



Das Dialog Projekt

WS 01/02 bis SS 03

Rückblick auf 2 Jahre Projektarbeit

Grobe Gliederung der Präsentation

- Inhalt des Projektes
 - Szenarien
 - Systemaufbau
 - Versuche
- Erläuterungen zu Nuance
 - Nuance Communications
 - Nuance Grammar Builder
 - Nutzung des Erkenners
 - Beispielanwendung
- Subjektive Einschätzung
 - Erfahrungen und Tipps

Personen im Projekt Dialog

- Projektleiter/betreuer:
 - Prof. Dr. B. Krieg-Brückner
 - Prof. Dr. J. Bateman
 - Dr. R. Moratz
 - T. Vierhuff

- Projektteilnehmer:

Thomas Berndt, Philip Büschenfeld,
Matthias Kay, Daniil Liberman,
Andreas Meyer, Jan Plagge,
Sören Schwertfeger, Andi Winterboer

Einführung

- ❑ Entwicklung eines natürlichsprachlichen Dialog-Systems zur Steuerung von mobilen Robotern
- ❑ Übertragung der adaptiven Fähigkeiten des Menschen auf ein System durch geschickte Dialogmodellierung
- ❑ wissenschaftliche Experimente und umfangreiche Evaluation haben zur Analyse und Verbesserung unseres Systems beigetragen

Mögliche Einsatzgebiete

- ☐ Ältere Menschen
- ☐ Behinderte Menschen
- ☐ Vereinfachung der Bedienung von komplexen technischen Geräten
- ☐ Hand- oder Eye-Busy-Anwendungen

Szenarien - I

Pioneer

- ☐ Mobiler Roboter
- ☐ Laser-Range-Finder
- ☐ Differenzierung von Objekten
- ☐ Ziel: per Sprachbefehl zu einem bestimmten Objekt fahren

Aibo (Sony)

- ☐ Mobiler Roboter
- ☐ Kamera
- ☐ Kann neben Farben auch Formen differenzieren
- ☐ Ziel: s. Pioneer

Szenarien - II

Rolland

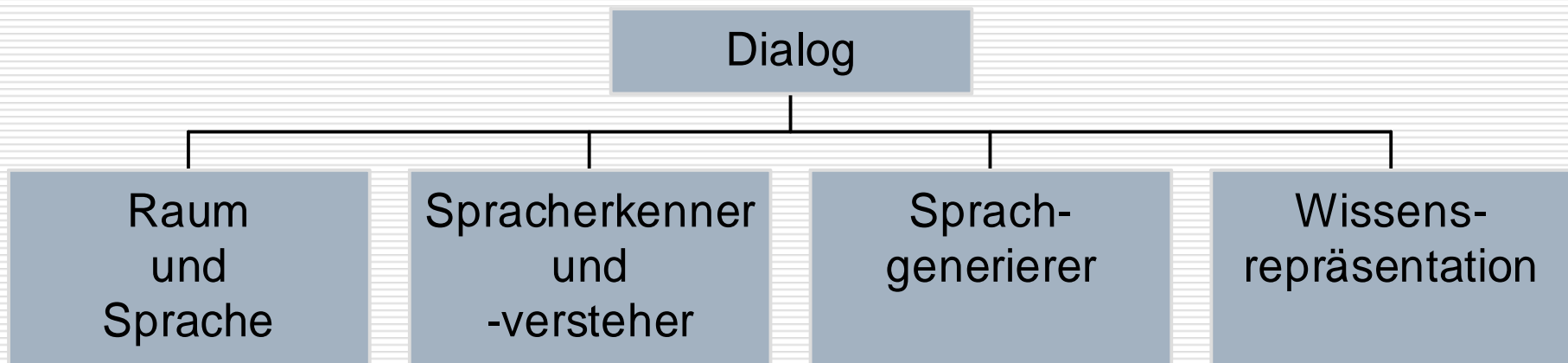
- Semi-autonomer Rollstuhl
- Ziel: bestimmte Befehle per Sprache ausführbar machen

Autoradio (Becker)

- Auto-Navigations- bzw. Telekommunikation
- Ziel: Entlastung des Fahrers durch Sprachbefehle

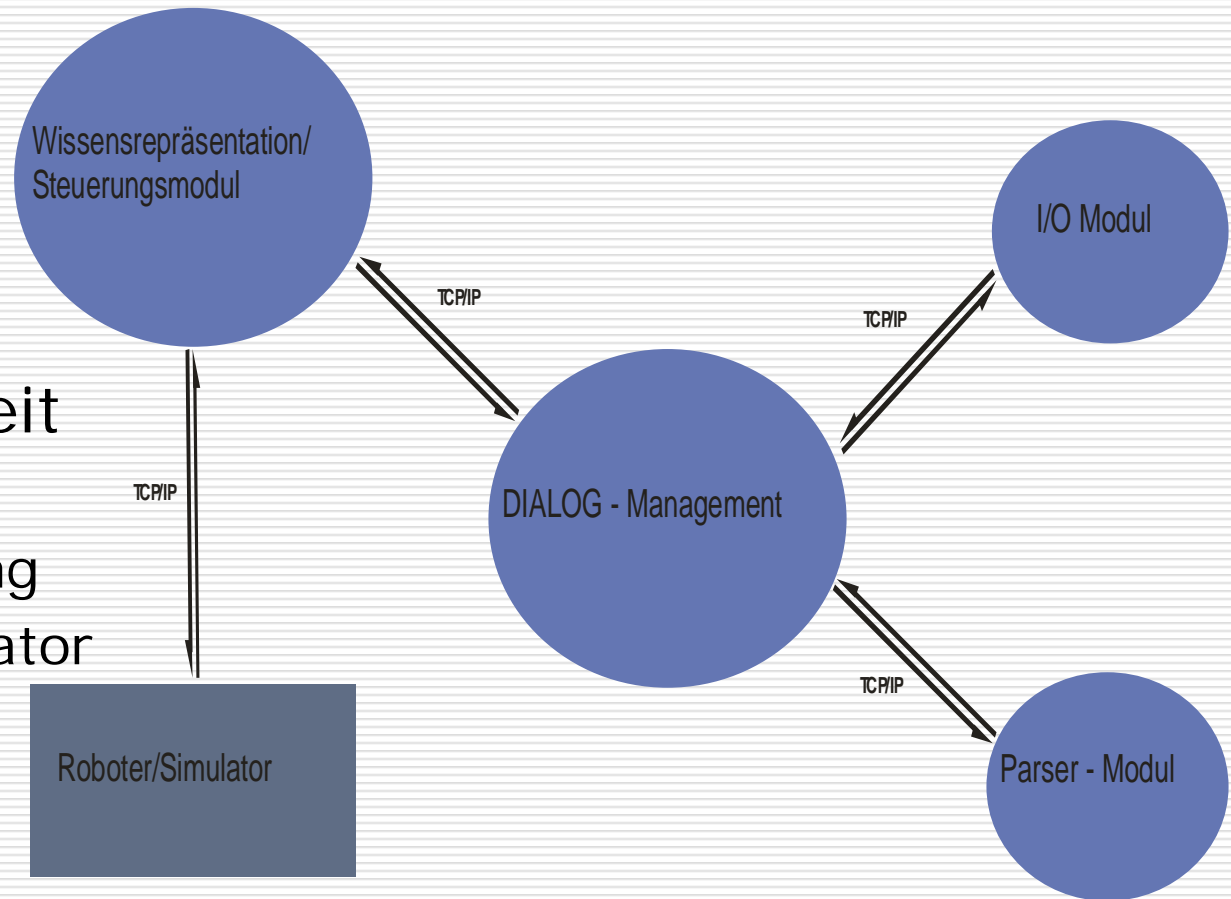
Themengebiete des Dialogprojekts

Grundlagen



Modularer Systemaufbau

- Vier Module
- Versenden XML-Daten über TCP/IP
- Austauschbarkeit der Module:
 - Sprachänderung
 - Roboter/Simulator
 - Text-/ Sprach-IO



Kommunikation im Dialogsystem

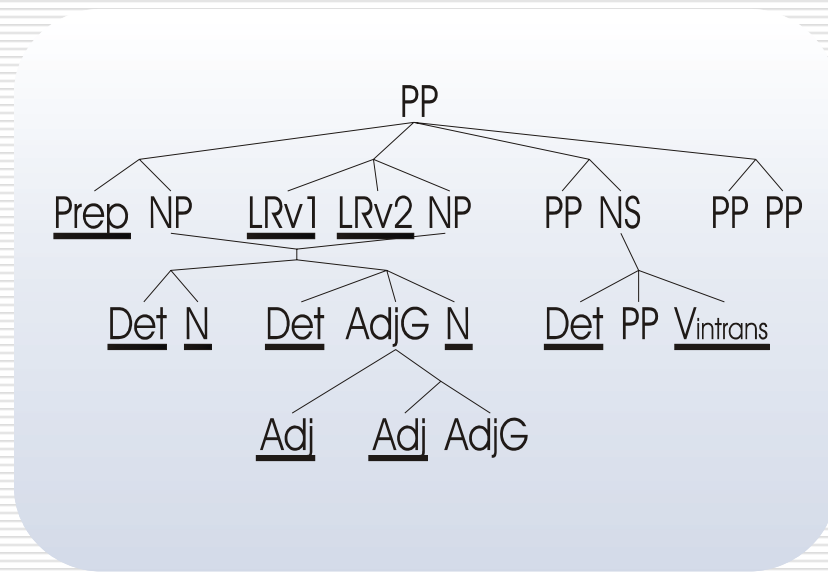
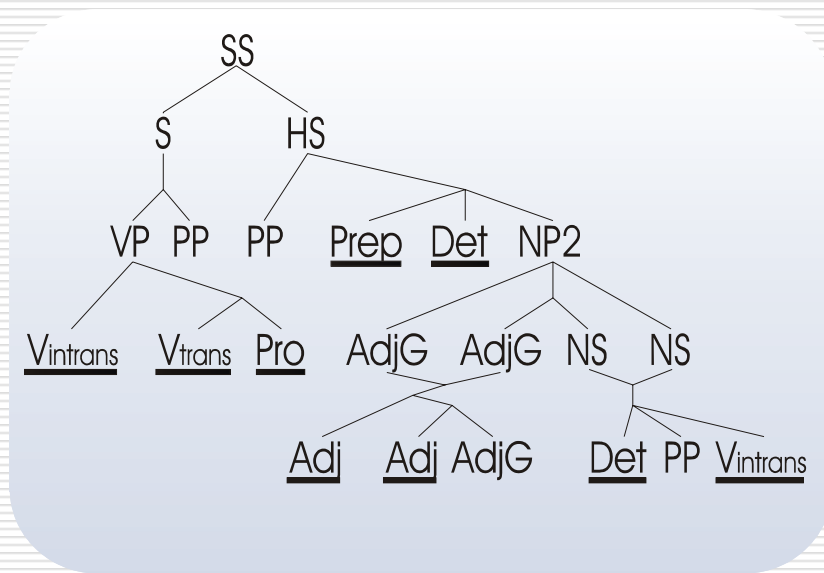
- Kommunikation mit XML via TCP/ IP
 - Betriebssystemunabhängigkeit
 - Programmiersprachenunabhängigkeit
 - Menschenlesbarkeit
 - Austauschbarkeit der Dialog-Module

Erkennung I

- Gesprochene Eingaben
 - Nuance-Spracherkennung
 - Sprecherunabhängig
 - Erkennung mittels vordefinierter Grammatik bzw. Lexikon
 - Umwandlung des Signals in Zeichenkette
- Getippte Eingaben
 - Selbstimplementierte Benutzerschnittstelle
- Senden der Zeichenkette zur syntaktischen Erkennung (via Dialogmanagement)

Erkennung II

□ Syntaktische Erkennung



Erkennung III

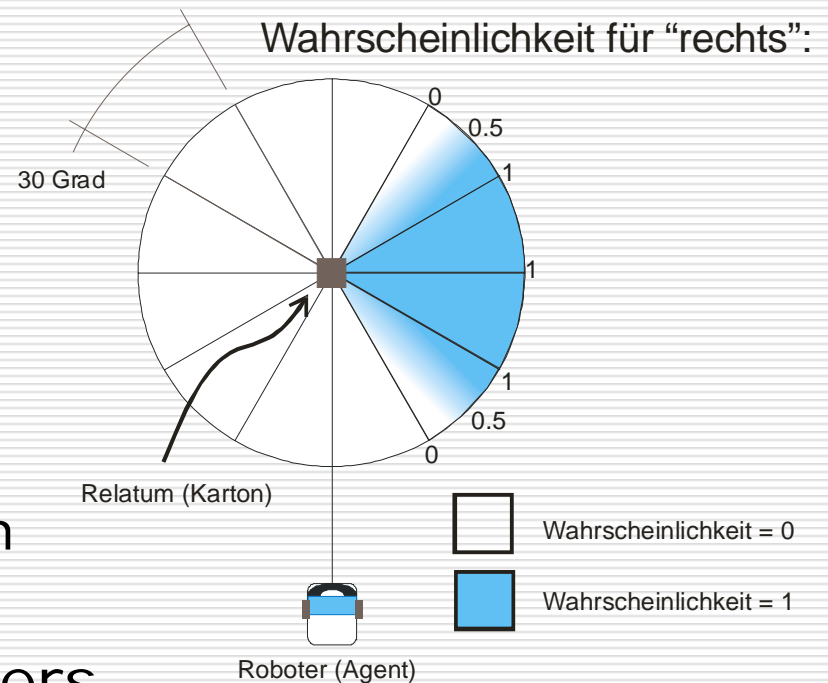
- Semantische Erkennung
 - Traversal des syntaktischen Baumes
 - Aus Wortart und Wort werden Objekte generiert
 - Erkannte Objekte werden dem Topelement des Stacks hinzugefügt
 - Danach werden die Objekte auf den Stack gelegt
 - Instruktion wird an Wissensrepräsentation übergeben (via Dialogmanagement)

Dialog-Management

- ❑ Stellt Serverdienste zur Verfügung
- ❑ Datenübertragung zwischen den Modulen des Systems
- ❑ Steuerung des Dialogs
- ❑ Textgenerierung
- ❑ Ausgabe entweder als Text oder als Sprache

Wissensrepräsentation

- Räumliches Schließen
 - Auffindung des Objekts über ein Bezugsobjekt
 - Gleichsetzen der vom Benutzer erwähnten Objekte mit den Laserscanner-Objekten
 - Test der Gleichsetzungen anhand des Modells
- Ansteuerung des Roboters



Experimente

- ❑ Fachübergreifend mit Linguisten
- ❑ ca. 50 Versuchspersonen in 2 Versuchsreihen
- ❑ 14 verschiedene Versuchskonfigurationen, also Anordnungen der Objekte
- ❑ mehr als 350 Testläufe
- ❑ Auswertung per Video, Logfiles und Protokoll

Ergebnisse der Versuche

- ❑ größere Probleme bei der Steuerung bei technisch versierten Probanden
- ❑ bei Problemen Tendenz zu immer simpleren Anweisungen
- ❑ Erfolgreiche Formulierungen werden beibehalten und sehr wenig variiert
- ❑ Versuchspersonen haben keine Probleme bei der räumlichen Orientierung

Erläuterungen zu Nuance

Firmenprofil von Nuance Communications

- ❑ 1994 gegr. von ehem. Mitarbeitern des SRI International (Stanford Research Institute)
- ❑ Jahresumsatz 2003 52 Mio. Euro
- ❑ Nuance lieferte weltweit erste Software-Plattform, die für Sprachlösungen optimiert war
- ❑ Gilt heute als Weltmarktführer bei den sprachbasierten Telefondiensten

Von Nuance angebotene Anwendungen und Lösungen

- ☐ Stimmauthentifizierung
- ☐ Text-to-Speech-Systeme
- ☐ Interactive Voice Response Systems (IVR) – sprachgesteuerte Telefondirektwahl
- ☐ Self-Service-Banking
- ☐ Sprachgesteuertes Customer-Relationship-Management
- ☐ Sprachgesteuerte Reservierungssysteme
- ☐ Natürlichsprachliche Dialogsysteme

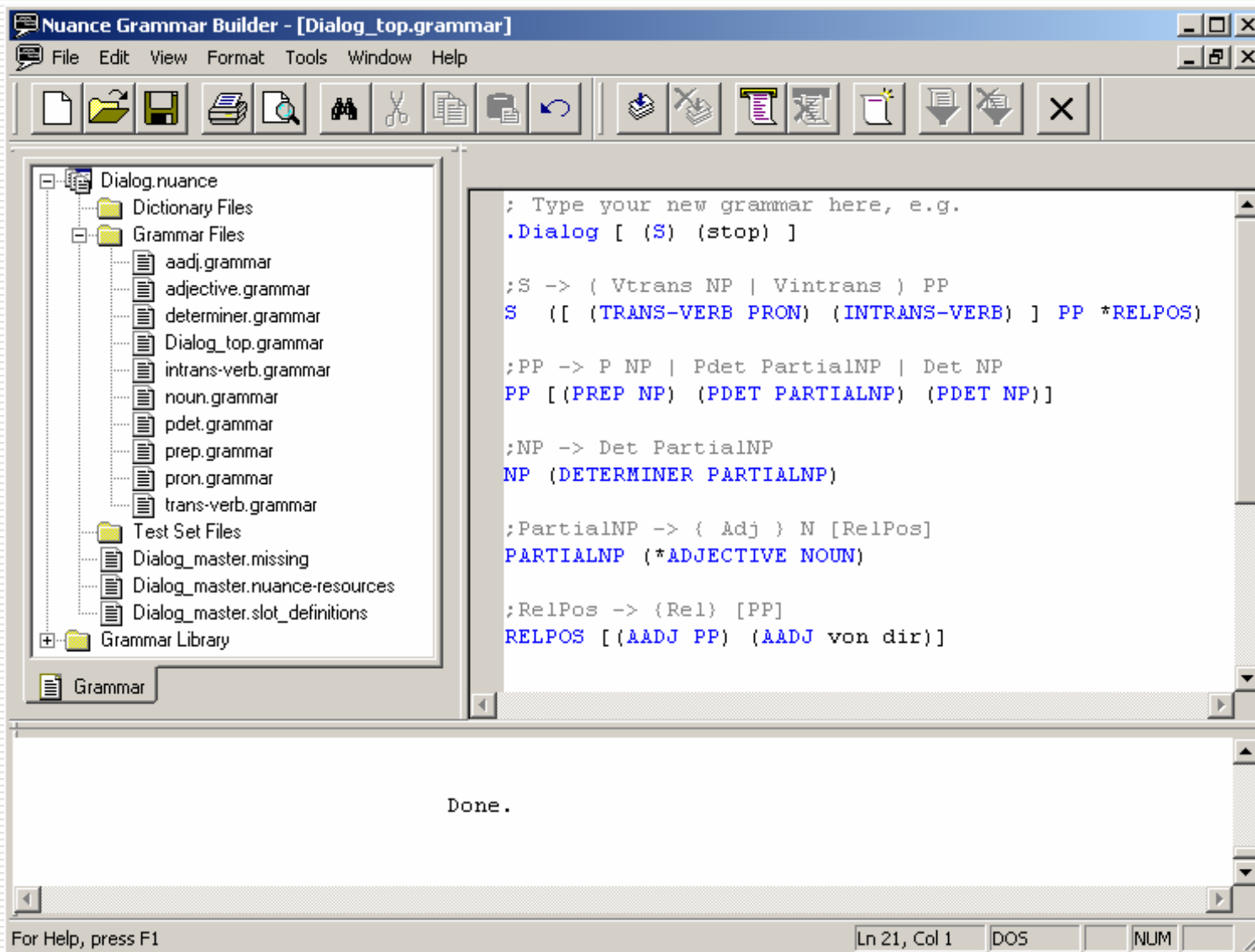
Funktionsweise von Spracherkennung (Kurzform)

- ❑ Sprache wird in ein Mikrofon gesprochen
- ❑ Innerhalb des Computers zu einem Erkennen transportiert
- ❑ In viele Stücke geteilt, die Phoneme heißen
- ❑ Diese werden dann identifiziert und mit einer Liste verglichen, die die Software zur Verfügung stellt
- ❑ Die als am wahrscheinlichsten eingeschätzte Möglichkeit wird dann verwendet

Grammatiken in Nuance mit GSL - Grammar Specification Language

- Grammatik bietet die Möglichkeit, erwartete Phrasen zu definieren
- Dabei muss antizipiert werden, was eventuell gesagt werden kann und in welcher Reihenfolge
- Alle vermutlich verwendeten Wörter müssen in den Subgrammars auftauchen
- Einfache Multiple Choice Grammatiken:
 - schnelle Erkennung
 - geringe Mächtigkeit
 - z.B. Banking
- komplexe Grammatiken:
 - Modellieren ganze Sätze
 - Lange Erkennungszeit
 - Benutzerfreundlich

Nuance Grammar Builder



Dialog Grammatik I

```
.Dialog [ (S) (stop) ]

;S -> ( Vtrans NP | Vintrans ) PP
S    ([ (TRANS-VERB PRON) (INTRANS-VERB) ] PP *RELPOS)

;PP -> P NP | Pdet PartialNP | Det NP
PP   [(PREP NP) (PDET PARTIALNP) (PDET NP)]

;NP -> Det PartialNP
NP   (DETERMINER PARTIALNP)

;PartialNP -> { Adj } N [RelPos]
PARTIALNP (*ADJECTIVE NOUN)

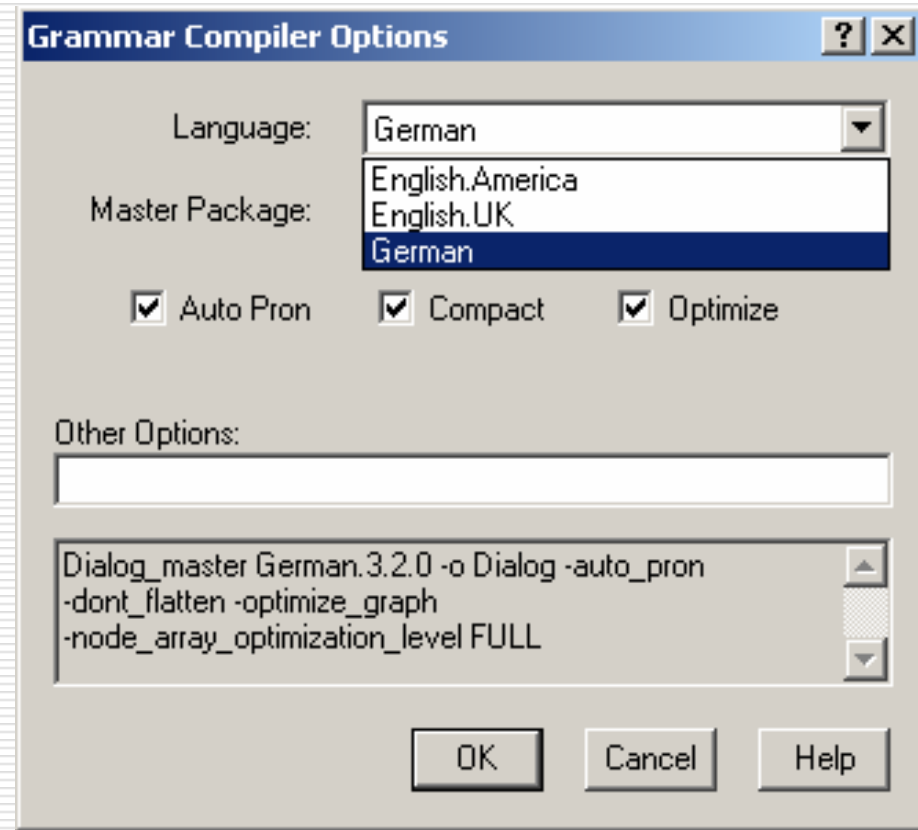
;RelPos -> { Rel } [PP]
RELPOS [(AADJ PP) (AADJ von dir)]
```


Dialog Grammatik: Substantiv

```
NOUN [ box      {< noun "Box">}
        zylinder {< noun "Zylinder">}
        würfel   {< noun "Würfel">}
        klotz     {< noun "Klotz">}
        kasten   {< noun "Kasten">}
        objekt    {< noun "Objekt">}
        tonne     {< noun "Tonne">}
        eimer     {< noun "Eimer">}
        drehung   {< noun "Drehung">}
        mülleimer {< noun "Mülleimer">}
        kiste     {< noun "Kiste">}
        karton    {< noun "Karton">}
        röhre     {< noun "Röhre">}
        säule     {< noun "Säule">}
        start     {< noun "start">}
        flasche   {< noun "Flasche">}
    ]
```

Nuance Grammatik kompilieren

- ❑ dauert ziemlich lange
- ❑ Auswahl der Sprache
- ❑ Verschiedene Optionen
- ❑ Auto Pron(unciation):
für unbekannte
Wörter Aussprache
generieren



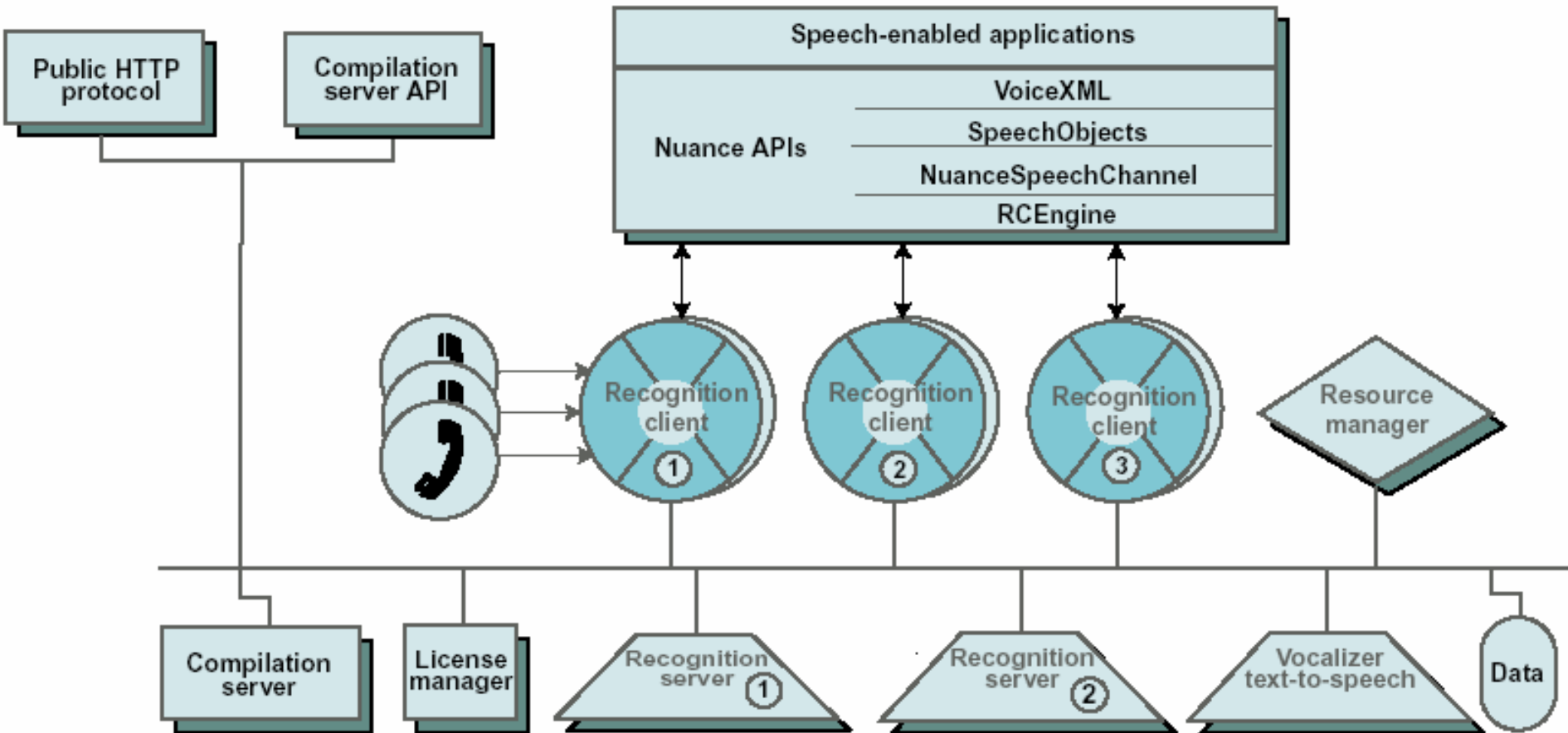
Nuance benutzen I

- ❑ Nuance Licence Manager mit gültiger Lizenz starten:
 - `nlm ntk8-800-g-x26-33224222`
- ❑ Nuance Recognition Server mit kompilierter Grammatik starten:
 - `recserver -package .\speechIO\Dialog`
`Im.Addresses=localhost`
 - Recserver leistet Erkennung des Satzes anhand der Phoneme
 - `Im.Addresses` : Adresse des Licence Managers
 - Eigendlicher Sprachverstehrer
- ❑ Vocalizer für Sprachausgabe starten:
 - `vocalizer.exe -gender Male -language German`

Nuance benutzen II

- Anwendung mit Client starten:
 - `java NuanceExplain -package .\Dialog Im.Addresses=localhost
audio.Provider=native client.NoSpeechTimeoutSecs=2
client.TTSAddresses=localhost`
 - `client.NoSpeechTimeoutSecs`: Timeout bei Erkennung
 - `client.TTSAddresses`: Adresse des Text-to-Speech-Servers
 - `audio.Provider`: Eingabe der Sprachdaten per Soundkarte
 - Nuance eigentlich als Telefonsprachdienstleister =>
Telefonkarten als Hardware => schlechter
Soundkartensupport
 - Client sendet Sprachdaten zum Recognition-Server bzw.
zum TTS-Server

Nuance Architektur



Nuance Client Code I

```
1  import vcommerce.core.util.*;
2  import vcommerce.core.sc.*;
3  import nuance.core.sc.*;
4  import nuance.core.util.*;
5
6  class NuanceExplain{
7      public static void main(String args[]){
8
9          CorePromptPlayer player;
10         NuanceSpeechChannel nsc = null;
11         String[] arg = {"client.TTSAddresses=localhost"};
12         try{
13             // Create a NuanceConfig object from the command-line arguments
14             NuanceConfig config = new NuanceConfig();
15             String[] extra_args = config.buildFromCommandLine(args);
16             // Create the SpeechChannel object
17             nsc = new NuanceSpeechChannel(config);
18             player = nsc.getPromptPlayer();
```

Nuance Client Code II

```
19         while(true){
20             RecResult rec_result = nsc.playAndRecognize(".Dialog");
21             if (rec_result.getNumResults() > 0) {
22                 SingleResult sr = rec_result.getSingleResult(0);
23                 String erkannt = sr.getRecognizedString();
24                 System.out.println(erkannt);
25                 if(!erkannt.startsWith("<")) player.appendTTS(erkannt);
26             }
27         }
28     } catch(Exception e){
29         /*Errormeldung*/
30         return;
31     }
32 }
33 }
```

Unsere Erfahrungen mit Nuance

☐ Positiv

- Sehr gute Erkennungsrate
- Tatsächliche Sprecherunabhängigkeit
- Akzeptiert viele Nebengeräusche
- Relativ schnelle Erkennungszeiten
- kostenlos

☐ Negativ

- Läuft nur auf wenigen PC's
- Sehr betriebssystemabhängig (Windows 2000)
- Schlechter Support

Projekterfahrungen

Gruppen- bzw. Projektarbeit allgemein

- Neue Anforderungen in der Wirtschaft: Soft Skills
 - Teamarbeit
 - mehreren Aufgabe parallel bearbeiten
 - Vernetzt Denken
- Gruppen nicht nur nach Freundschaft aussuchen – besser:
 - homogene Gruppen
 - Interessensgebiete berücksichtigen

Gruppenarbeit – unsere Tipps + Erfahrungen

Tipps

- ❑ “Orga-Team” bestimmen
- ❑ Sinnvolle “Themen”-Gruppen bilden
- ❑ Frühzeitig Deadlines setzen
- ❑ CVS einsetzen
- ❑ Kontinuierliches Arbeiten am Projektbericht
- ❑ Modulprüfungen so schnell wie möglich

Erfahrungen

- ❑ Keine Zeitverschwendung für später nicht benutzte Tools
- ❑ Kein Semester ist zuviel, immer am Ball bleiben
- ❑ Projektwochenenden waren produktiv und unterhaltsam: Stärkung des Gruppenzusammenhaltes

Fragen?

Wir freuen uns über Fragen!