



Bildverarbeitung

NAO Gesichtserkennung

Daniel Schäfer, Hendrik Schlee





Inhalte

- Motivation
- Grundlagen
- Gesamtmodell
- NAO Framework & Sdk
- Face-Vacs
- Further Work





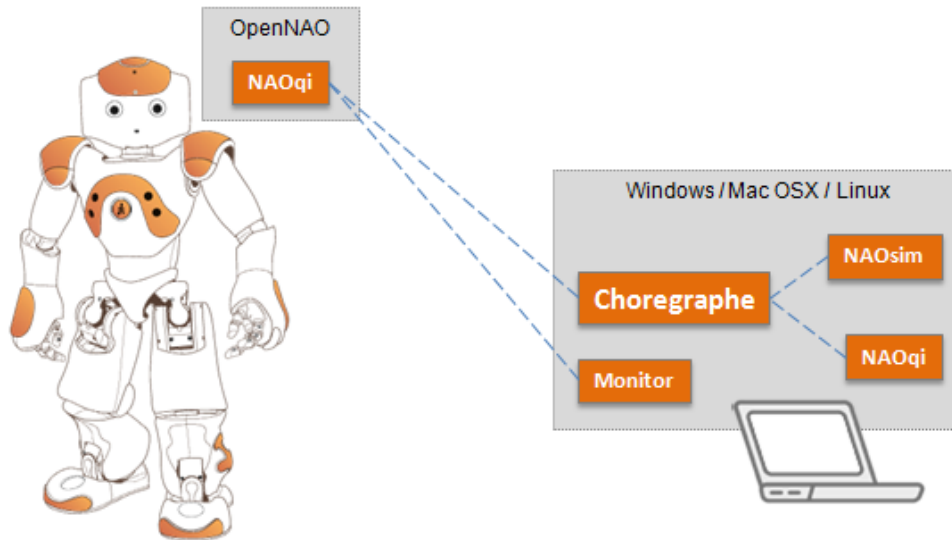
Motivation

- **FaceReco:** Remote Anbindung an den NAO zur Gesichtserkennung mit dem Erkennungssystem FaceVACS SDK des Weltführers Cognitec System
- **NAO-System:**
 - Bilderkennung, Bildverfolgung auf dem Nao ist nicht störungsfrei
 - Begrenzte Rechenleistung
- **Nutzen:**
 - Gesichter Problemlos erkennen, wiedererkennen und evtl. eine Kommunikation starten
 - Robustheit erhöhen, Fehlerrate zu minimieren, leichtere Steuerbarkeit

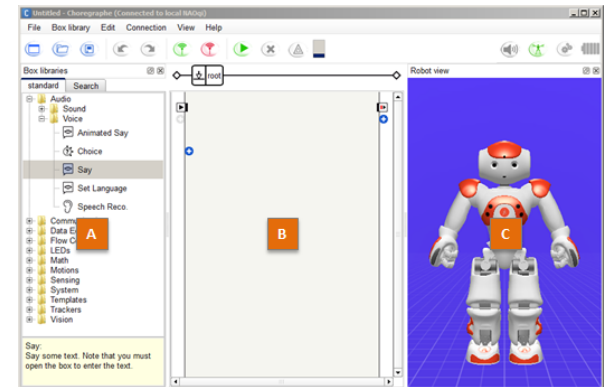




Grundlagen



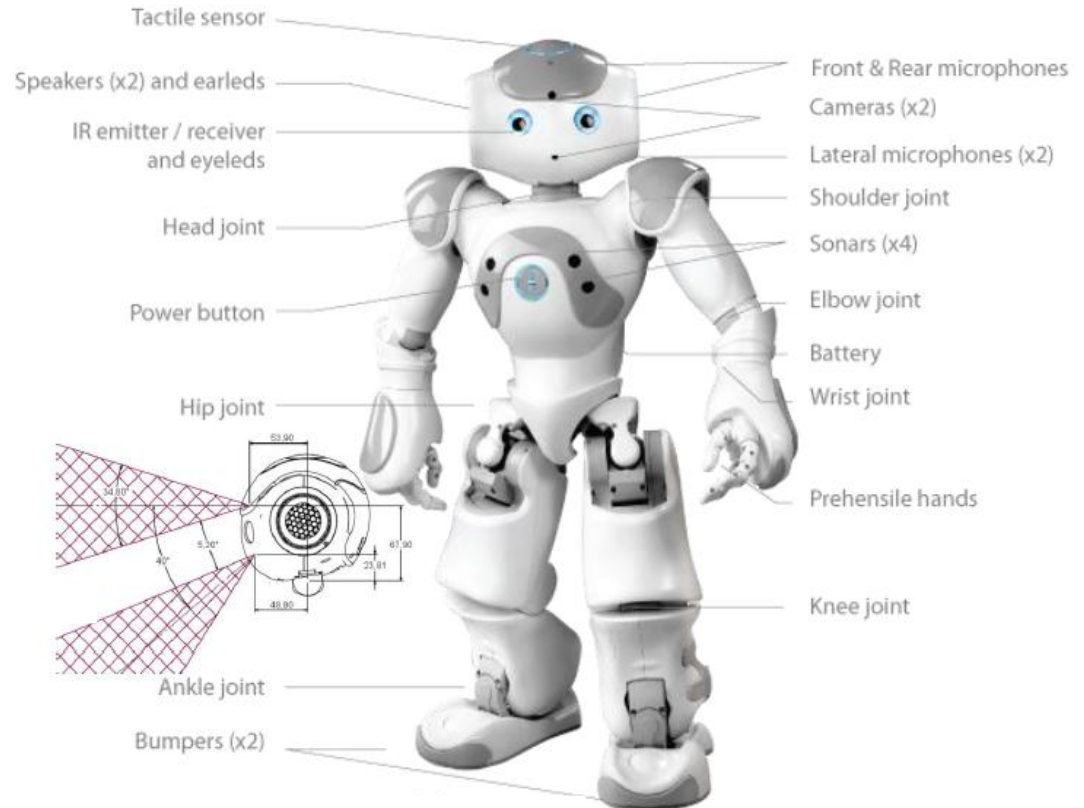
- A) Bibliothek Box
- B) Flow Diagram Panel
- C) Robot View Panel





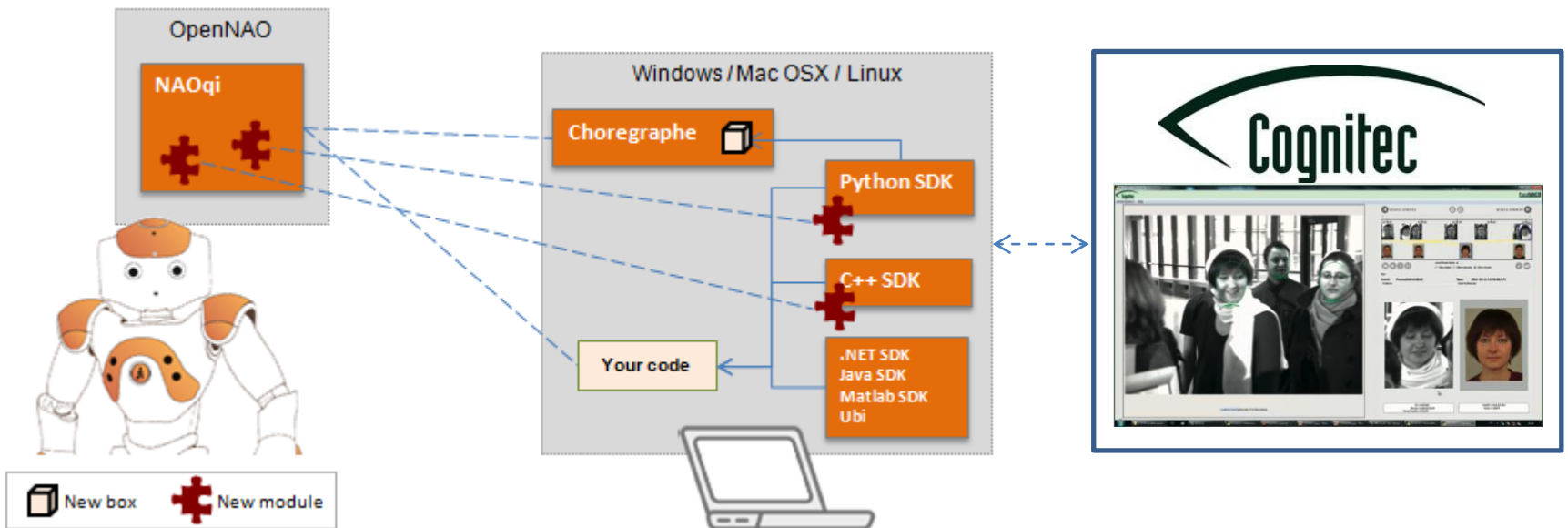
Grundlagen

- Stirn und Mund des Roboters befinden sich Digitalkameras
- 30 Bilder pro Sekunde bei einer Auflösung von 1288 x 968 Pixeln





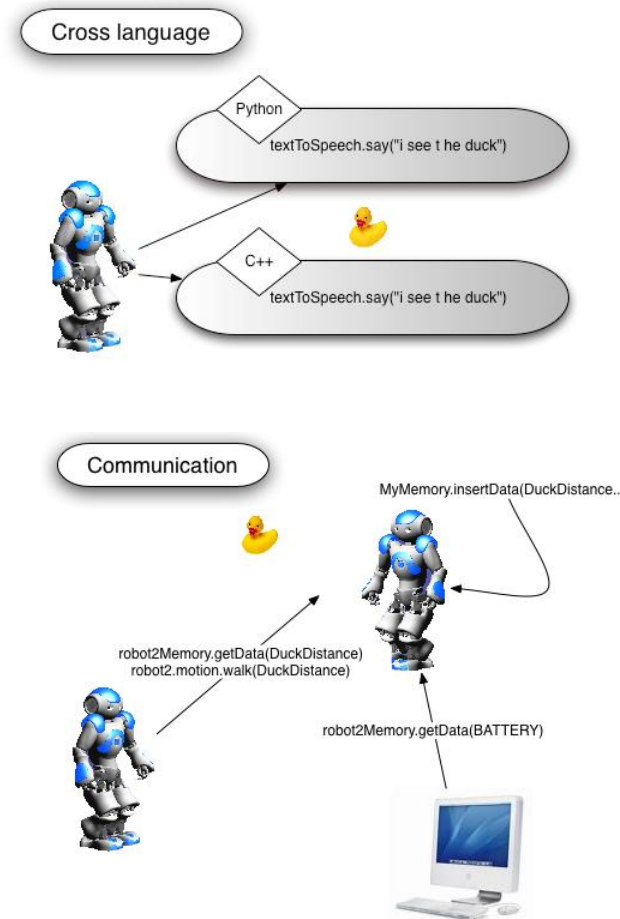
Gesamtmodell





NAOqi Framework

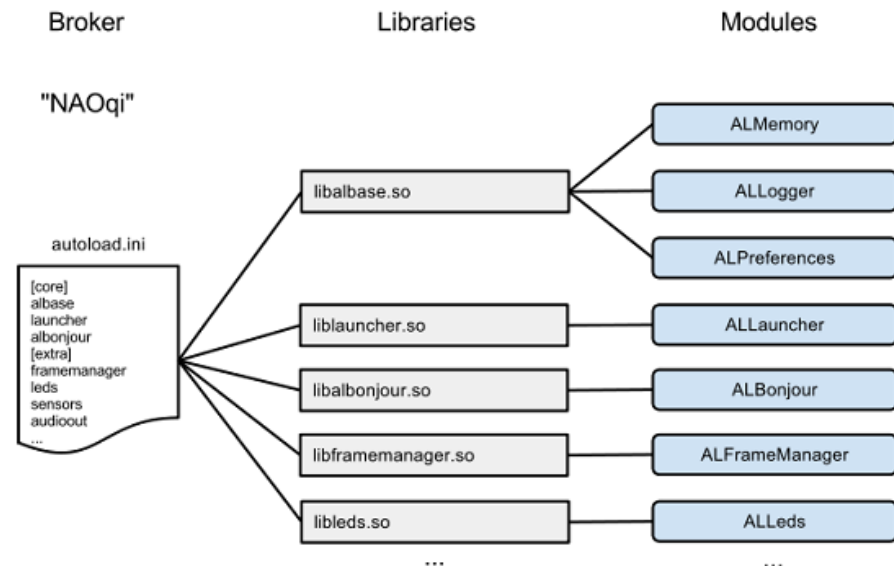
- Grundgerüst und Struktur der Programmierumgebung für Robotic Anwendungen
 - Ressourcenverwaltung
 - Kommunikation
 - Synchronisierung
- Cross-platform
 - Linux, Mac und Windows
- Cross-language
 - API für C++ und Python identisch
- Introspection
 - Self-awareness für laufende Programme
- Cross-platform communication
 - Roboter und Rechner lassen sich untereinander verbinden





NAOqi process

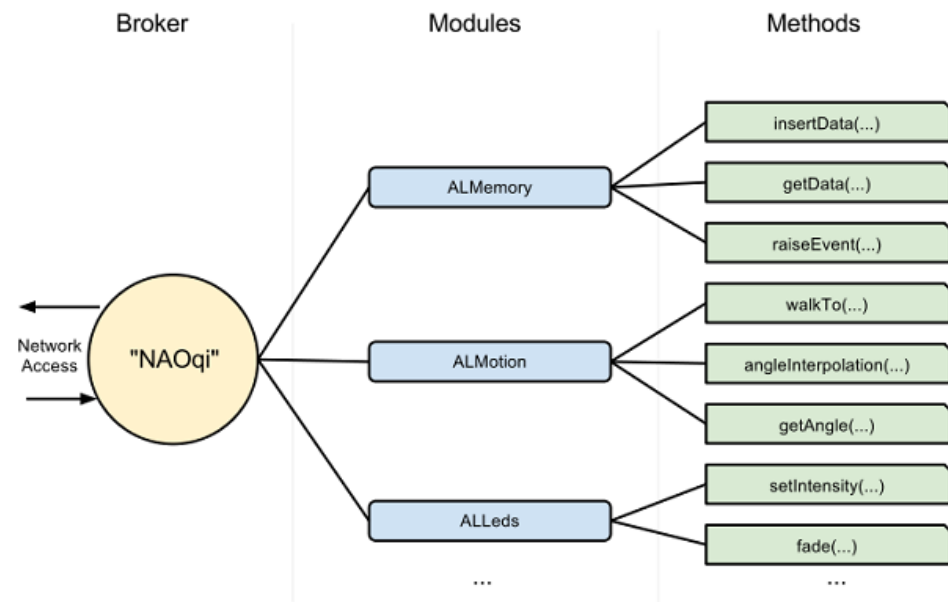
- Broker
 - Lädt Bibliotheken
- Bibliotheken
 - Verknüpfung von Modulen
- Module
 - Enthalten Methoden
 - Lokal oder Remote
- Methoden
 - Werden über den Broker für andere Module zur Verfügung gestellt





NAOqi process

- Kommunikation über Broker
 - Broker als „advertiser“
 - Verkündet seine Module und Methoden
 - Stellt diese für andere Broker zur Verfügung
 - Broker lassen sich über IP miteinander verbinden
- Kommunikation über Proxys
 - Modul als Proxy einbinden
 - Benötigt Name des Moduls und IP Adresse des zugehörigen Brokers





NAO SDK

- NAOqi API
 - Unterstützung von 8 Programmiersprachen
 - Programmierbar auf Windows/Linux/Mac
 - Lauffähige Module für NAO nur mit C++ und Python
 - Nur Remote-Zugriff mit anderen Sprachen
- C++ Framework bei Service Methoden
- Python Framework für Behaviors
- Debuggen und Testen von Code über NAO Simulation möglich
- Viele bereits fertige Module und Methoden
 - Bsp: NAOqi Vision → Video und Bild erfassung





FaceVacs-VideoScan

- FaceVACS-VideoScan erkennt und identifiziert Personen in Echtzeit.

The screenshot displays the FaceVACS-VideoScan software interface. It features a dashboard with several sections:

- Events:** A grid of person thumbnails with labels like "valued customer" and "manager visit".
- Personnel:** A grid of person thumbnails with labels like "currently visiting only".
- Dashboard:** A bar chart titled "Number of Visiting Persons (1:24 PM)" and a pie chart titled "Gender Distribution (1:24 PM)" showing 5 females (50%) and 4 males (44%).
- Person Profile:** A detailed view for "Dana Smith" (Internal ID: SLG797856), identified as a "Valued Customer". It includes a video player showing a visit on 7/3 Today.
- Tablet:** A tablet displaying "Visit Of Person In Gallery" data, including event match status, gallery images, and gallery case details for person ID MEOSYG14.





Further Work

- Bild oder Videostream der NAO-Kamera auf externer Recheneinheit darstellen
- Informationen mit der Cognitec Software FaceVacs verarbeiten, Gesichtserkennung durchführen
- Ausgewertete Bildverarbeitungsdaten an den NAO zurücksenden
- Interaktion mit seinem gegenüber beginnen





Vielen Dank für Ihre Aufmerksamkeit

Fragen ?





Quellen

- /1/ Florian Johannßen Cloud Robotics anhand des humanoiden Roboters NAO
Hochschule für Angewandte Wissenschaften Hamburg
- /2/ Florian Kopelke Einführung und Grundlagen zum Roboter NAO H25
http://fb03.h-bonn-rhein-sieg.de/Personen/Professoren/Irene+Rothe/Nao_Roboter.html
20.05.14
- /3/ Aldebaran Robotics <https://community.aldebaran-robotics.com/doc/1-14/dev/naoqi/index.html#naoqi-overview>
20.05.14





Quellen

/4/ Stefan S. Horstmann

NAO programming in SDK
<http://tectest.reutlingenuniversity.de/projekte/robocup-rt-lions/projekte>

