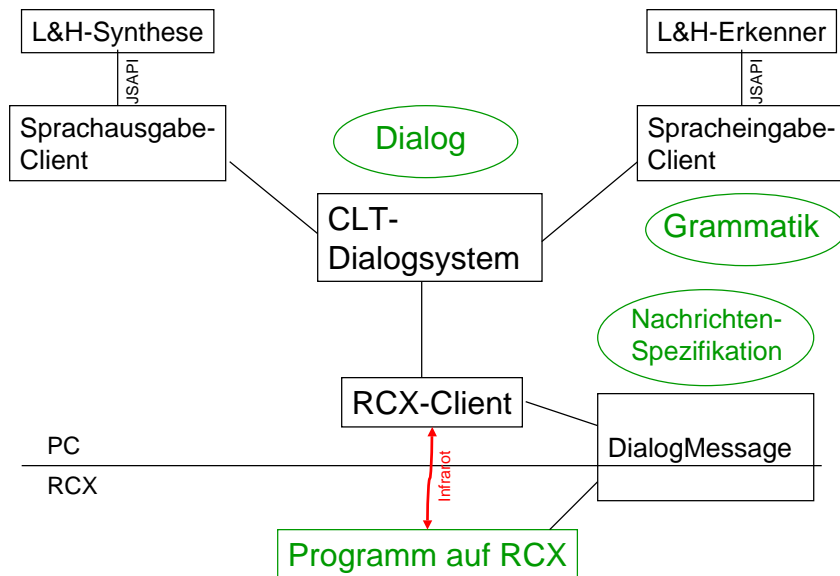


Dialog

Alexander Koller

Softwareprojekt "Sprechende Roboter"
21. Mai 2004

Struktur des Systems auf dem PC



Der RCX-Client

- ◆ Kommunikation zwischen Dialogsystem und RCX.
- ◆ Nachrichtenspezifikation wird mit "ant project-pc" in DialogMessage-XXX.jar kompiliert.
- ◆ Start des Clients mit

```
java -cp build/lib/DialogMessage-xxx.jar;  
      build/lib/RCXClient.jar  
      de.saar.coli.lego.clients.rcxclient.RCXClient
```
- ◆ Zusätzliches Argument "--dummy": Kommunikation mit RCX wird nur simuliert.

Der GUISpeech-Client

- ◆ Kommunikation zwischen Dialogsystem und Spracherkenner/-synthese.
- ◆ Lädt Grammatik im JSGF-Format (Java Speech Grammar Format).
- ◆ Kann Strings per Synthese aussprechen oder direkt WAV-Dateien abspielen.
- ◆ Beispiele:
 - projects/Lift/fahrstuhl.gram
 - projects/Chess/chess-cont.gram

Starten von GUISpeech

- ◆ Starten mit

```
java -jar build/lib/GUISpeech.jar  
<Name Grammatikdatei>
```

- ◆ Im aktuellen Verzeichnis muss eine Datei "speech.props" liegen: Aus projects/Lift oder projects/Chess kopieren.

Diamant

- ◆ Das CLT-Dialogsystem wird gestartet z.B. mit "Diamant.exe" im Verzeichnis der CLT-Distribution.
- ◆ Beispieldialoge:
 - projects/Lift/lift.xml
 - projects/Chess/chess-cont.xml

Diamant und Clients

- ◆ Diamant muss sich mit den beiden Clients verbinden. Dazu als Geräte eintragen:
 - RCX-Client auf Port 1234
 - GUISpeech auf Port 1236.

Patterns in Diamant

- ◆ Diamant schickt einfach Strings ("...") an den Sprach-Client.
- ◆ Sprach-Client schickt entweder Strings oder Strukturen zurück -- je nach Tags in der Grammatik.
- ◆ Matching von Strings mit Patterns wie z.B. "abcd".
- ◆ Speech-Client kann Strukturen zurückgeben: Matching mit
$$\{l_1=p_1, l_2=p_2, \dots, l_n=p_n\}$$
- ◆ p_1, \dots, p_n sind wieder Patterns, also z.B. Strings mit "..." oder Variablennamen.

Nachrichten RCX-Client

- ◆ Kommunikation mit RCX-Client geht über Tabellen, die der Nachrichten-Spezifikation gehorchen.
- ◆ Tabelle hat einen Eintrag "type", in dem der Typ steht, sowie einen Eintrag für jedes Argument der Nachricht.
- ◆ Werte der Einträge sind Strings -- also entweder eine Zahl als String oder der Name des Atoms.
- ◆ Konversion Zahl in String: `toStr(zahl)` (siehe Hilfe/Funktionen).

Starter

- ◆ Um alle drei Komponenten des Lego-Systems bequem zu starten, verwende

```
java -jar build/lib/Starter <dateiname>
```
- ◆ In Datei werden Arbeitsverzeichnisse, Programmnamen usw. übergeben.
- ◆ Verwende Variablen aus `declarations.xml`.
- ◆ Beispiel:
 - `starter/lift.xml`

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.