregated to gesture stokes to provide higher levels information. To treat different input devices, used in the SMARTKOM scenarios (`device.pen`, `device.gesture`, `device.mouse`) alike, an abstract representation for pointing devices was defined. Each input device supplies coordinates to the pool `recorded.coordinates` (see Fig. 1). The gesture recognition module reads these coordinates and identifies gesture strokes within a given vocabulary. Recognized gesture strokes are stored in the pool `recognized.gestures`, which provides the interface to the gesture analysis module for media fusion. This module correlates the recognized gestures with the current content on the display (Engel and Pfleger, 2006).

1.3 Gesture Vocabulary Definition

Classical gesture recognition, e.g., as provided by the SiViT technology, determines static gestures like a selection, when the user points at some location for a certain time span. As a substantial enhancement in this contribution the notion of *dynamic gestures* is introduced. Dynamic gestures result from motions the user performs while expressing a pointing gesture. A typical example for a dynamic gesture is encircling, a group of objects in order to select them.

Besides *tarrying* over a displayed object for selection and *encircling* the Wizard-of-Oz experiments accompanying the SMARTKOM project show that users tend to strike through an object to select it. A *linear stroke* is also used as supporting gesture while reading or searching in lists.

Thus a gesture vocabulary meeting the requirements of proper adaption to the natural gestures performed by users and taking into account the capabilities of the input device can be established on the basis of three classes (*tarrying*, *linear stroke* and *encircling*) and a class of undefined gestures. This vocabulary can be modeled by these geometric primitives: point, line and ellipse.

1.4 Interfaces

To support the above-defined abstraction of pointing input devices and the introduced vocabulary proper XML structures have been defined.

A pointing device supplies `coordinateData` to the pool `recorded.coordinates`. In Fig. 2 an example of a coordinate lattice is shown. Multiple single-coordinate recordings (`coordinateEdge`) are joined into one `coordinateData` structure. The information of the position reading is augmented by the event type, if applicable the pressure supplied (e.g., for pen input) and the confidence, modeled as probability, that a pointing event has been successfully detected.

```
<coordinateData>
  <source>          gesture                        </source>
  <sampleRate>      25                           </sampleRate>
  <startTime>       2001-09-25T15:25:33.002     </startTime>
  <endTime>         2001-09-25T15:25:33.953       </endTime>
  <numberOfSamples> 25                       </numberOfSamples>
  <coordinateEdge>
    <recordedTime>  2001-09-25T15:25:33.082  </recordedTime>
    <screenPosition>
      <x>                      0.604022                  </x>
      <y>                      0.770749                  </y>
    </screenPosition>
    <pressure>                 0                    </pressure>
    <type>                     pointing                 </type>
    <probabilityPointing>      1       </probabilityPointing>
    <probabilityUndefined>     0     </probabilityUndefined>
    <buttonDown>               false            </buttonDown>
  </coordinateEdge>
  . . .
</coordinateData>
```

Fig. 2. Example recorded coordinates

The `recognizer.gesture` takes measurements from any of the sources given in Sect. 1.2 and assembles them to meaningful gesture strokes. The data structure for a gesture stoke consists of a number of coordinate readings and the gesture primitive from the vocabulary defined in Sect. 1.3. The primitive is coded in the `<type>` structure. For encircling gestures the elliptical model is approximated by a bounding box, which proved to be sufficient information for the media fusion in the `analyzer.Gesture`. Figure 3 shows an example of a gesture stoke lattice. The lattice may contain multiple elements of the `gestureStroke` structure.

1.5 Recognition Process

The recognition process is dominated by the sequential nature of the input data. For a timely response, a geometric gesture primitive should be reported as early

```
<gestureStrokeLattice>
  <gestureStrokeCount>  1                  </gestureStrokeCount>
  <gestureStroke>
    <startTime>         2001-09-25T15:25:33.372 </startTime>
    <endTime>           2001-09-25T15:25:33.953   </endTime>
    <positionCount>     16                     </positionCount>
    <position>
      <x>               0.593003                          </x>
      <y>               0.354864                          </y>
    </position>
    . . .
    <type>              encircling                     </type>
    <probability>       0.8                     </probability>
    <geometricPrimitive>
      <probability>     0.8                     </probability>
      <boundingBox>
        <topLeft>
          <x>           0.578558                          </x>
          <y>           0.289846                          </y>
        </topLeft>
        <bottomRight>
          <x>           0.899767                          </x>
          <y>           0.692392                          </y>
        </bottomRight>
        <variance>      1.56025e-007              </variance>
      </boundingBox>
    </geometricPrimitive>
  </gestureStroke>
</gestureStrokeLattice>
```

Fig. 3. Example gesture lattice

as possible and be extended if more measurements arrive that fit into the current geometric model. Two distinct modes of operation can be separated.

In the first mode no gesture stroke has been identified yet. Measurements are added to the hypothesis for the gesture primitives in parallel. A number of heuristics are applied to verify each hypothesis. They can be described for the three classes by:

- **Tarrying:** Are the singular measurement points spatially and temporally sufficiently close to each other and is there a sufficient number of data readings?
- **Linear-stroke:** Does a straight line approximation fit the given data well?
- **Encircling:** Can the stroke be approximated by fitting an ellipse to the data-points?

If one of these hypotheses has been verified, the gesture stroke is passed to the output pool. Afterwards the second mode is entered. There the question is: Does the new measurement extend the current gesture stroke or does it initiate a new stroke?

Fig. 4. Encircling gesture for cinema reservation

Experimental results within the SMARTKOM framework have demonstrated the robustness and suitability of the chosen approach. Gesture recognition is performed at 50 Hz on 320- by 240- resolution video images. Users accepted gesture input and used it intuitively. The gesture recognition modules are used in all three SMARTKOM scenarios (Public, Home, Mobil). Figure 4 shows the user's pointing gestures in the "cinema reservation" dialogue, when indicating which seats should be reserved. Another example for easy data input by pointing gestures is entering a phone number on a projected keypad for the public scenario.

2 Handling Real Objects

Besides video-based gesture recognition, yet another video-based approch is used to integrate real objects into the SMARTKOM world. By real objects we denote the class of those physical items by which a user may interact with SMARTKOM. Within the context of the SMARTKOM-Public case, these are objects whose visual appearance (or informational contents) shall be transmitted, e.g., *paper documents* (to be shown on the display, to be transmitted by e-mail or by fax) and also *hands* (for biometric authentication). In this section we give an overview of the functionalities of the module *device.realObjects* as well as implementation issues.

2.1 Interface to the SmartKom Testbed

The module provides three complementary interfaces:

1. The general XML-based interface by which it communicates with the other modules in the system

2. A private graphical user interface (GUI) for adapting the module to the current setup, e.g.,
   - Set input video amplifier parameters (brightness, contrast, saturation, hue)
   - Set threshold for contour detection
   - Set acquisition region for documents
   - Set *pan, tilt, zoom* range for panoramic image acquisition
   - Have the recorded images rotated to allow for camera mounting flexibility
3. A GUI controlled by the testbed to set the current working parameters, e.g., activate (automatic) document detection, chose between simple and panoramic imaging, select binarization method for fax document generation

2.2 XML Interface

The XML interface[1] consists of three parts:

- The *control/command* interface
- The *State* interface (Sect. 2.2.2)
- The *Data* interface (Sect. 2.2.3)

An element called *documentID* is included in most data structures, as a reference to recorded real objects. This is necessary as recorded object may be referenced several times (e.g., send as e-mail, then send by fax) and the request–response mechanism of the SMARTKOM testbed is not suitable for this purpose. Implementation was done in C++ by creating a direct mapping of the XML types to C++ objects. In particular, XML derivation could be mapped to C++ derivation.

2.2.1 Command Interface

All commands to the module are derived from `RecognizerRealObjectsCommand`. This allows us to submit a sequence of commands to the module within one package. Furthermore, implementation of a command queue is straightforward when using C++ with STL containers.

- `RecognizerRealObjectsCommands` = *recognizerRealObjectsCommand*+ (`RecognizerRealObjectsCommand`)
  Abstract container for a sequence of control commands.
- `RecognizerRealObjectsCommand`
  Abstract control command for the device.realObjects recognizer.
- `GetRecognizerRealObjectsStatus` ⟶ `RecognizerRealObjectsCommand`
  Request state information.
- `ScanDocument` ⟶ `RecognizerRealObjectsCommand` = *documentID* (`NonNegativeInteger`),
  Scan real document into internal format.

[1] In the following section we use "⟶" to denote derivation, "+" to denote "one or more" and "{" "}" to denote choice of one element.

- `TranscribeImage` ⟶ `RecognizerRealObjectsCommand` =
  documentID (`NonNegativeInteger`),
  imageCodingType (`ImageCodingType`),
  imageSize (`ImageSize`),
  imageColorType (`ImageColorType`)

  Transcribe image with id *documentID* to representation of type *imageCodingType*. Optionally, the image's size and color type are converted.

- `ImageSize` =
  width (`NonNegativeInteger`),
  height (`NonNegativeInteger`)

  Size of the image in number of pixels.

- `ImageColorType` ⟶ `String` = { "Gray", "Color" }
  Color type of the image, either 24B-bit RGB or 8-bit gray.

- `ImageCodingType` ⟶ `String` = { "SFF", "PNG", "HCT" }
  Coding type of the image data, either Portable Network Graphic (PNG) or Structured Fax File (SFF) format. HCT is a proprietary format for transmitting the contour of a hand for biometric hand contour authentication (Grashey and Schuster, 2006).

2.2.2 State Interface

This interface is used to report the current module state, which is not only the state of the internal state machine (Sect. 2.3), but also information about the command received last.

- `TranscribeImage` ⟶ `RecognizerRealObjectsCommand` =
  documentID (`NonNegativeInteger`),
  recognizerState (`RecognizerState`),
  commandID (`NonNegativeInteger`),
  message (`RealObjectsMessage`),
  imageCodingType (`ImageCodingType`),
  imageSize (`ImageSize`),
  imageColorType (`ImageColorType`),

- `RecognizerState` ⟶ `String` = { "init", "scanning", "scanningHand", "scanned", "transcribing", "transcribed" }

  (State machine) state of the module (see Sect. 2.3).

2.2.3 Data Interface

This is the container for transmitting (binary) coded object data within the SMART-KOM system.

- `RealObjectRepresentation` =
  documentID (`NonNegativeInteger`),

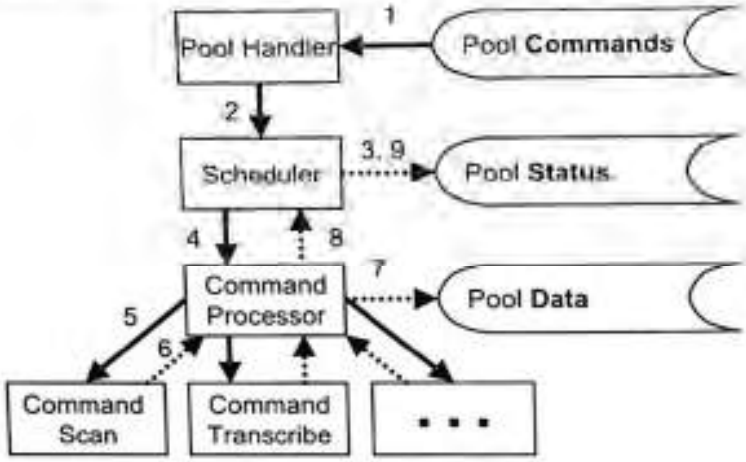

Fig. 5. General structure of Recognizer.Realdocument

This a base class for object representations. Currently only imagelike representations are used.

- `RealObjectImage` =
  imageCodingType (`ImageCodingType`),
  imageSize (`ImageSize`),
  imageColorType (`ImageColorType`),
  codedImageData (`Binary`),

  Binary (i.e., base-64 coded) image representation of the real object (according to *imageCodingType*). As all current coding formats are (originally) file based, the `Binary` data type was implemented with a stream interface.

2.3 General Module Structure

The general structure of the module is shown in Fig. 5. It communicates with the SMARTKOM system via three pools, i.e., a pool for commands, a pool for results (data) and a pool for telling its state. The latter ones are kept separately as not all of the module's subscribers need both kinds of information. The process flow on executing a command consists of the following steps. The step numbers correspond to those in Fig. 5, demonstrating a typical data flow:

1. The command is received from the pool by the pool handler.
2. The decoded XML information is a ready-to-use command object, which is added to the command execution queue of the command scheduler.
3. The scheduler sends a status message indicating the start of command processing to the status pool.
4. The scheduler computes the next internal command (i.e., target state) resulting from the external command and the current state, according to Fig. 6 (e.g., if a `ScanDocument` command has been received, the first internal command will be a `WaitForDocument`) and sends it to the processor for execution.

5. The processor then calls the concrete command implementation, which processes the data, e.g., converts a previously recorded image to the desired output format.
6. The result is written by the processor to the data pool, and
7. command completion is signaled to the scheduler,
8. which generates an appropriate status message for the status pool.
9. If the new state (now reached) is not stable, the next state is computed, continuing with step 4.

2.4 Object Detection

Detection of the presence of the real object to be acquired is a fundamental feature of the module. The implementation utilizes the same set up as gesture recognition, i.e., an IR camera–based approach resulting in a robust contour-based image segmentation.

A *real object* is considered to be present, if there is only one object, and it meets the following criteria:

- Contour and area exceed a given size/length.
- The contour does not hit the recording area's boundary.
- The objects dimensions do not vary too much for a given time

2.5 Image Acquisition

Real objects are recoded via a PAL pan-tilt camera (Sony EVI-D30) connected by an S-VHS link and controlled through RS232. If not recording in panoramic mode (see next section) the resulting image to be encoded is just the "raw" camera image, optionally rotated to compensate a camera mounting angle.

Fig. 6. Moore-type state machine of Recognizer.Realdocument. Stable states are denoted by a *nonfilled circle*; transient states by a *filled circle*

2.5.1 Panoramic Imaging

When operating in panoramic mode, the image to be transmitted is assembled from multiple images, recorded with a given overlap, as shown in Fig. 7.

The panoramic image recording consists of the following steps:

1. Recording of the image sequence. This is done by moving the camera to a given number of intermediate/interpolated positions between the limiting positions.
2. Estimation of the (camera) transformation between consecutive images of the sequence. We investigated two different approaches:
   - Tracking of a set of feature points followed by either computation of a translation vector or a linear transformation, as described, e.g., in Bhat et al. (2000): Let $\mathbf{x} = (x, y)$ be a pixel in an image I that translates to $\mathbf{x}' = (x + t_x, y + t_y)$ in the subsequent image J. The affine transformation between I and J is described by

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} ax + by + f \\ cx + dy + h \end{bmatrix},$$

where $\mathbf{x}'$ is the coordinate of one (feature) point in image J, and $\mathbf{x}$ is its corresponding feature point in image I. The feature points are selected by the

Fig. 7. Example of a (vertical) panoramic image creation. Recording of five overlapping images (scheme, *top left*) last (bottom) image (*bottom left*), resulting panoramic image (*right*)

algorithm from Shi and Tomasi (1994). Tracking is done by the algorithm described in Lucas and Kanade (1981) and Bouguet (1999), followed by an outlier removal based on variance and median filtering.
The approach showed good results and could be computed in real time if motion blur was low. Unfortunately the angular speed of the EVI-D30 depends on the current zoom, which leads to total motion blur at the required zoom level. The linear mapping was adequate for the given setup, but was very sensitive against erroneously detected feature points.

- Translation only: As the angular range within the SMARTKOM-Public setup is low, it was possible to use a correlation-based approach to robustly and sufficiently compute the translation. Figure 6 shows an exampled for this approach.

3. Stitching, i.e., assembling pieces of the recorded images to form the panoramic image. We distinguish two concepts: *Patching* uses exclusively the pixel information of one image to set a pixel value within the panoramic image and *blending* uses the pixel information from (some or all) overlapping pixels to compute the final value for the panoramic pixel. It has been found that for the SMARTKOM-Public setup (which utilizes an artificial illumination via the video projector) the first approach is sufficient, as can be seen from the example.

2.6 Preparation for Image Transmission

Image data can be transmitted via to ways: as an email attachment or as an ISDN fax. The main difference is that the chosen ISDN/fax data format provides only a black-and-white mode and requires the specific image width of exactly 1728 pixels.

2.6.1 Transmission by E-Mail

The Portable Network Graphics (PNG) format was chosen as a container to transport image data via e-mail. The system uses the free PNG library written in C (http://www.libpng.org/pub/png/) to implement the image coding. To enable data transport via the XML interface, a C++ stream wrapper was implemented. Images are always encoded as recorded, i.e., besides the optional camera rotation compensation no further processing takes place.

2.6.2 Transmission by ISDN–Fax

In SFF format images are run-length encoded black-and-white (b&w) bitmaps with a line width of 1728 (the format allows arbitrary line lengths, but it has been found that at least some G3 fax machines will not output the image if the line length differs from 1728). Our implementation features a stream-based interface, thus the implementation of the XML `Binary` type can be interfaced directly. As the recorded image does not meet these requirements, it is first converted to 8-bit gray level. Afterwards it is scaled up by an interpolating algorithm described in Rieg (1994), which has been stripped down to the gray-scale case. Finally, the color depth is reduced to b&w.

Color to B&W Conversion

First, some well-known dithering algorithms, like Floyd–Steinberg error diffusion were implemented. While the results are good as expected, the run-length encoding is not very efficient on the dithered data. Thus the resulting SFF file is very large, leading to very long transmission times via ISDN, which even causes some G3 fax devices to produce errors. In particular, the following algorithms were implemented, from Ulichney (1987):

- Floyd–Steinberg error diffusion
- Bayer's ordered dither with a 16×16 matrix
- Three sizes of 45-degree clustered-dot dither (3×3, 4×4, 8×8)

As a consequence, we investigated segmentation techniques, e.g., Otsu's algorithm, to perform the binarization. In particular, the following algorithms were implemented:

- Bipartite histogram partitioning minimizing intra- and interclass variance, according to Otsu (1979).
- Thresholding surface from a potential surface passing through the image at local maxima of the gradient image, according to Yanowitz and Bruckstein (1989).

Because of the present hardware setup (defined illumination), Otsu's algorithm shows good results, as long as the real object does not contain too much shading (e.g., a sheet of white paper with black text will be segmented well). As this is not always the case, we decided to use Yanowitz and Bruckstein's method by default. This proved to be a good compromise between the amount of data and the image quality. The methods described here have been applied successfully in the SMART-KOM-Public scenario to capture and transfer images of real objects and to enable hand-contour verification.

References

K.S. Bhat, S. Mahesh, and K. Pradeep. Motion Detection and Segmentation With Pan-Tilt Cameras Using Image Mosaics. *IEEE Int. Conf. on Multimedia and Expo*, 3:1577–1580, 2000.

J.Y. Bouguet. *Pyramidal Implementation of the Lucas Kanade Feature Tracker – Description of the Algorithm*. OpenCV Documentation, Intel Corporation, Microprocessor Research Lab, 1999.

R. Engel and N. Pfleger. Modality Fusion, 2006. In this volume.

S. Grashey and M. Schuster. Multiple Biometrics, 2006. In this volume.

B.D. Lucas and T. Kanade. An Iterative Image Registration Technique With an Application to Stereo Vision. In: *Proc. 7th IJCAI*, Vancouver, Canada, 1981.

C. Maggioni and H. Röttger. Virtual Touchscreen – A Novel User Interface Made of Light — Principles, Metaphors and Experiences. In: *Proc. 8th Intl. Conf. on Human-Computer Interaction*, pp. 301–305, Munich, Germany, 1999.

N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions Systems, Man and Cybernetics*, 9(1):62–66, 1979.

M. Rieg. Superelastisch verlustarmes Skalieren von Bitmap-Grafiken. *c't 11/94*, p. 302, 1994.

J. Shi and C. Tomasi. Good Features to Track. In: *Proc. Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593–600, Seattle, WA, June 1994. IEEE Computer Society.

R. Ulichney. *Digital Halftoning*. MIT Press, Cambridge, MA, 1987.

S.D. Yanowitz and A.M. Bruckstein. A New Method for Image Segmentation. *Computer Vision, Graphics and Image Processing*, 46(1):82–95, April 1989.

The Facial Expression Module

Carmen Frank, Johann Adelhardt, Anton Batliner, Elmar Nöth, Rui Ping Shi, Viktor Zeißler, Heinrich Niemann

Friedrich-Alexander Universität Erlangen-Nürnberg, Germany
{frank,adelhardt,batliner,noeth,shi,zeissler,
niemann}@informatik.uni-erlangen.de,

Summary. In current dialogue systems the use of speech as an input modality is common. But this modality is only one of those human beings use. In human–human interaction people use gestures to point or facial expressions to show their moods as well. To give modern systems a chance to read information from all modalities used by humans, these systems must have multimodal user interfaces. The SMARTKOM system has such a multimodal interface that analyzes facial expression, speech and gesture simultaneously. Here we present the module that fulfills the task of facial expression analysis in order to identify the internal state of a user.

In the following we first describe the state of the art in emotion and user state recognition by analyzing facial expressions. Next, we describe the *facial expression recognition module*. After that we present the experiments and results for recognition of user states. We summarize our results in the last section.

1 State of the Art

The knowledge about a user's facial expression can be used, e.g., to recognize that a helpless user needs some help or to adapt presented information in case of confusion. The systems dealing with classification of facial images, which were developed over the past years, can be differentiated according to the following characteristics:

- sequence or single image
- recognizable classes
- used features
- used classifier

For the field of **single images**, the decision for a class is done with respect to one single image. Methods using single images can be found in Wiskott (1997); Chen and Huang (2002), Kumar and Poggio (2000), and Thomaz et al. (2001). **Images sequences** are used in Kaiser et al. (1998), Essa and Pentland (1997),and Müller et al. (2001). These methods and similar ones use the variation over time, for instance the optical flow.

Some systems cannot be associated with single-image or image-sequence methods. Samples of such hybrid systems can be found in Schwerdt et al. (2000), Lien

et al. (1998a), and Tian et al. (2001). The first system classifies each frame of a sequence, and the results form a trajectory of changes in the face, which is classified using an eigenspace. Both other systems use a *neutral* image to find relative changes to the current one. This comparison allows users to determine the intensity of the facial expression.

Depending on the field of application, a facial expression recognition system must have a special **category set**, to which the faces should be classified. Very common is the concept of six basic emotions (happiness, sadness, anger, fear, disgust and surprise) used by Ekman (Lyons et al., 1999; Thomaz et al., 2001; Otsuka and Ohya, 1997). Subsets of these emotions are also used. Often there is one additional expression, "raised eyebrows" (Müller et al., 2001; Schwerdt et al., 2000; Essa and Pentland, 1995).

Besides these complex categories, there are several methods which use action units (AUs) (Ekman and Friesen, 1978) as a category set: Lien et al. (1998a), Tian et al. (2000), Tian et al. (2001), Otsuka and Ohya (1999), and Braathen et al. (2001). AUs are the smallest units in facial movements similar to phonemes for speech. The major interest to the AUs arises from their relevance in field of psychology. Because AUs are a standard technique for coding facial expressions, automatic coding would provide a much larger set of data for further experiments in psychology.

There are only a few papers concerning classification of user states that are not part of the basic emotions. Most systems that detect states necessary for a human computer interaction (attention, point of interest, stress, fatigue, etc.) use touch sensors. One example for the recognition of fatigue from an image sequence is shown in Li et al. (1999).

AUs are not only used as a category system, but also as **features**. Some systems using AUs as features are explained in Lien et al. (1998b), Otsuka and Ohya (1999), and Cohn et al. (1998).

For feature extraction to expression classification, one can distinguish between template-based and analytic methods. Template-based methods match a face model to a face. Then positions of facial features are extracted from this model, or muscle movements are derived. Template-based methods use graphs (Wiskott, 1997), models of facial features, e.g., the mouth (Oliver et al., 1997; Tian et al., 2000), or work with hand-segmented facial features. Essa explains (Essa and Pentland, 1997) a two-step system by matching a triangle mesh, using optical flow (Simoncelli, 1993) to the face and matching the result to a muscle model of the face.

In feature-based methods one can often find systems using wavelets. Two common types of wavelets are Haar wavelets (Kumar and Poggio, 2000) and Gabor wavelets (Lyons et al., 1999; Hu et al., 2002).

The systems for facial expression **classification** found in literature differ not only in the used features but also in the used classifier. They can be subdivided into model-, rule- and neural network–based methods. As model-based methods, principal component analysis, PCA (Schwerdt et al., 2000; Thomaz et al., 2001; Kirby and Sirovich, 1990) or support vector machines SVM (Braathen et al., 2001) are used. If the classification is done using image sequences and not only single images, the information of the single frames can be combined. For this task Hidden

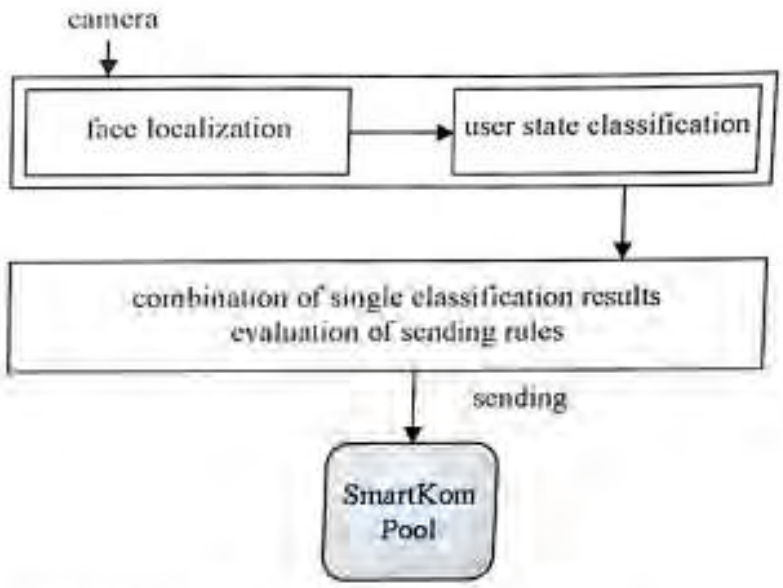

Fig. 1. The general architecture of the facial expression module

Markov Models are well known (Otsuka and Ohya, 1997; Müller et al., 2001; Oliver et al., 1997; Lien et al., 1998a). Rule-based methods include rules to map features to classes. In Chen and Huang (2002) the results from different classification stages are fused using handmade rules. An expert system using AUs for classifying facial expressions is explained in Pantic (2001). Methods based on neural nets can be found in Tian et al. (2000), Hu et al. (2002), and Zhang et al. (1998).

Furthermore, methods and systems for facial expression classification can be differentiated using facts like

- the use of handmade markers in the face
- the initialization of a sequence by hand
- the selection of features by hand
- person dependence or independence

The presented facial expression module uses a localization technique based on color because this is a feature common to all faces. As classifiers, eigenspaces PCA and SVM are used. Both allow the use of pixel values as features instead of segmented facial features such as eyebrows and lip corners. The motivation for not hand selecting features is the success of probabilistic, segmentation-free methods for spontaneous speech recognition. Facial expressions during a multimodal dialogue are equivalent to spontaneous utterances.

2 Module Description

The facial expression module consists of three main parts: **localization, classification**, and **combination and sending** of data (Fig. 1) in order to fulfill the task of

Fig. 2. Original image (*left*), result when skin color segmentation is done (*right*). The image parts with blue and green color shades disappear, but white shades and natural colors stay

facial expression recognition as part of a multimodal dialogue system. They are explained in more detail in the following sections.

The facial expression recognition does not try to detect the basic emotions like *disgust*, etc., but tries to detect user states. User states are all these internal states of a user that influence his or her interaction with the automatic system. For an information system similar to SMARTKOM , user states like *hesitant* or *angry* are suitable.

2.1 Localization

The localization of a face in the SMARTKOM environment has several constraints. There is at most one person using SMARTKOM , but this person is unknown to the system. The background is dynamic and is also unknown. It is not possible to use motion information to locate a person. For this reason we decided to use facial color as a first step in person localization.

Because skin color is not very clearly defined in the RGB color space, we use the YUV color space. In Fig. 2 one can see the results of a skin color segmentation. A nontypical image is used for better illustration: nonfacial skin color can be found as well. It is possible to eliminate "cold" colors. The resulting pixels are used as hypotheses for the further calculations.

To find the real position of the face, the facial expression module uses a holistic classifier and a filter. The holistic classifier can be a SVM, see Osuna et al. (1996) or PCA, see Turk and Pentland (1991a).

A problem when separating faces from "nonfaces" is that a nonface is not clearly defined. It is not possible to collect a set of all kinds of nonfaces. Therefore, the holistic classifiers sometimes select an area in an image as a face which is, for a human observer, clearly not a face. This problem can be reduced by a filter. We use a bank of circle frequency (CF) filters (Kawato and Tetsutani, 2002). The CF filter tries to find an area with alternation of light and dark values, which can be found near the between-eyes point (dark eyes, light forehead and nose). The effects of this filter are illustrated in Fig. 3.

Fig. 3. The *left* face is the original face used to calculate the CF filter. The *middle* image shows the pixels with high weighting in light values. For a better illustration, both images are combined (right)

2.2 Classification

After the position of the face/the person is localized, the user state of this person can be classified. The classification is done by holistic classifiers as well. The method proposed by us for the recognition of facial expressions is a modification of a standard eigenspace classification for user identification. Eigenspace methods are well known in the field of face recognition (Turk and Pentland, 1991b; Yambor et al., 2000; Moghaddam and Pentland, 1994). In a standard face recognition system, one eigenspace for each person is created using different face images. The set of face images for each person is used to create a probability distribution or a representative for this person in the face space. Later, when classifying an image of an unknown person, this image is projected onto the face spaces. The probability distribution or representative that best matches the new image is chosen as the class in question.

To create an eigenspace with training images, a partial Karhunen–Loève transformation, also called principal component analysis (PCA), is used. This transformation is a dimensionality reduction scheme that maximizes the scatter of all projected samples, using N sample images of a person $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with values in an n-dimensional feature space. Let μ be the mean image of all feature vectors. The total scatter matrix is then defined as

$$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T , \tag{1}$$

In PCA, the optimal projection W_{opt} to a lower-dimensional subspace is chosen to maximize the determinant of the total scatter matrix of the projected samples

$$W_{opt} = \arg\max_{W} |W^T S_T W| = [w_1, w_2, \ldots, w_m] , \tag{2}$$

where $\{w_i | i = 1, 2, \ldots m\}$ is the set of n-dimensional eigenvectors of S_T corresponding to the set of decreasing eigenvalues. These eigenvectors have the same dimension as the input vectors and are referred to as "eigenfaces". In Fig. 4 the first four eigenfaces of the *angry* eigenspace are shown. In the following we assume that high-order eigenvectors correspond to high eigenvalues. Therefore, high-order eigenvectors contain more relevant information.

Fig. 4. The *left* image is the average image; the following images are the first four eigenvectors of the *angry* eigenspace (Frank and Nöth, 2001)

For face classification a new image is projected to each eigenspace and the eigenspace which best describes the input image is selected. This is accomplished by calculating the residual description error.

Imagine we have training sets F_κ of l samples $\mathbf{y}_i$ with similar characteristics for each class Ω_κ, $\kappa \in 1, \ldots, k$. There is different illumination, different face shape, etc., in each set F_κ. Reconstructing one image $\mathbf{y}_i$ with each of our eigenspaces results in k different samples $\mathbf{y}^\kappa$. The reconstructed images do not differ in characteristics like illumination, because this is modeled by each eigenspace. But they differ in the facial expression of specific regions, such as the mouth or the eyes area.

With a set of eigenspaces for each class Ω_κ we receive distances ν_κ of a test image $\mathbf{y}_i$ to each class

$$\nu_\kappa = ||\mathbf{y}_i - \mathbf{y}^\kappa||^2 \tag{3}$$

$$k = \arg \min_{j \in 0..k} \nu_\kappa. \tag{4}$$

An image is attributed to a class k with minimum distance as criterion.

In addition to the modeling of separate models for each user state, we use only a subset of the available pixels, i.e., only "significant" pixels; the extraction of this subset is described in Frank and Nöth (2003). This method creates a map of significancy of pixels and allows us to delete less significant ones. In Fig. 5 one can see such a map (leftmost image). Only pixels significant for facial expression (mouth and eye areas) are selected as features for the holistic classifier.

2.3 Combination and sending

The classification step produces results for every captured frame, but a facial expression has a longer duration than a single frame. Normally, it starts with a neutral expression, then the expression of the user state evolves and goes back to the neutral expression. Therefore, the single classification results are combined in order to send fewer messages to the SMARTKOM pool.

There are some rules to decide whether a message should be sent or not:

1. A new user state is detected.
2. No new user state has been detected for a long time.
3. The intensity of the recent user state changed significantly.

Fig. 5. The *leftmost* image shows the map of significance for a *smile–anger* classification. Different thresholds are used to create the other images (white pixels represent significant ones)

The second rule is necessary because the system needs to know that nothing happened, but the facial expression module is still "alive" and keeps watching the user. The values for *long time* and *significantly* used in rules 2 and 3 can be set using the GUI of the module. This is necessary because the amount of data processed by the facial expression module depends on the computational resources needed by other modules.

The sent data are a combination of single classification resulting from the last analyzed frames. The older a single classification result is, the lower is its influence to the overall result.

3 Experiments

For examination of the proposed classification methods we need a special dataset, including video sequences and corresponding user state labels. In the WOZ dataset of the SMARTKOM project (Schiel and Türk, 2006), one can only find a very small subset showing expressive user state. Therefore, we collected our own multimodal dataset with acted user states (Adelhardt et al., 2006). For the following experiments we used sequences of ten persons with, for a human being, recognizable expressions, in a leave-one-out manner. The whole face is used to create four eigenspaces for four facial expression classes. When a naive user tried to express an utterance in an angry manner, he/she often displayed neutral facial expressions before and after speaking, thus not every image of a sequence shows the same user state. The same is true for the other user states. The labeling of each single image is not possible due to the complexity. A consequence of that is a low recognition rate for neutral, because each class-specific dataset includes neutral faces. The classification of faces with internal movements, produced due to speaking, is very difficult. We achieve a low recognition rate of 32%. The confusion matrix is shown in Table 1. A problem is the user state *angry*. Anger a facial expression that is shown in many different ways by different users, whereas joy is consistently marked by risen lip corners.

The same procedure applied to a dataset (presented in Martinez and Benavente (1998)) of mugshots, showing four different facial expressions, yields 59%. The high recognition rate of the still images are due to reasons given above.

Table 1. Confusion matrix of user state recognition with facial expression data (%)

Reference user state	Results			
	Neutral	Joyful	Angry	Hesitant
Neutral	**7**	23	36	33
Joyful	5	**54**	22	20
Angry	4	62	**17**	16
Hesitant	6	12	35	**48**

One idea to solve the latter problem is the elimination of face regions influenced by speaking. To find out which part of a face is able to identify which facial expression, we performed experiments using only parts of a face for classification.

For this task it is necessary to know the position of eyes, nose and mouth and to have full-blown facial expression. Therefore, we use the database presented in Martinez and Benavente (1998). So the first-mentioned problem (not every frame shows an expression) does not influence our results and we can concentrate on the face regions.

To classify face regions, the faces are split into three parts (eye, nose, and moutharea). Samples of these three regions can be seen in Fig. 6. When only using one facial region for expression classification, the region centered at the mouth gives the best results (79%). The eye area achieves 63%, and the nose area 65%. The confusion matrices for all three face regions are shown in Table 2. It is noticeable that the different regions have different capabilities in indentifying different facial expressions. The mouth area gives really good results identifying *smile* and *shout* expressions, but *neutral* and *anger* are mixed up more often. However, these two classes can be told apart by the eye region. Astonishingly, the eye area had a good classification rate for identifying shouting. The reason for this is obvious when viewing images with the facial expression *shout* (Fig. 7). Normally, a human closes or pinches his/her eyes while shouting. This is one feature to separate *shout* from *open the mouth* (which is not part of the dataset). All other expressions are classified with nearly equal recognition rates.

So if one of the face regions is occluded or for other reasons is not available for classification, it is possible to recognize facial expressions. But this is not the case in all situations. While using an automatic dialogue system, a user has to speak. Speaking changes the mouth area but does not affect the user state. This makes the mouth

Fig. 6. The three parts of a face used for facial expression classification

Table 2. Confusion matrix (values in %) of facial expression recognition for the face areas. The best result in each line is in **boldface**: the eye area (**top**, overall recognition rate: 63%), the nose area (**middle**, overall recognition rate: 65%), the mouth area (**bottom**, overall recognition rate: 79%)

Eye area				
Reference facial expression	Results			
	smile	*neutral*	*anger*	*shout*
smile	**60**	13	19	8
neutral	14	**52**	29	5
anger	9	26	**59**	6
shout	5	8	4	**83**
Nose area				
Reference facial expression	Results			
	smile	*neutral*	*anger*	*shout*
smile	**77**	6	7	10
neutral	1	**46**	47	6
anger	2	25	**71**	2
shout	22	7	5	**66**
Mouth area				
Reference facial expression	Results			
	smile	*neutral*	*anger*	*shout*
smile	**97**	2	1	0
neutral	2	**55**	43	0
anger	0	33	**67**	0
shout	2	2	0	**96**

area classifier less trustworthy during speaking. A fusion of the classification results of other region classifiers may improve the overall result. Therefore, we examine the fusion of classification results.

3.1 Fusion

Each facial expression may be accentuated in another face region. We developed rules to combine results of the classifier for each face region dependent on user states. For the class *shout* it is clear (96%) that the mouth area is a trustworthy indicator.

Fig. 7. Some samples of eye regions where the mouth concerning facial expression *shout* is clearly visible

The last section of Table 2 also shows that *smile* is classified correctly (with 97%). So the first two rules are:

Rule 1

IF	mouth area classifier classifies shout
THEN	overall result is shout

Rule 1

IF	mouth area classifier classifies smile
THEN	overall result is smile

These two rules are based on the low false alarm rate of the mouth area classifier for the classes *smile* and *shout* (4% for *smile* and 0% for *shout*). The classification rate is 96% and 97% for *shout* and *smile*, respectively. The high recognition rates combined with low false alarm rate makes the mouth area a trustworthy face region for classification of *smile* and *shout*.

During speaking it is not possible to use the mouth area for classification, because this face region has considerable changes due to moving lips. In this case the other regions of the face should be used for facial expression classification. The decision whether the person is speaking or not at the moment can be obtained from a speech recognition system.

The following experiments use only test images with classes not equal to *shout*. The eye area classifier achieves 57%, and the nose area classifier 64%. We introduce the following rules:

Rule 3

IF	both classifiers have same result
THEN	overall result is that result

Rule 4

IF	(one classifier says shout) AND (one classifier says smile)
THEN	overall result is smile

Rule 5

IF	(eyes classifier says neutral) AND (nose classifier says anger)
THEN	overall result is neutral

Rule 6

IF	not classified yet
THEN	overall result is that of nose classifier

The rules were developed using a subset of the total data. This subset was deleted from the test set in order to not influence recognition rates. Using these rules gives an improvement of the recognition rate of 10% (eye area), resp. 3% (nose area).

Fig. 8. Neither the eye nor the nose area classifier was able to classify these images. The displayed facial expressions are (from *left* to *right*): *neutral*, *smile*, *anger*. The first one was classified as *smile* (eye area), *shout* (nose area); The second as *anger* (both), the third as *smile* (eye area) and *neutral* (nose area)

The recognition rate of combining the eye area classifier and the nose area classifier achieves 67%. All classifiers missed the right decision for 23% of the images. It is not possible to solve the classification for these images. Some of these misclassified images are shown in Fig. 8. The confusion matrix for the fusion of results from eye and nose area classifiers is given in Table 3.

4 Conclusion

Facial expression recognition in an automatic dialogue system context has two additional problems compared to facial expression recognition known from the literature (Lien et al., 1998b; Essa and Pentland, 1995; Müller et al., 2001): the expressions are weak because users try to hide them, and users are speaking. The latter, a normal behavior for a dialogue, results in facial changes near the mouth not related to internal user states.

In cases where the mouth area could not be used for classification, a classification and fusion of local area classifiers allows the detection of the current user state. A disadvantage is that the combination results depend on the facial expression. Each face region displays each facial expression with different weights. The combination rules shown above obtain an improvement of 3% (resp., 10%) absolute to 67%.

Table 3. Confusion matrix (values in %) of fusion of results from eye and nose area classifiers; overall recognition rate is 67%

Reference facial expression	Results			
	smile	*neutral*	*anger*	*shout*
smile	**76**	8	11	5
neutral	9	**66**	19	6
anger	3	36	**58**	3

References

J. Adelhardt, C. Frank, E. Nöth, R.P. Shi, V. Zeißler, and H. Niemann. Multimodal Emograni, Data Collection and Presentation, 2006. In this volume.

B. Braathen, M.S. Bartlett, G. Littlewort, and J.R. Movellan. First Steps Towards Automatic Recognition of Spontaneous Facial Action Units. In: *Proc. ACM Conf. on Perceptual User Interfaces (PUI'03)*, pp. 319–242, Vancouver, Canada, 2001.

X.W. Chen and T. Huang. Facial Expression Recognition: A Clustering-Based Approach. *Pattern Recognition Letters*, 24:1295–1302, 2002.

J.F. Cohn, A.J. Zlochower, J.J. Lien, and T. Kanade. Feature-Point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expressions. In: *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pp. 390–395, 1998.

P. Ekman and W.V. Friesen. *The Facial Action Coding System — A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, CA, 1978.

I.A. Essa and A.P. Pentland. Facial Expression Recognition Using a Dynamic Model and Motion Energy. In: *Proc. 5th Int. Conf. on Computer Vision*, pp. 360–367, Cambridge, MA, 1995.

I.A. Essa and A.P. Pentland. Coding, Analysis, Interpretation, and Recognition of Facial Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):757–763, 1997.

C. Frank and E. Nöth. Automatic Pixel Selection for Optimizing Facial Expression Recognition Using Eigenfaces. In: *Pattern Recognition, Proc. 25rd DAGM Symposium*, pp. 378–385, Magdeburg, Germany, 2001.

C. Frank and E. Nöth. Optimizing Eigenfaces by Face Masks for Facial Expression Recognition. In: *Computer Analysis of Images and Patterns – CAIP 2003*, LNCS, pp. 1–13, Berlin Heidelberg New York, 2003. Springer.

T. Hu, L.C.D. Silva, and K. Sengupta. A Hybrid Approach of NN and HMM for Facial Emotion Classification. *Pattern Recognition Letters*, 23:1303–1310, 2002.

S. Kaiser, T. Wehrle, and S. Schmidt. Emotional Episodes, Facial Expressions, and Reported Feelings in Human-Computer Interactions. In: *Proc. 10th Conf. of the Int. Society for Research on Emotion*, pp. 82–86, Würzburg, Germany, 1998. ISRE Publications.

S. Kawato and N. Tetsutani. Real-Time Detection of Between-the-Eyes With a Circle Frequency Filter. In: *Proc. 5th Asian Conference on Computer Vision (ACCV)*, pp. 442–447, Melbourne, Australia, 2002.

M. Kirby and L. Sirovich. Application of the Karhunen–Loève Procedure for the Characterization of Human Faces. *TPAMI*, 12(1):103–108, 1990.

V.P. Kumar and T. Poggio. Learning-Based Approach to Real Time Tracking and Analysis of Faces. In: *Automatic Face and Gesture Recognition 2000*, pp. 96–101, 2000.

H. Li, A. Lundmark, and R. Forchheimer. Video Based Human Emotion Estimation. In: *Int. Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging*, Sept. 1999.

J.J. Lien, T. Kanade, J.F. Cohn, and C.C. Li. A Multi-Method Approach for Discriminating Between Similar Facial Expressions, Including Expression Intensity Estimation. In: *Proc. Computer Vision and Pattern Recognition (CVPR'98)*, pp. 853–859, 1998a.

J.J. Lien, T. Kanade, J.F. Cohn, and C.C. Li. Automated Facial Expression Recognition Based on FACS Action Units. In: *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pp. 390–395, 1998b.

M.J. Lyons, L. Budynek, and S. Akamatsu. Automatic Classification of Single Facial Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(12):1357–1362, 1999.

A.M. Martinez and R. Benavente. The AR FAce DAtabase. Technical Report 24, Computer Vision Center, Purdue University, West Lafayette, IN, 1998.

B. Moghaddam and A.P. Pentland. Face Recognition Using View–Based and Modular Eigenspaces. In: *Automatic Systems for the Identification and Inspection of Humans, SPIE'94*, vol. 2257, 1994.

M. Müller, F. Wallhoff, F. Hülsken, and G. Rigoll. Facial Expression Recognition Using Pseudo 3-D Hidden Markov Models. In: *Pattern Recognition, Proc. 23rd DAGM Symposium*, pp. 291–297, Munich, Germany, 2001.

N. Oliver, A.P. Pentland, and F. Berard. LAFTER: Lips and Face Real Time Tracker. In: *Proc. Computer Vision and Pattern Recognition (CVPR'97)*, pp. 123–129, Puerto Rico, 1997.

E.E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: Training and Application. Technical Report A. I. Memo No. 1602, Massachusetts Institute of Technology, Cambridge, MA, 1996.

T. Otsuka and J. Ohya. Recognizing Multiple Persons' Facial Expressions Using HMM Based on Automatic Extraction of Significant Frames From Image Sequences. In: *Int. Conf. on Image Processing (ICIP 1997)*, pp. 546–549, Oct 1997.

T. Otsuka and J. Ohya. Extracting Facial Motion Parameters by Tracking Feature Points. In: *Proc. 1st Int. Conf. AMCP'98*, LNCS 1554, pp. 433–444, Berlin Heidelberg New York, 1999. Springer.

M. Pantic. *Facial Expression Analysis by Computational Intelligence Techniques*. PhD thesis, Faculteit der Informatietechnology en Systemen, TU Delft, The Netherlands, 2001.

F. Schiel and U. Türk. Wizard-of-Oz Recordings, 2006. In this volume.

K. Schwerdt, D. Hall, and J.L. Crowley. Visual Recognition of Emotional States. In: *Proc. Int. Conf. on Multimodal Interfaces (ICMI'00)*, pp. 41–48, Beijing, China, 2000.

E.P. Simoncelli. *Distributed Analysis and Representation of Visual Motion*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, 1993.

C.E. Thomaz, D.F. Gillies, and R.Q. Feitosa. Using Mixture Covariance Matrices to Improve Face and Facial Expression Recognitions. In: *Proc. 3rd Int. Conf. of Audio- and Video-Based Biometric Person Authentication AVBPA R01*, LNCS 2091, pp. 71–77, Berlin Heidelberg New York, June 2001. Springer.

Y. Tian, T. Kanade, and J.F. Cohn. Recognizing Action Units for Facial Expression Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(2):97–115, 2001.

Y.L. Tian, T. Kanade, and J.F. Cohn. Recognizing Lower Face Action Units for Facial Expression Analysis. In: *Proc. Int. Conf. on Automatic Face and Gesture Recognition*, pp. 484–490, 2000.

M. Turk and A.P. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991a.

M. Turk and A.P. Pentland. Face Recognition Using Eigenfaces. In: *Proc. Computer Vision and Pattern Recognition (CVPR'98)*, pp. 586–591, 1991b.

L. Wiskott. Phantom Faces for Face Analysis. *Proc. 7th Intern. Conf. on Computer Analysis of Images and Patterns, CAIP'97, Kiel*, 1296:480–487, 1997.

W.S. Yambor, B.A. Draper, and J.R. Beveridge. Analyzing PCA-Based Face Recognition Algorithms: Eigenvector Selection and Distance Measures. In: *2nd Workshop on Empirical Evaluation Methods in Computer Vision*, Dublin, Irland, 2000.

Z. Zhang, M. Lyon, M. Schuster, and S. Akamatsu. Comparison Between Geometry-Based and Gabor-Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron. In: *Proc. 3rd Int. Conf. on Automatic Face and Gesture Recognition*, pp. 454–459, Nara, Japan, 1998.

Multiple Biometrics

Stephan Grashey and Matthias Schuster

Siemens AG, CT IC 5, München, Germany
{Stephan.Grashey,Matthias.Schuster}@siemens.com

Summary. Authentication is undoubtedly an important task for all systems providing a component responsible for the interaction between humans and computers. Traditional authentication techniques like PINs, passwords or ID cards show significant drawbacks: they might be forgotten, misplaced or lost, or even stolen, copied or forged. Biometrics use physical or behavioral characteristics to verify the identity of a person and thus overcome these problems. In this respect, biometric technology may be easier and more comfortable to use.

First, a short overview of biometric technology in general is given. Then, the main part of this chapter explains the biometric technologies integrated in the SMARTKOM system. In particular, a new approach to combine several biometrics in a multimodal device is presented. The performance of this proposed combination method is shown to be superior to that of the single biometric subsystems.

1 Biometrics

In the mobile and networked society, the use of many devices and services requires a preceding authentication. Today, authentication is mostly performed by checking possession (e.g., physical access using a key or ID card) or knowledge (e.g., access to a computer system using a password). For high-security transactions like banking, a combination of both (e.g., access to an ATM by credit card and PIN) is applied. The problem with possessions is that they can be lost, stolen, forgotten or misplaced. Knowledge bears the problem that it is difficult to remember the increasing number of passwords and PINs required. Many people write their PINs down or choose simple passwords that can easily be guessed by an adversary. As a result, the system cannot distinguish between an authorized person and an impostor who has acquired the necessary knowledge or possession.

Biometrics, on the other hand, is a method to provide genuine person identification. Biometric technology measures physical or behavioral characteristics to determine the identity of a person. The technology acts as a front end to a system that requires authentication before it can be accessed or used. While a few years ago biometrics were mainly used in science fiction, the technology is now mature and ready for application.

The assignment of an identity can be an identification or the verification of a claimed identity. In case of identification only the biometric feature is presented to the system and the system assigns the identity by comparing the feature to all references in a database (one-to-many search). For verification, the user first claims an identity and the system then checks the biometric feature against the reference assigned to this identity (one-to-one comparison).

Obviously, identification is very comfortable, since no user interaction besides the biometric feature acquisition is requested, but identification requires a precise biometrics and a full search through a central reference database. In contrast, verification systems are generally faster and more accurate than identification systems.

According to the respective biometric characteristics, biometric systems can broadly be classified into two categories: physiological and behavioral. Physiological biometrics verify fingerprints or hand contours, i.e., characteristics that practically do not change over a person's lifetime. Behavioral biometric systems analyze the way a person is acting. Examples for monitored behaviors are signature movements or voice-prints. It should be noted that behavioral biometrics also have a physiological component, e.g., the articulatory system in the case of speech.

In general, biometrics is a pattern recognition task. A biometric system consists of the components signal acquisition, feature extraction and classification (pattern or model match), see Fig. 1. For new users reference templates or models have to be generated first (so-called enrollment). The classification is subsequently performed by comparing the actual recorded sample (image or signal) with the stored reference template and calculating the matching difference. Identification or verification is achieved by setting a threshold for the matching difference to discriminate between accepted and rejected trials.

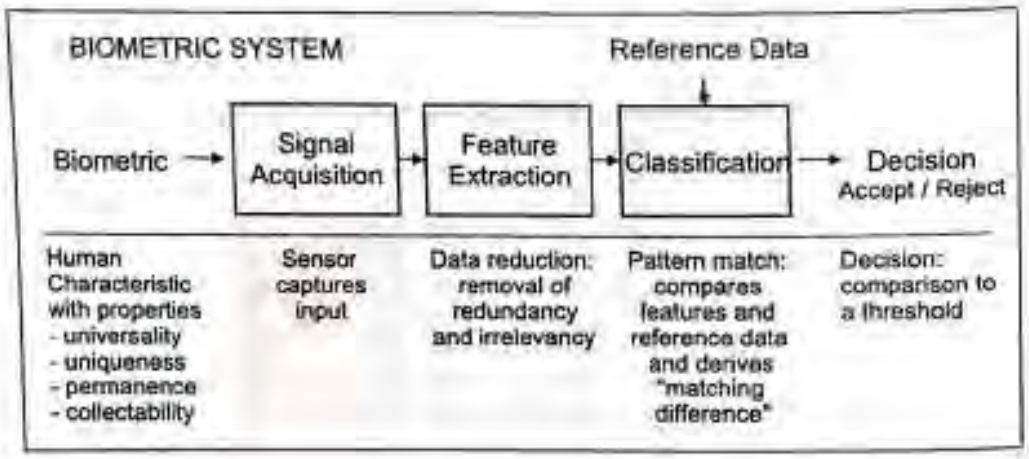

Fig. 1. Components of a biometric system. Individual features are derived from the sensor signal and compared to a reference data set. A biometric system performs pattern recognition

One possibility to measure the performance of biometric systems is the determination of the two error rates: the false acceptance rate (FAR) and the false rejection rate (FRR). The FAR is the probability of letting an unauthorized user pass for someone else. This error rate must be low enough to provide sufficient security for a given application. The FRR is the probability that the biometric does not recognize an authorized user and therefore denies him or her access. A low FRR is so important because this type of error can occur every time an authorized person uses the device. The FRR therefore is critical for user acceptance.

The working point of a biometric device, and thereby the effective value of the FAR and FRR, can be adjusted by increasing or decreasing the decision threshold of the device. How much each error rate is affected by altering the threshold is a characteristic of each manufacturer's biometric system. A system may exhibit an extremely low FAR, but the corresponding FRR may be totally unacceptable. The balance of the two error rates for a given application is critical for the success of a biometric installation.

The point at which the FAR equals the FRR is called the equal error rate (EER). The ERR is often used as a good indication of the overall performance, though the effective working point might be somewhere else.

1.1 Biometrics in SmartKom

1.1.1 Background

One intention of SMARTKOM is to create intelligent user interfaces that accept and support the natural communication style of humans and thus contribute to a user-friendly and user-centered design of information technology. In this context it is also necessary, of course, to provide not only safe and reliable but also natural access to knowledge, information and data. This implies a natural way of authentication: As we go about our daily lives, we are used to identifying other people by the way they look, speak or walk — we are performing a biometric authentication. Thus, in SMARTKOM biometrics is also the method of choice when talking about a convenient and natural way of authentication.

Which biometrics is most suitable depends, on the one hand, on the type of application and, on the other hand, on the personal attributes and preferences of a user:

- Applications make high demands on security, that are probably not satisfiable by every type of biometrics.
- In other cases, the application scenario advises a particular type of biometrics, e.g., speaker recognition in telephony applications.
- In addition, user acceptance is not the same for all kinds of biometrics but, for example, depends on the data acquisition device, hygienic aspects, intrusiveness and so on.
- Further on, a particular biometrics may be unsuitable for some users at some times, e.g., because of hoarseness in the case of a speaker recognition, unstable signatures in the case of a signature verification or an injury in the case of a hand contour recognition.

Therefore, because of the need for both an increased reliability and security and an enhanced user comfort and user acceptance, a multiple biometric approach was chosen. This approach consists of a hand contour recognition, a signature verification and a speaker verification. Such multiple biometrics are also easily integrated into a multimodal system like SMARTKOM.

Verification (one-to-one match) was chosen as the appropriate method of authentication, mainly because of performance reasons and privacy issues, especially in SMARTKOM-Public. Below, the particular biometric technologies either enhanced or developed in SMARTKOM are introduced.

1.1.2 Hand Contour Recognition

In SMARTKOM a gesture recognition is integrated, where the hand is used for pointing — so why not use it to recognize the user?

Biometric systems based on hand contour authentication are not new and have been available since the early 1970s. One of the most well known hand contour application is the INSPASS system in the Unites States, allowing frequent travelers to bypass waiting lines of immigration ("Your hand is your passport"). Hand contour verification was also used, for example, to secure the 1996 Olympic games in Atlanta.

Commercial hand recognition systems are small boxes that look like a trap. Up to five pegs ensure an accurate placement of a user's hand (Jain et al., 1999). This design limits usage by certain populations, e.g., people with very small or very big hands. Contrary to that, in SMARTKOM a hand contour recognition system was developed that does without such contraptions and allows free positioning of the user's hand within the work space. Here, also a life check can easily be performed to protect against simple fakes by asking the user to close his fingers and spread them again.

The silhouette of the hand as obtained from the SiViT video camera and processed from the device "realDocument" is used to extract invariant and characteristic features like palm size, finger lengths, finger thickness and curvatures. Although these metrics do not vary significantly across the population, they are distinctive enough for verification or usage in multiple biometric systems like SMARTKOM. Some of the advantages of hand contour verification are (Nanavati et al., 2002):

- It is nonintrusive.
- It is able to operate in challenging environments.
- It has a high user acceptance.
- It is the biometrics of choice in combination with gesture-based devices.

1.1.3 Signature Recognition

The signature is a widely spread method to authorize documents. Signing a document is a socially and legally accepted binding agreement.

Usually signatures are verified by comparison of images, i.e., by the form of the signature. This is called static verification. In dynamic signature verification the way

of signing is also taken into account. Typical features are velocity, acceleration and pressure (Gupta and McCabe, 1997). Since the signature is a quite variable characteristic, five to six signatures are required during enrollment.

The main advantages of dynamic signature recognition are (Nanavati et al., 2002):

- It leverages existing processes.
- It is perceived as non intrusive.
- It has a high user acceptance.
- Users can change their signatures if needed.
- It has a "built-in" liveness test.

Dynamic signature verification is ideally suited for online applications. The signature may be performed on the paper document or pasted into an electronic document. Hardware expenses are comparably low: only an off-the-shelf PC tablet is required. In SMARTKOM, the flat surface of the tablet also serves as the projection and pointing area of the SiViT system.

The main field of application is the authentication of electronic documents or everywhere else where today simple signing is used. Financial transactions (from insurance to credit cards) and any form of contract may be safeguarded by this biometric technology.

1.1.4 Speaker Recognition

The voice is a good characteristic of every human. In communication we are used to identifying our partners from their voices, for example, on the telephone. In speaker verification the biometric system does the same.

Human speech depends on the anatomy of the vocal tract (mouth and nose cavities) of the person and on the linguistic behavior (pronunciation, speaking rhythm) the speaker has developed. Voice verification or speaker recognition systems model the acoustic features of speech that have been found to differ among individuals and that reflect both physical and learned behavioral patterns (Campbell, 1997).

In principle, speaker verification can operate in text-dependent as well as text-independent mode.

- Most speaker recognition systems use text-dependent input, which involves selection and enrollment of one or more voice pass-phrases. In this case the system knows what will be spoken and can perform a pattern match, but does not necessarily have explicit (phonetic) knowledge about the pass-phrases:
  - Language-independent input is possible if the user is free to choose the pass-phrase. In this case the system has no explicit (phonetic) knowledge about the input.
  - Text-prompted input is a version of text-dependent input which is used when there is concern about an impostor playing back a previously recorded voice pass-phrase. Here the system prescribes the pass-phrases to be spoken and can rely on explicit phonetic knowledge about them. Such a system is typically language dependent.

- Text-independent mode would allow free speech for verification where the variability of words and the natural personal variability have to be compensated. In this mode, the verification is performed independently of what is spoken. Explicit knowledge could only be gained after successful speech recognition and possibly language identification. This mode is still a research topic.

Generally, text-dependent recognition systems allow a more precise modeling and therefore yield higher security. Hence, also in SMARTKOM a text-dependent recognition was applied. The advantages of speaker recognition are:

- It is ideally suited in the context of communication applications and has the ability to leverage existing telephony infrastructure.
- It is perceived as nonintrusive.
- It allows for synergy effects in combination with other processes such as speech recognition.

To account for the natural variations in everyone's voice a minimum of three enrollments is required. Furthermore, to achieve a high accuracy of speaker recognition, also some quality measures were developed:

- An outlier detection ensures that a particular user does not change his pass-phrase between the separate enrollments.
- The length of the chosen pass-phrase is checked — utterances that are too short are rejected.
- Noise and channel compensation are integrated to minimize the effects of background noise or different channels.

2 Multiple Biometrics

2.1 Background

In the development of human computer interaction, the aspect of multimodality gains increasing importance. In the future small personalized mobile devices and public information and transaction terminals,such as those being developed in the SMARTKOM project, will allow the user to interact with different modalities such as voice, handwriting, gestures or mimic. This has also an effect on the usage of biometrics. By different input sensors like microphone, camera or touch screen, several biometric features such as voice, signature, hand contour or facial form can be recorded, interpreted and combined. This is called multiple biometrics and offers authentication with a higher level of security or comfort (Hong and Jain, 1999). Within SMARTKOM a new multiple and multimodal approach was developed to combine the three individual biometric technologies described above: speaker recognition, signature verification and hand contour recognition (Fig. 2).

Biometric systems are always two-class decision problems where the features of the authorized users, the genuines, must be separated from those of the impostors. The underlying similarity measurements are either costs or scores. Costs are the

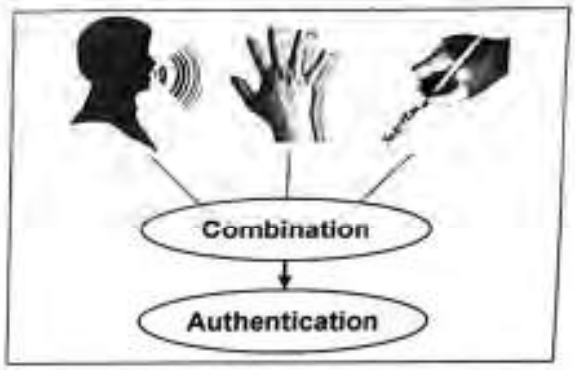

Fig. 2. Multimodal biometrics in SMARTKOM

calculated distances between reference and test patterns, whereas scores indicate the probability that the patterns stem from the same user. Multimodal biometrics are also binary classification problems, where the dimensions of the feature space depend on the number of single biometrics considered. The demanding task in multiple biometrics is to combine or fuse the expert outputs optimally in order to reach a higher performance than any single biometrics. An ideal multiple biometric system should meet the following requirements:

- higher degree of security than the best integrated individual biometric technology
- low and possibly selectable FAR
- low and possibly selectable FRR
- flexibility with respect to modification
- fast decision time

In addition, multiple biometric systems may also exhibit the following characteristics:

- Increase in robustness and reliability because of redundancy.
- Spoofing attacks become more difficult.
- Increase in user-acceptance (e.g., if a user can chose his or her preferred biometric).
- Enlargement of the population (e.g., if the universality of a particular biometric is insufficient).

In general, we distinguished between two main strategies to combine different biometrics: feature fusion and decision fusion (Ross and Jain, 2003).

Feature fusion (Fig. 3) or early fusion combines all features extracted from different modalities. As the new feature space has a higher dimensionality and all features can be weighted optimally, the strategy is generally believed to be more discriminative. However, this fusion method assumes feature access to all used biometrics and expert knowledge about the single biometrics and their features. As the type and number of features differs in different combinations and therefore the feature

combination and weighting has to be readjusted, the feature fusion method is not well suited for flexible biometric combinations with varying types and numbers of biometrics.

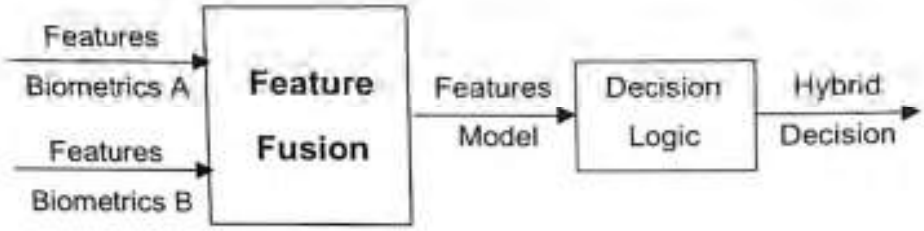

Fig. 3. Feature fusion

Most multimodal approaches are based on decision fusion or late fusion, which combines the output of all single biometrics (Fig. 4). This output of the so-called expert modules are either similarity measurements (costs, scores) or binary decisions. Usually the supervisor or decision module does not need any expert knowledge about the single biometrics and their features. Therefore this method is more flexible regarding the modification of the combination (changing number or type of biometrics). In some approaches the decision fusion unit can be trained and therefore adjusted automatically by training output data of the single biometrics.

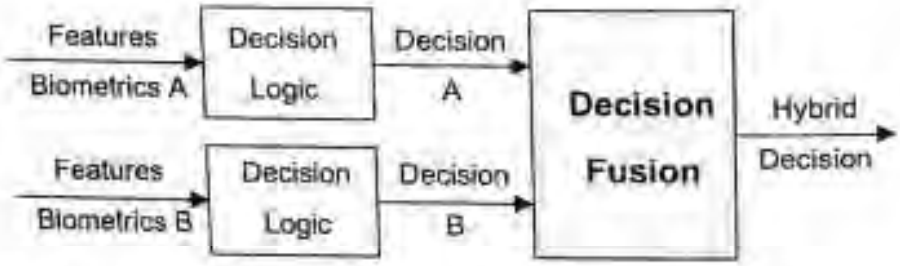

Fig. 4. Decision fusion

Typical decision fusion strategies are logic combination (Dieckmann et al., 1997), where the decisions of the single biometrics are combined by logic operators, or the calculation of a global weighted score from the costs or scores of the single biometrics (Brunelli and Falavigna, 1995). Further approaches are neural networks (Dieckmann et al., 1997), support vector machines (Ben-Yacoub, 1999), and Bayesian statistics (Bigün et al., 1997).

The existing methods have some disadvantages, however (Daugman, 1999). These may be inherent assumptions about the distribution of the costs or scores of the single biometrics, large training effort, insufficient error rates, or difficult modification for the usage of different types or numbers of biometrics as the combinations are optimized for a special configuration.

2.2 SmartKom Approach

The multimodal biometric approach developed in the SMARTKOM project is also a decision fusion method, but it does not use any of the above-described combination methods and does not make any assumption about the distribution of scores or costs like in Bayesian statistic approaches (Bigün et al., 1997). It is based on the multidimensional probability density of the similarity measurements for the genuines and imposters, which can be derived from the real measured cost or score distributions of the individual biometrics.

In the following the approach is described by an example, where biometrics 1, B_1, which is based on costs, and biometrics 2, B_2, which provides scores, are combined to an overall multiple biometric system. The cost or score probability densities of a single biometric are calculated by normalizing the integrals of the single distributions to unity. Figures 5 and 6 show the probability densities $p_g(x)$, $p_i(x)$ of B_1 and $p_g(y)$, $p_i(y)$ of B_2 for the genuines (g) and impostors (i).

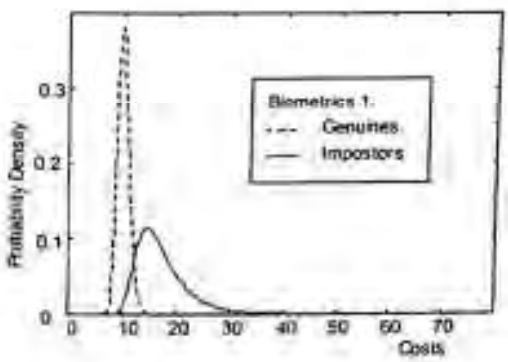

Fig. 5. Cost probability density B_1

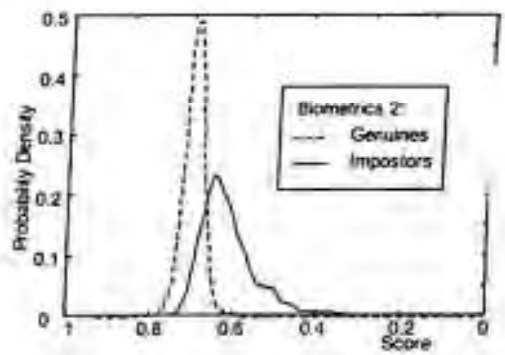

Fig. 6. Score probability density B_2

If the different biometrics can be considered to be statistically independent, the multidimensional probability densities of the costs or scores are given by the product of the probability densities of the single biometrics for both genuines and impostors. The assumption of statistical independence is fulfilled in most cases, especially with biometrics that use different modalities.

The multidimensional cost or score probability density of two statistically independent biometrics is calculated as follows:

$$p_g(x,y) = p_g(x) \cdot p_g(y), \qquad \text{Genuines} \tag{1}$$

$$p_i(x,y) = p_i(x) \cdot p_i(y), \qquad \text{Impostors} \tag{2}$$

Figure 7 shows the probability densities of the similarity measurements for genuines and impostors in the combination of B_1 and B_2.

The concept of our decision strategy is to find suitable borderlines in the multidimensional cost or score space that separate the class of genuines from the impostor

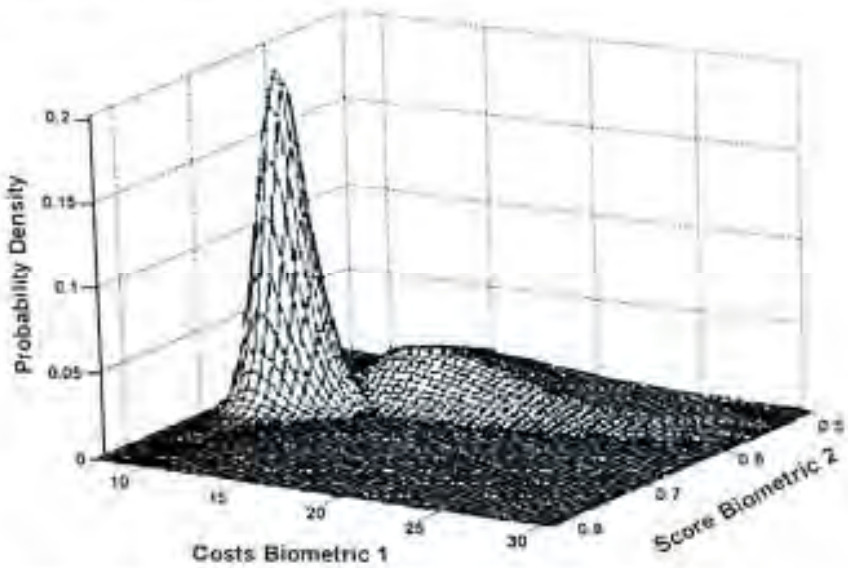

Fig. 7. Two-dimensional cost–score probability densities of B_1 and B_2

class in an optimal way and under special constraints, e.g., a given FAR or FRR. The multidimensional probability densities of the genuines and impostors are used to calculate these decision curves. There are several methods to find such borderlines based on the probability densities. The SMARTKOM approach is based on a strategy that determines the decision curves under the constraint of a given FAR. Within this method the decision borderlines are the curves that are described by the points where the ratio of the two probability densities has a constant value V. This value has to be adjusted properly by test data to yield the intended FAR value, FAR_{Target}. A score or cost point (x,y) is assigned to the class of genuines when:

$$\frac{p_g(x,y)}{p_i(x,y)} > V(FAR), \qquad FAR = FAR_{Target}. \tag{3}$$

In Fig. 8 the decision border and the area of acceptance is illustrated for an exemplary combination of B_1 and B_2 with this fusion method. In addition, lines of constant probability density of the similarity measurements for genuines and impostors are shown.

2.3 Experiments and Results

To examine the performance of the SMARTKOM multimodal biometric method, qualitative measurements were made for the combination example of B_1 and B_2. The corresponding error rates were calculated and compared with those of the single biometrics and logic combination methods, which are almost solely used in present multimodal biometric systems.

For that, the data of costs and scores from two of the three biometrics integrated in SMARTKOM, speaker verification and signature verification, were used for the

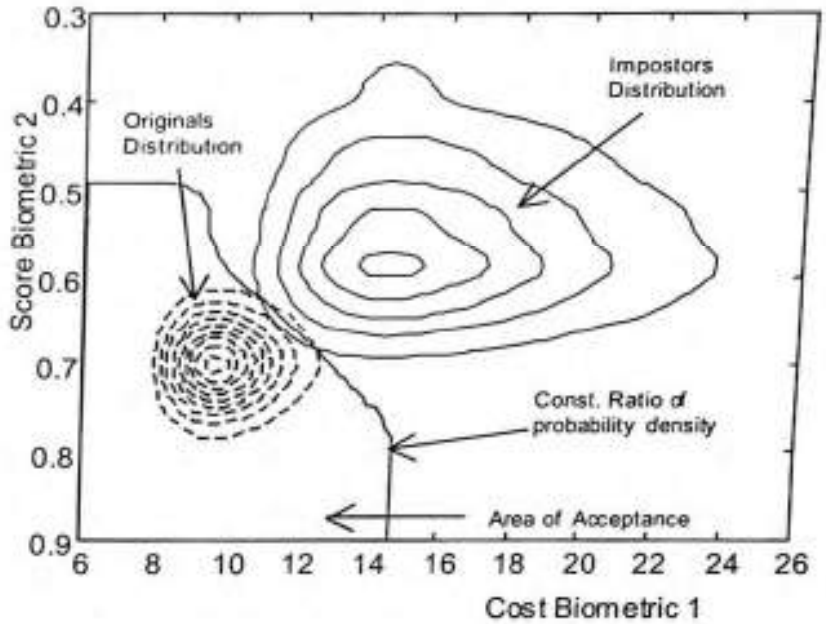

Fig. 8. Area of acceptance

determination of the single cost or score distributions, the borderlines in the decision space and for evaluating the combination method.

For speaker verification 1500 genuine and 70,000 imposter costs were available. The values result from a database of 47 persons and the SMARTKOM speaker verification system, which utilizes two-digit random numbers between 21 and 99. The SMARTKOM dynamic signature verification system and a signature database of 20 persons offered 2150 genuine and 2200 imposter scores of the second biometric.

All similarity measurements were divided into a training set for the calculation of the decision criterion and into a distinct evaluation set to evaluate the described combination method. For training and evaluation the cost and score values were combined randomly to more-dimensional score–cost vectors for genuines and impostors, because the signature and speaker data do not stem from the same persons. Thus, statistical independence of the data used is guaranteed.

Within the evaluation, the FRR values were calculated for several given FAR values. This was performed not only for the developed combination method, but also for the individual biometrics and logical AND and OR combination. The AND combination accepts the user solely when every single biometrics returns a positive result, whereas an OR combination requires only the acceptance from at least one biometric. Within the evaluation the best operating points and therefore the best FRR values for given FAR values were also calculated for these logic combinations.

In Fig. 9 the FRR values of the described combination method, of the two logic combinations, and of biometrics 1 and biometric 2 are illustrated as a function of given FAR values. This type of diagram is called receiver operating characteristics

(ROC), and is a standard method to compare different biometric systems (Biometric Working Group).

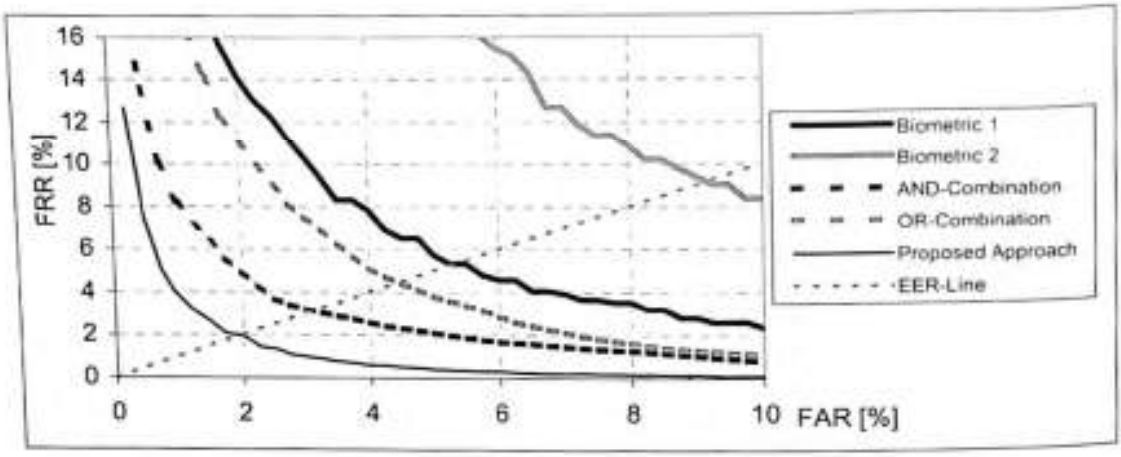

Fig. 9. ROC curves

The ROC diagram shows that biometric 1 has better performance than biometric 2. The logic combinations achieve better results than the single biometrics. As the best operating points of the logic combinations were calculated for given FAR values, the resulting threshold values differ from the thresholds of the single biometrics for the same FAR. Therefore a performance improvement compared to the single biometrics is also plausible for the OR combination. But the SMARTKOM combination yields the best results: The method based on the multidimensional probability densities outperforms the single biometrics and the logic combinations in all operating points.

3 Conclusion

Within SMARTKOM, biometrics ensure natural, comfortable and secure user authentication. Three biometric technologies are integrated: hand contour verification, signature verification and text-dependent speaker verification.

An approach for the combination of these individual biometrics to a multiple and multimodal biometric system was developed. It is based on the multidimensional probability densities of the costs or scores for genuines and impostors. There are several strategies to find decision borders in the space of the similarity measurements spanned by these probability densities. Here an example was described were two biometrics were combined with the constraint of a given FAR value. Possible decision borders can be curves where the ratio between the genuine and impostor probability density has a constant value. The resulting FRR values are compared to the corresponding error rates for each single biometric and also to the results from

logical AND and OR combinations. In every operating point the SMARTKOM approach exhibits superior performance to the stand-alone biometrics and the logic combinations. Furthermore, the method flexibly allows modification and integration of different types and numbers of biometrics.

References

S. Ben-Yacoub. Multi-Modal Data Fusion For Person Authentication Using SVM. In: *Proc. 2nd Int. Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, pp. 25–30, 1999.

E.S. Bigün, J. Bigün, B. Duc, and S. Fischer. Expert Conciliation for Multi-Modal Person Authentication Systems by Bayesian Statistics. In: J. Bigün, G. Chollet, and G. Borgefors (eds.), *Audio and Video based Person Authentication – AVBPA97*, LNCS 1206, pp. 311–318, Berlin Heidelberg Germany, 1997. Springer.

Biometric Working Group. Best Practices in Testing and Reporting Performance of Biometric Devices. http://www.cesg.gov.uk/site/iacs/itsec/media/protection-profiles/BBP.pdf. Cited 5 April 2006.

R. Brunelli and D. Falavigna. Person Identification Using Multiple Cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(10):955–966, 1995.

J.P. Campbell. Speaker Recognition: A Tutorial. *Proc. IEEE-1997*, 85(9):1437–1462, 1997.

J. Daugman. Biometric Decision Landscapes. Technical Report TR482, University of Cambridge Computer Laboratory, Cambridge, UK, 1999.

U. Dieckmann, P. Plankensteiner, and T. Wagner. SESAM: A Biometric Person Identification System Using Sensor Fusion. *Pattern Recognition Letters*, 18:827–833, 1997.

J. Gupta and A. McCabe. A Review of Dynamic Handwritten Signature Verification. Technical report, James Cook University, Townsville, Australia, 1997.

L. Hong and A. Jain. Multimodal Biometrics. *Biometrics – Personal Identification in Networked Society*, pp. 327–344, 1999.

A.K. Jain, A. Ross, and S. Pankanti. A Prototype Hand Geometry-Based Verification System. In: *2nd Int. Conf. on Audio and Video-Based Biometric Person Authentication (AVBPA)*, pp. 166–171, Washington, DC, 1999.

S. Nanavati, M. Thieme, and R. Nanavati. *Biometrics — Identity Verification in a Networked World*. John Wiley, New York, 2002.

A. Ross and A.K. Jain. Information Fusion in Biometrics. *Pattern Recognition Letters*, 24(13):2115–2125, 2003.