Language Technology II: Natural Language Dialogue

Dialogue System Design and Evaluation

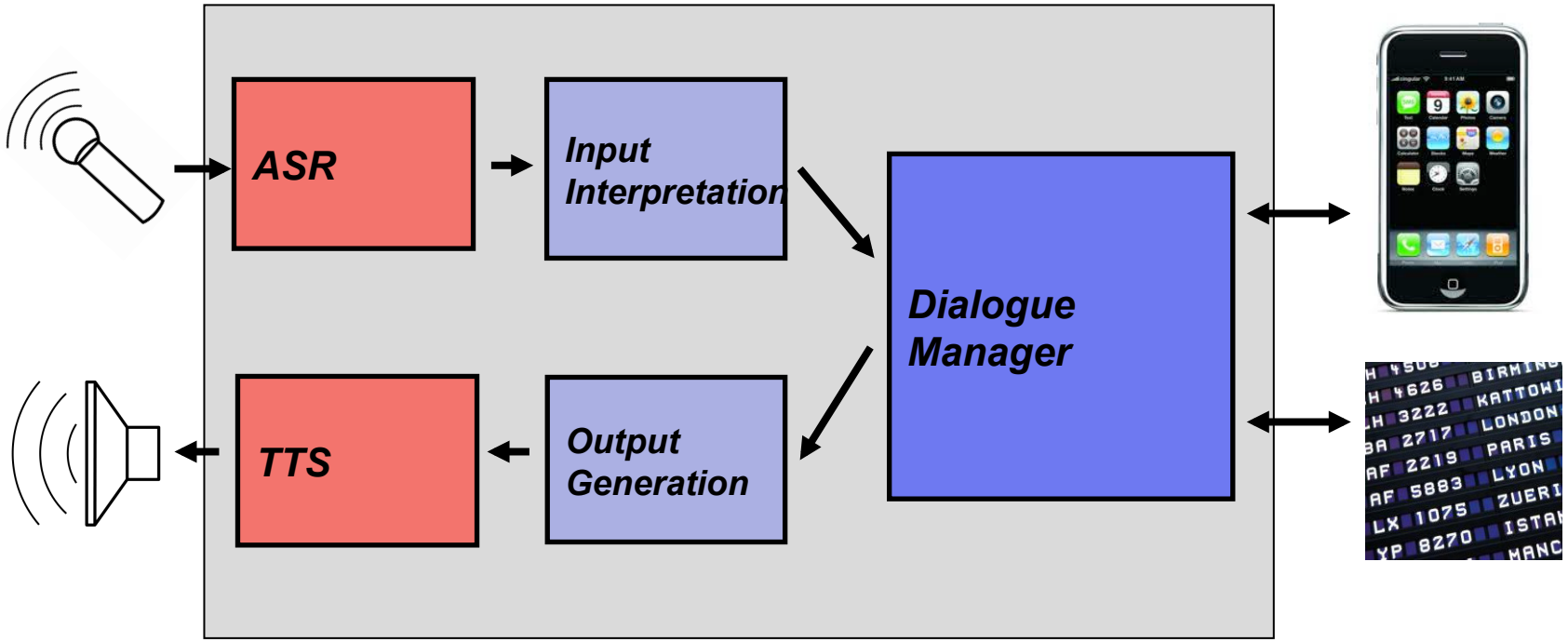
Ivana Kruijff-Korbayová
ivana.kruijff@dfki.de

(slides based on Manfred Pinkal 2012)

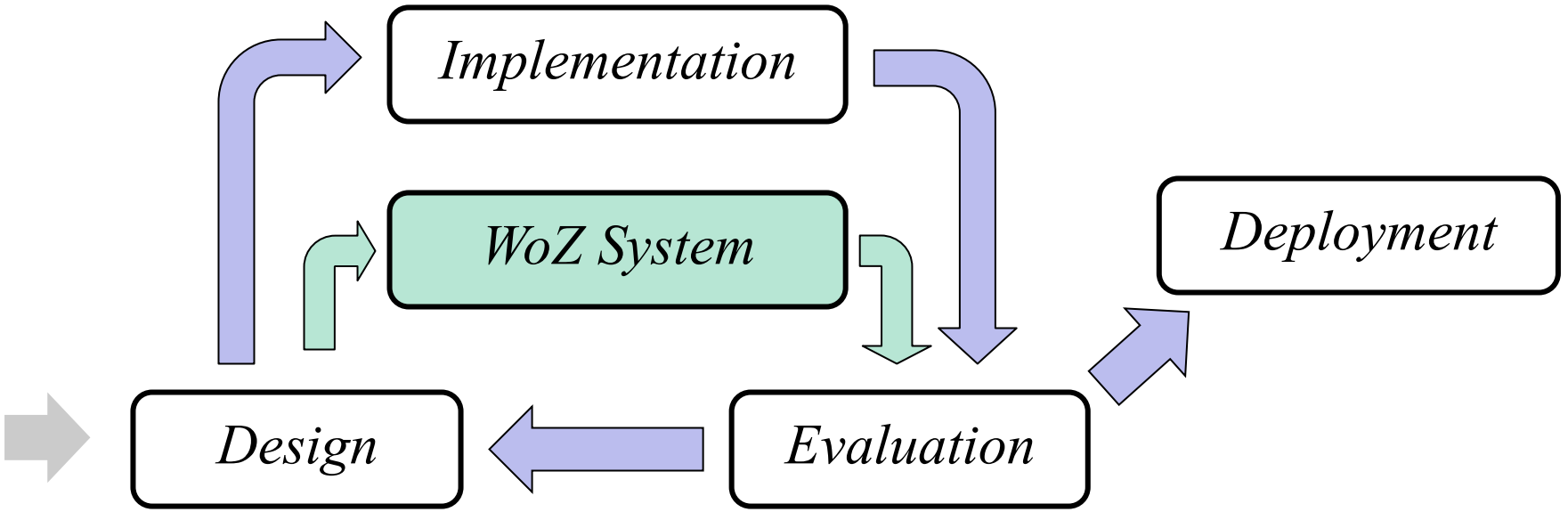
Outline

- Dialogue system architecture
- Wizard of Oz simulation methodology
- Input interpretation
- Output generation
- Design principles
- Evaluation

Dialog System: Basic Architecture



Wizard-of-Oz Simulation



Wizard-of-Oz Studies

- Experimental setup, where a hidden human operator (the “wizard”) simulates (parts of) a dialogue system.

- Subjects are told that they interact with a real system.

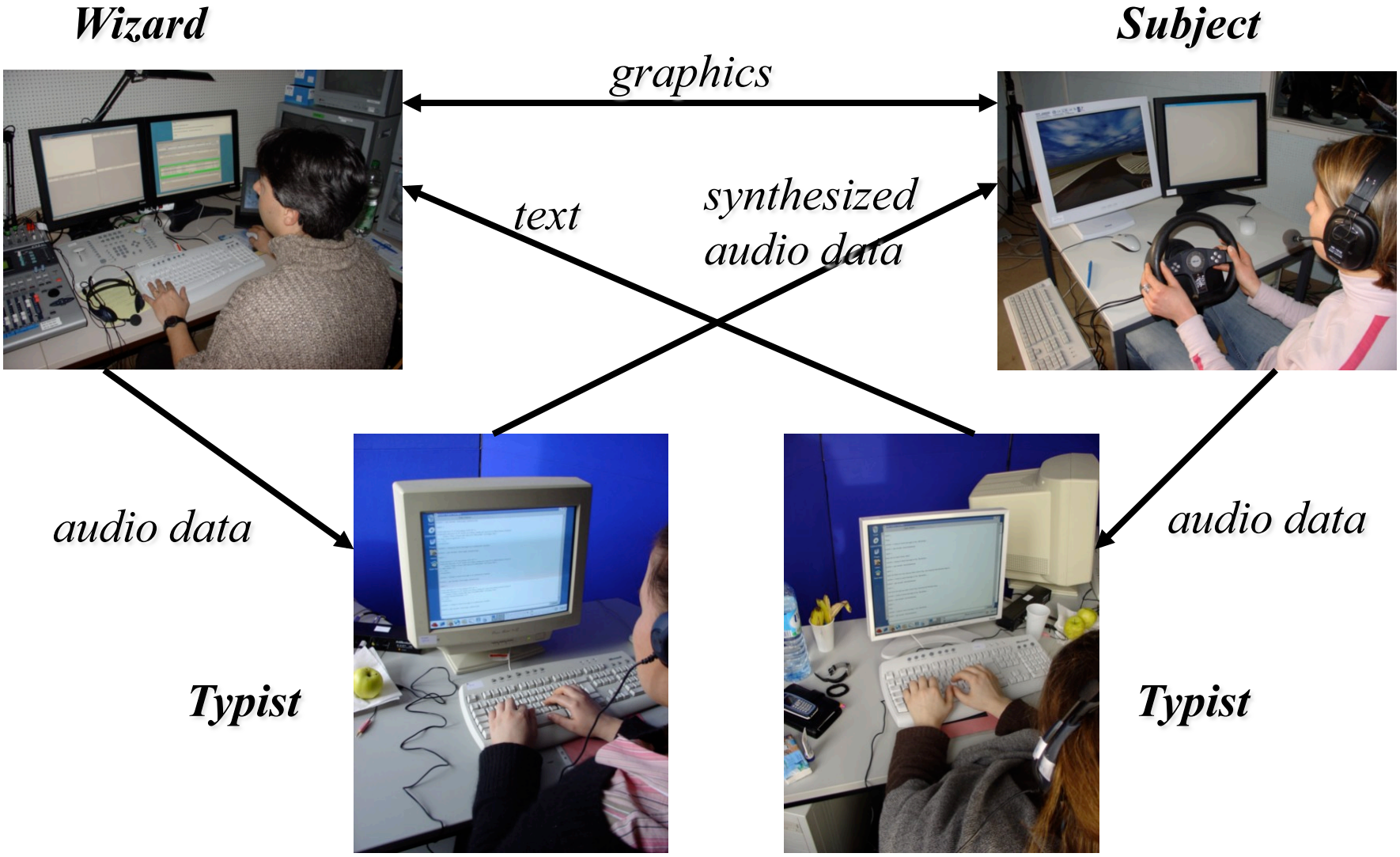
Wizard-of-Oz Studies

- The challenge of providing a convincing WoZ environment:
  - Produce artificial speech output by typing + TTS (speed!)
  - Induce recognition errors by introducing artificial noise, or presenting input to wizard in a typed version, randomly overwriting single words
  - Constrain natural, conversationally smart wizard reactions by predefining possible system actions and output templates, which the wizard must use.
  - Computer systems are much more efficient in database access, mathematical calculation etc.: Provide the wizard with appropriate interfaces for quick mathematical calculation and database lookup. (depends on task)

An example: WoZ Study in TALK

- Domain: MP3 Player
- Scenario: In-car and In-home
- Multimodal dialogue:
  - Input by speech and ergo-commander/ Keyboard
  - Output by speech and graphics (display)
- Example tasks for subjects:
  - Play a song from the album "New Adventures in Hi-Fi" by REM.
  - Find a song with “believe” in the title and play it.

Information Flow



WoZ Studies: Benefits

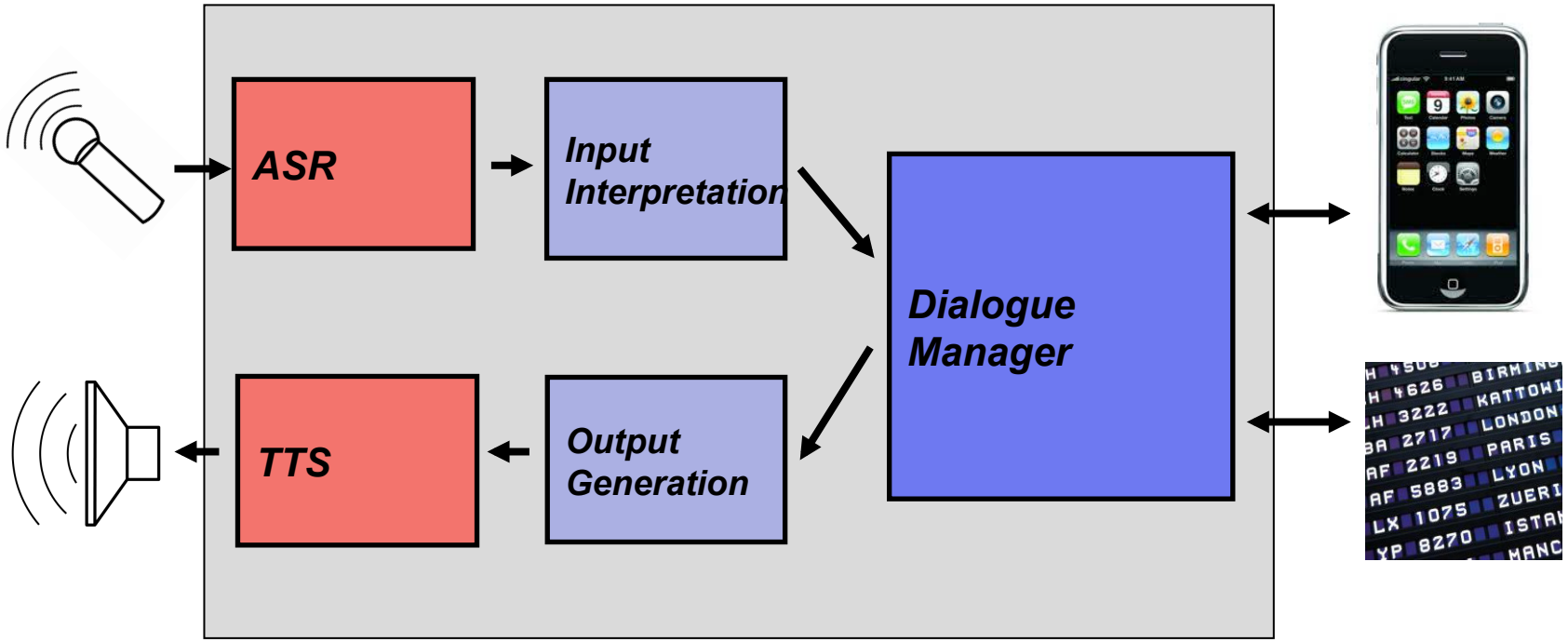
- Evaluation of system design at an early stage, avoiding expensive implementation.
  (However: don't underestimate complexity of WoZ set up)

- Full control over and systematic variation of speech recognition performance.
  (However: realistic ASR errors are hard to simulate)

- Collection of domain- and scenario-specific language data at an early stage:
  - for a qualitative analysis of the dialogue behavior of subjects
  - to train or adapt statistical language models

- Systematic exploration of dialogue strategies by varying instructions to the wizard.

Dialog System: Basic Architecture



Input Interpretation

- Typically, NL (speech) input is mapped to shallow semantic representations:
  - „Take me to the third floor, please“; „Third floor“; „Floor number three“; „Three“ all express the same information in the context of the question „Which floor do you want to go?“
  - „5:15 p.m.“, „17:15“ „a quarter past five“ express the same time information

Input Interpretation and Language Models

- How do we get from user input to representations of the relevant information that drives the dialogue manager?
- We use interpretation grammars.
- The status of interpretation grammars is different dependent on the different kinds of language models used in the ASR component of the dialogue system.
- Two basic methods:
  - Hand-coded Recognition Grammars
  - Statistical Language Models (SLMs)

Recognition Grammars

- Hand-coded Recognition Grammars
  - Typically written in BNF notation (Context-free grammars)
  - Typically shallow “semantic grammars” with no recursion
  - Are compiled to regular grammars/finite automata (by ASR system) without loss of information

- An example:

  $turn = [please] turn l turn $direction ;

  $direction= (backlbackward)l $side;

  $side = [to the](left l right)

Properties of recognition grammars

- Allow quick and easy specification of application-specific and dialogue-state specific language models
- Thereby drastically reduce search space for recognizer
  - Example: $yn_answer = yes I no
- But: Strictly constrain recognition results to the language specified in the grammar.
- Keyword Spotting
  - Working with wildcards

  Example:

  $turn = GARBAGE* turn I turn $direction GARBAGE* ;

  $direction= (backIbackward)I $side;

  $side = GARBAGE* (left I right)
  - No relevant lexical information is lost, but recogniser perfomance decreases

Recognition Grammars with Interpretation Tags

- An example:

```
$turn =  [please] turn {$.action="turn"}
    I turn $direction {$.direction=$direction} {$.action="turn"};
$direction= (backlbackward) {"backward"}I $side {$.side=$side};
$side = [to the](left {"left"} I right {"right"})
```

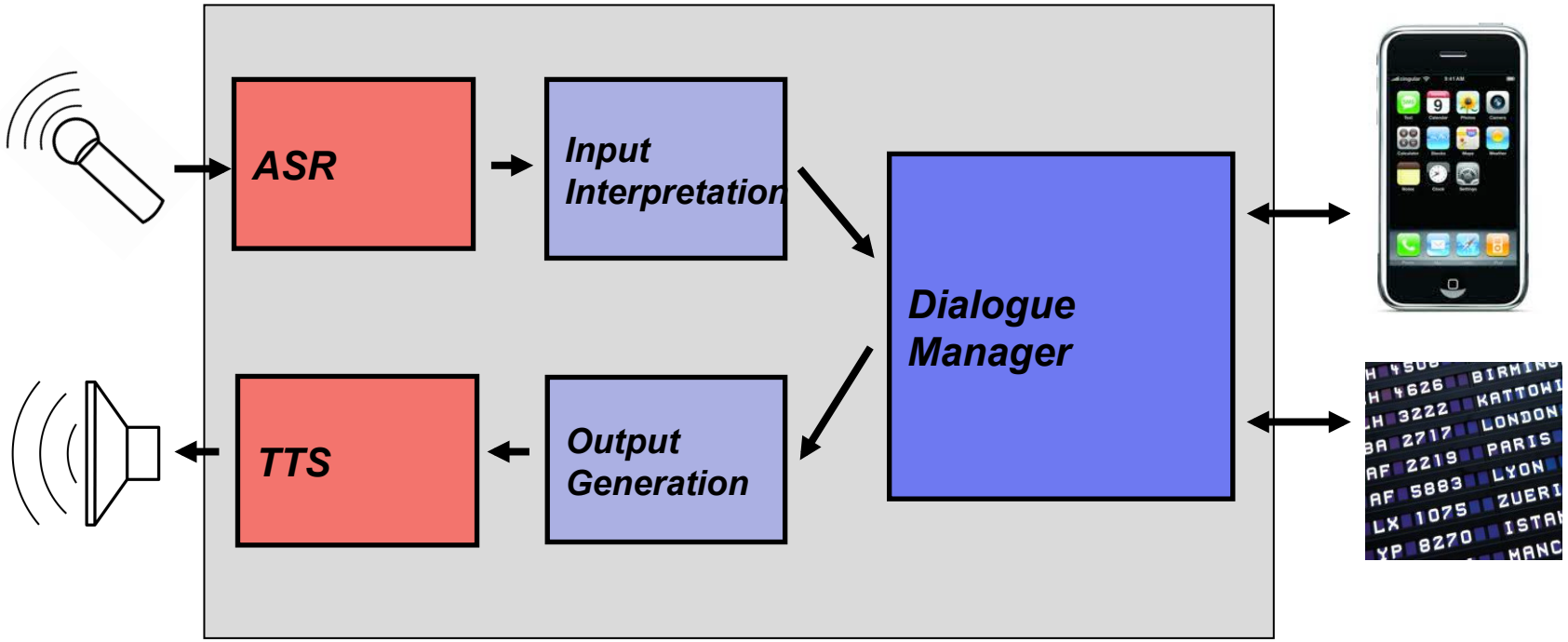
- Recognition grammars with interpretation tags have dual function. They (1) constrain the language model and (2) interpret the recognized input.

Interpretation Grammars for SLMs

- Statistical Language Models (SLMs) are
  - trained on text or transliterated dialogue corpora
  - based on n-gram (typically trigram) probabilities

  Return word-latice with confidences.
- SLMs are permissive with respect to the sequences they (in part erroneously) predict.
- Interpretation grammars for SLMs look like recognition grammars with interpretation tags.
- But they work differently : They parse the speech recognizer output (typically on the best chain)
- Fflexible parsers are needed, which may skip material (assigning a penalty for edits).
- An example: An Earley parser building up a chart, and selecting the best path (w.r.t. the number of omitted words).

Dialog System: Basic Architecture



Output Generation

- Canned text
  - When would you like to leave?
- Template-based generation for speech output:
  - The next flight to $AIRPORT will leave at $DAYTIME.
- Grammar-based generation
  - dialogue act → utterance planner → lexico-syntactic realizer → sentence

    inform(flight(070714;fra;10:30;edi;11:00)) → … → There is a flight on Monday July 7 from Frankfurt to Edinburgh, departing at 10:30, arriving at 11:00 a.m.

Dialog Design: Best Practice Rules

- Do not give too many options at once.
- Guide the user towards responses that maximize
  - clarity and
  - unambiguousness.
- Guide users toward natural ‘in vocabulary’ responses.
  - *How can I help you? vs.*
  - *Which floor do you want to go?*
  - *You can check an account balance, transfer funds, or pay a bill. What would you like to do?*
- Keep prompts brief to encourage the user to be brief.

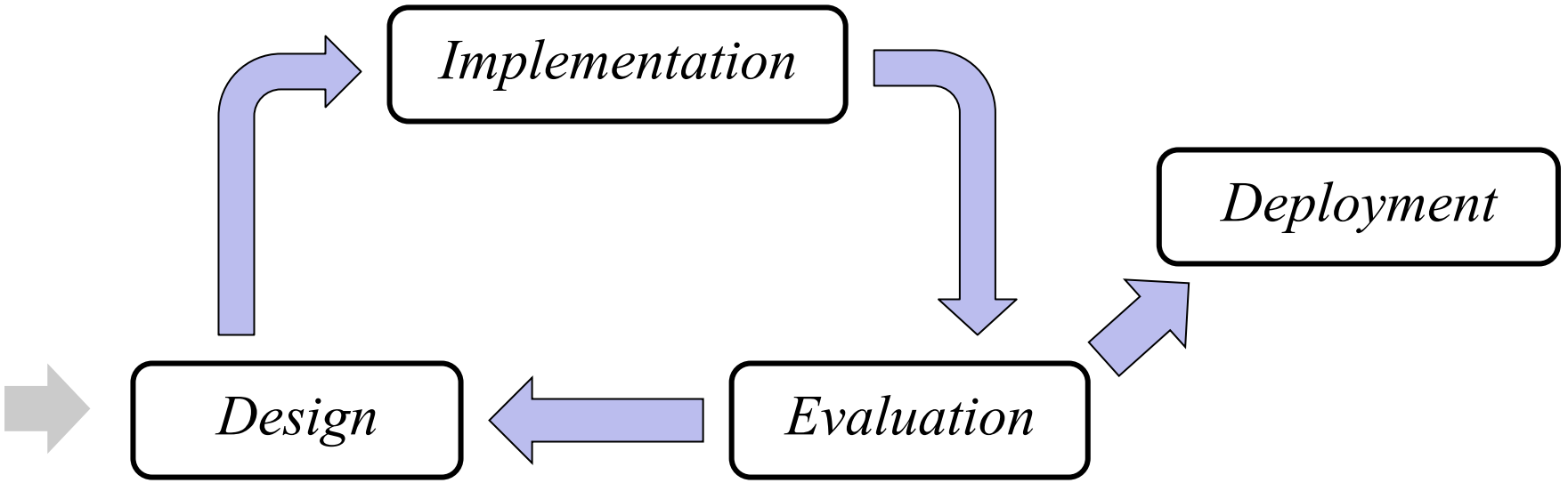
Dialog Design

"...If you'd like to hear all of your options again, press 49. If you've forgotten why you called in the first place, press 50."

Dialog Design: Best Practice Rules

- Allow for the user not knowing
  - the active vocabulary
  - the answer to a question or
  - understanding a question.
- Design graceful recovery when the recognizer makes an error.
- Allow the user to access (context-sensitive) help at any state; provide escape commands.
- Assume errors are the fault of the recognizer, not the user.
- Assume a frequent user will have a rapid learning curve.
- Allow shortcuts:
  - Switch to expert mode/ command level.
  - Combine different steps in one.
  - Barge-In

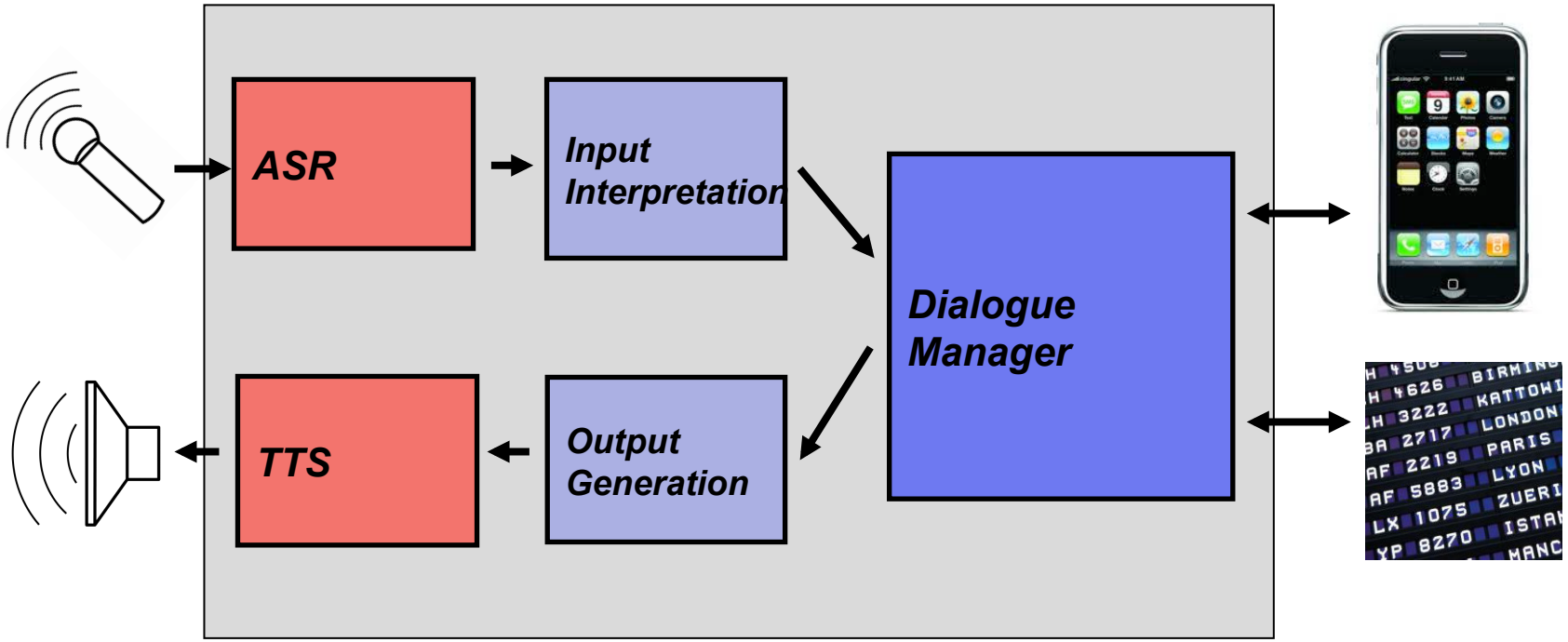
Dialogue Evaluation



Levels of Dialogue Evaluation

- Technical evaluation
- Usability evaluation
- Customer evaluation

Dialog System: Basic Architecture



Technical Evaluation

- Typically component evaluation
- ASR: Word-Error Rate, Concept Error Rate
- NLI: precision, recall
- TTS: Intelligibility, Pleasantness, Naturalness
- NLG: correctness, contextual appropriateness
- Linguistic Coverage: out of vocabulary, out of grammar rates (for in-domain user input)
- Dialogue flow, turn level: Frequency of timeouts, overlaps, rejects, help requests, barge-ins

Levels of Dialogue Evaluation

- Technical evaluation
- Usability evaluation
- Customer evaluation

Usability Evaluation

- Typically an end-to-end “black box” evaluation

- Main criteria are:
  - Effectiveness (Are dialogue goals fully/partially accomplished?)
  - Efficiency (Dialogue duration? Number of turns?)
  - User satisfaction

Evaluation of User Satisfaction

- SASSI („Subjective Assessment of Speech System Interfaces“): A Conceptual Framework for designing User Questionnaires
- Dimensions of user satisfaction:
  - System Response Accuracy: User‘s perception of the system as accurate and doing what they expect
  - Likeability: User‘s rating of the system as useful, pleasant, friendly
  - Cognitive demand: The perceived amount of effort needed to interact with the system and feelings arising from this effort
  - Annoyance: User‘s rating of the system as repetitive, boring, irritating, and frustrating
  - Habitability: The extent to which users knew what to do and what the system was doing
  - Speed: How quickly the system responded to user inputs

Levels of Dialogue Evaluation

- Technical evaluation
- Usability evaluation
- Customer evaluation

Customer Evaluation

- Costs
- Platform compatibility
- Maintenance properties
- Scalability
- Portability

Example: The TALK Project


Titel des Albums Mensch
Titel
1 Mensch
2 Neuland
3 Der Weg
4 Viertel Vor
5 Lache Wenn Es Nicht Zum Weine..
6 Unbewohnt
Mensch
DTC
19.0°
19.5°
MAX
AUTO
REST
BMW NAVIGATION PROFESSIONAL
DVD

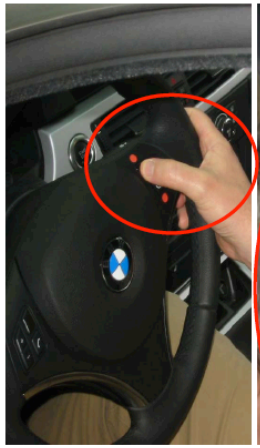
TALK Evaluation

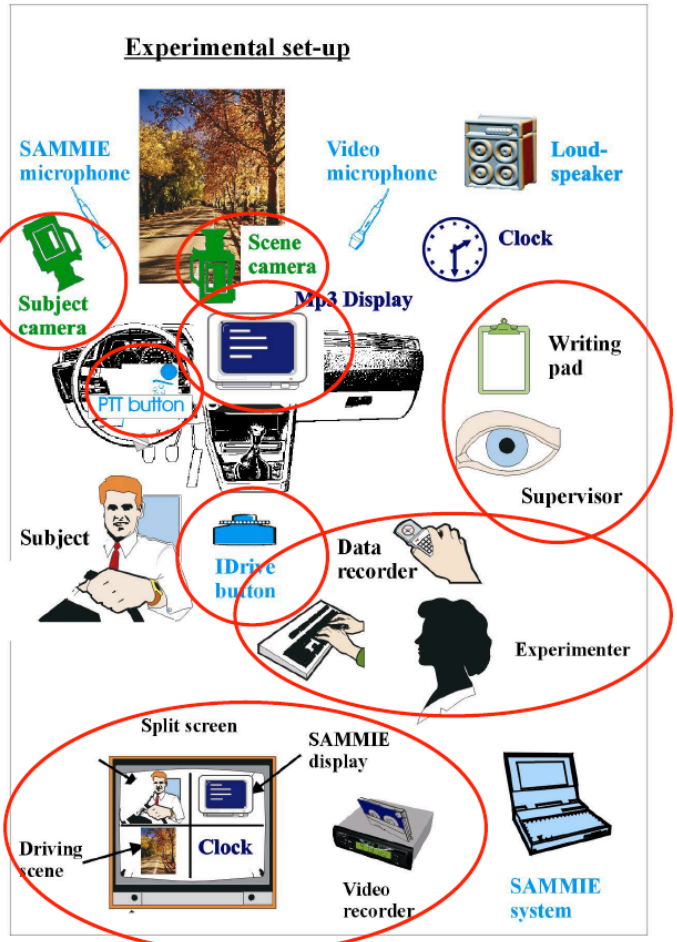
TALK Evaluation

- Sample of 21 subjects
  - 11 from TALK baseline evaluation 2005
  - 6 from other experiments (VICO, other)
  - 4 new
- 7 female / 14 male
- Average age 36,2 (20 - 50)
- Some / much MP3 experience
- Enough driving experience for safety reasons
- One experimental session lasted 3 hours, i.e. 2 subjects / day

TALK Evaluation

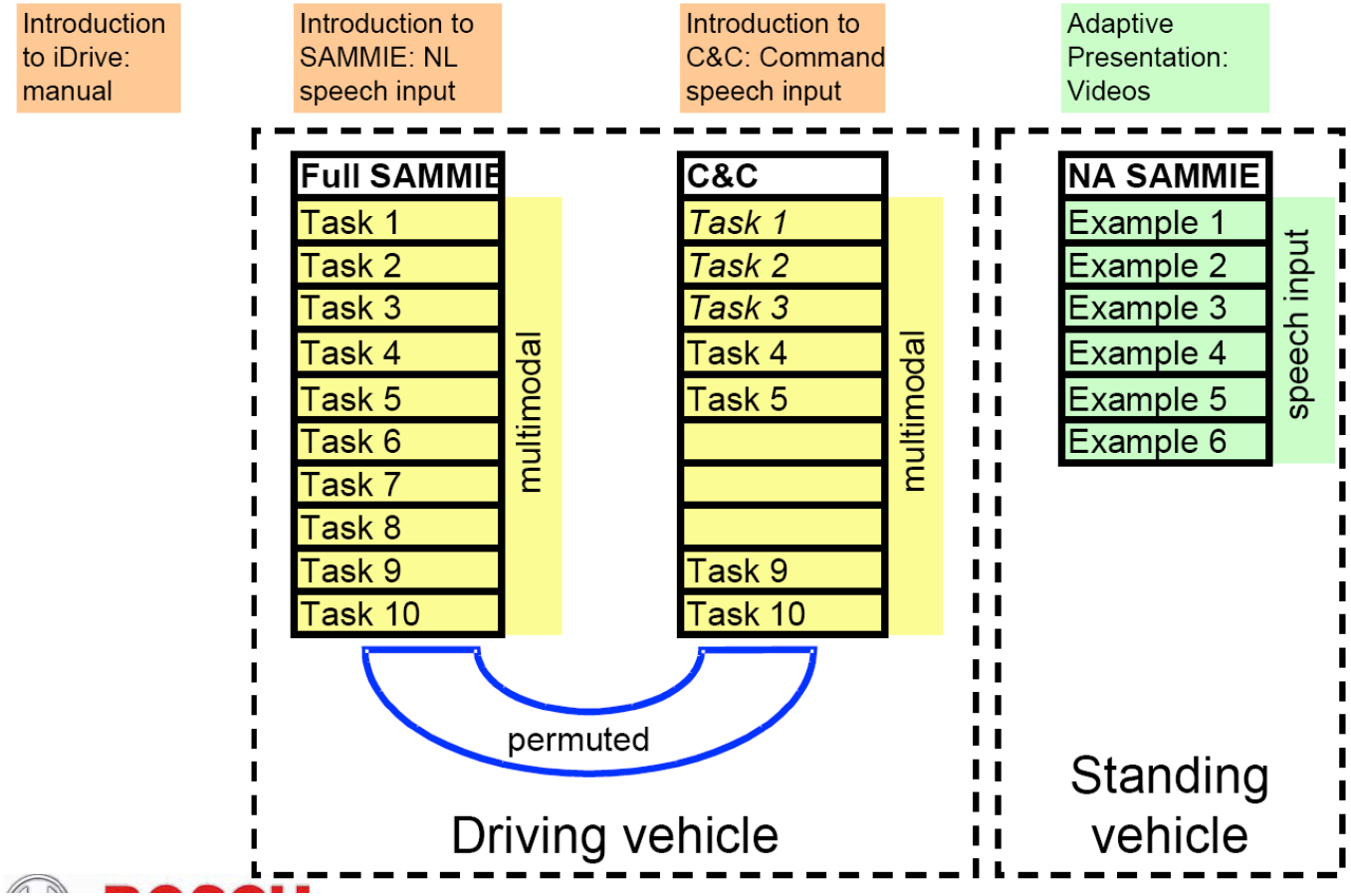
Setup in the BMW test car





TALK Evaluation

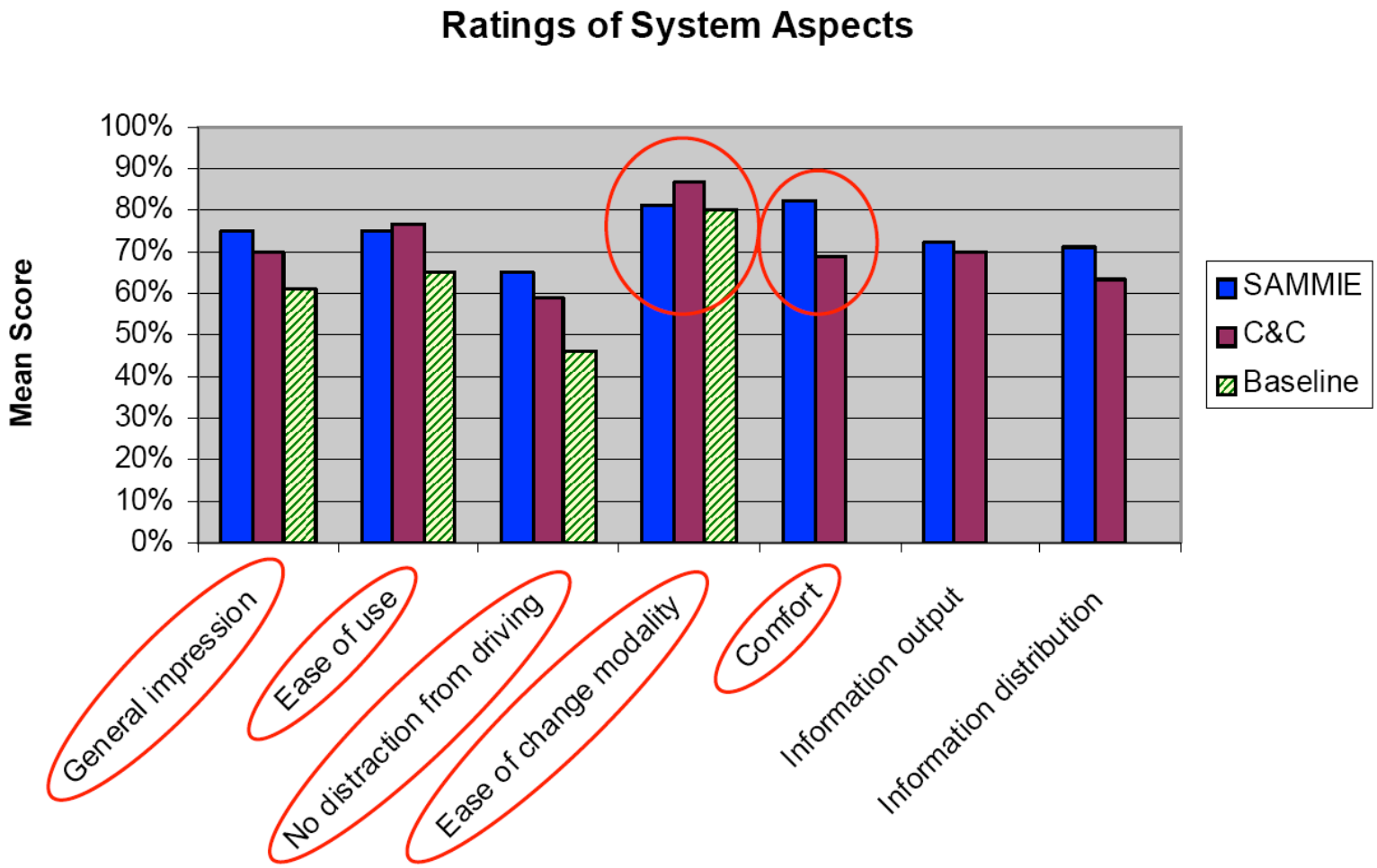
Experimental Session



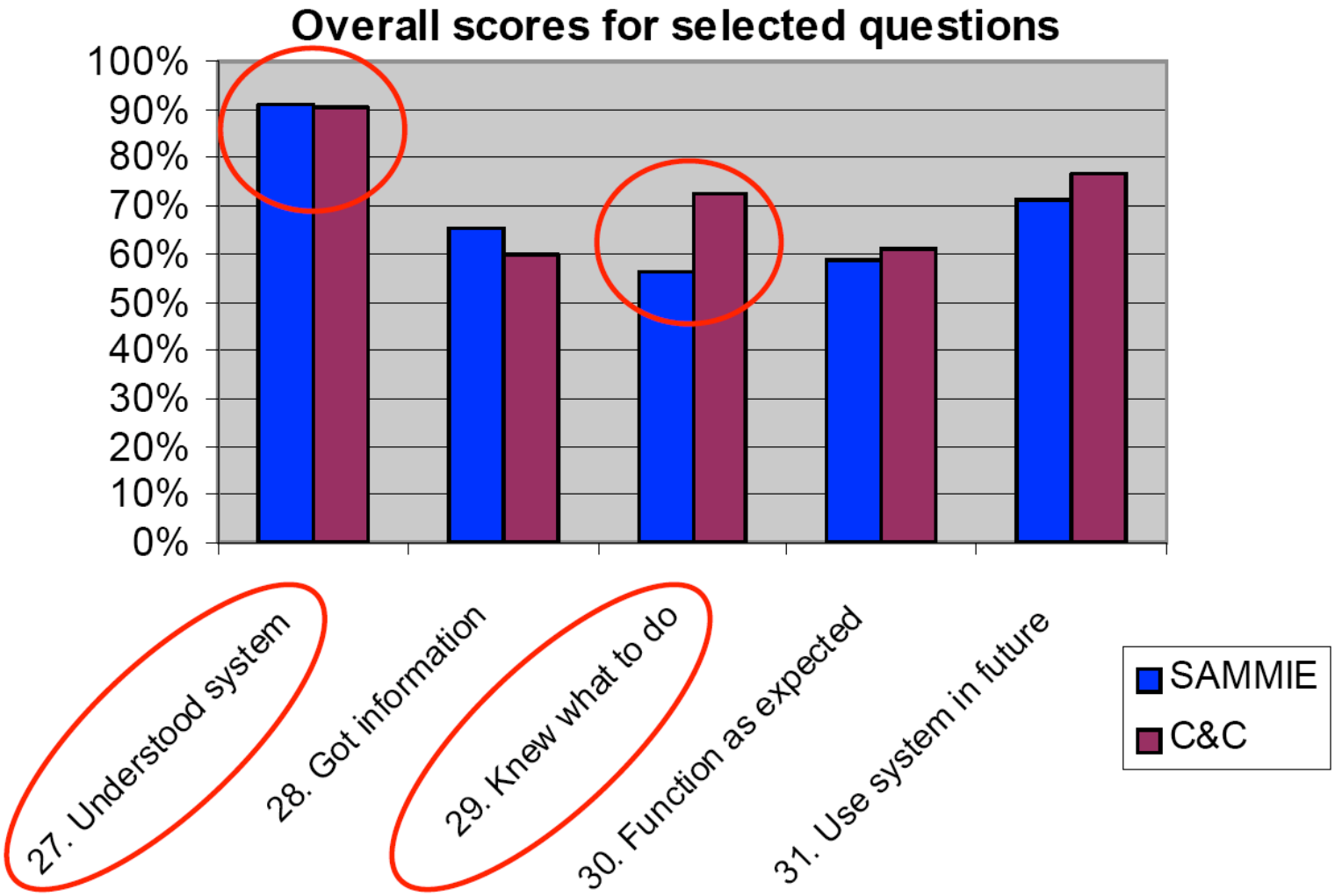
10 Dialogue Tasks

- 1. Ask for the existing albums
- 2. Play back the song ´Der Weg´by ´Herbert Grönemeyer´
- 3. Find out the songs on the playlist ´Pur Klassiker ´
- 4. Browse and search for the album ´Live´von ´Pur ´and play it back
- 5. Find and play back a Swing song by ´Michael Buble´
- ...

TALK Evaluation



TALK Evaluation




tALK
M NZ 4382