NetAvatar – „Interaktion mit einem humanoiden Roboter“

Jakob Wakolbinger
Weddigenstr. 9
86179 Augsburg
0043 0821 2297044
jakob@wakolbinger.net

Peter Kirchner
Schäfflerstr. 12
86343 Königsbrunn
0151 50647879
peter.kirchner@youngdesigners.de

ABSTRACT
In diesem Paper wird das Interface „NetAvatar“ vorgestellt, das die Fernsteuerung des humanoiden Roboters Nao ermöglicht. Damit sollen Wizard-of-Oz-Studien durchgeführt werden, die sich mit der Frage beschäftigen, wie Menschen im ungewohnten Kontakt mit humanoiden Robotern reagieren und interagieren.

Der Roboter ist vom Typ Nao von der Firma Aldebaran. Das grafische Interface wurde mithilfe des Python-Frameworks web2Py entwickelt. Als alternative Eingabemöglichkeit verwenden wir einen Xbox 360-Controller.

Erste Erfahrungen mit dem System sammelten wir auf der 40-Jahr-Feier der Universität Augsburg, wo Besucher den Roboter steuern durften und mit ihm Süßigkeiten an andere Besucher verteilten. Die Evaluation der Steuerung war durchweg positiv, die Verzögerung bei der Übertragung der Befehle müsste jedoch minimiert werden. Ein Fragebogen zur Interaktion mit dem Roboter ergab unter Anderem, dass ein optimaler Roboter etwa hüftgroß ist, eine etwas menschlichere Aussprache als der Nao besitzt, ihm vom Aussehen her aber gleichen sollte (nicht mechanischer und nicht menschlicher).

Categories and Subject Descriptors
I.2.9 [**Robotics**]

General Terms
Experimentation, Human Factors

Keywords
Robotics, Human-Computer-Interaction, Mixed-Reality, Web-Interface, GUI, Nao-Roboter, Aldebaran, web2py, Wizard-of-Oz-Studies,

1. MOTIVATION
Ziel ist es, ein grafisches User-Interface zur Fernsteuerung des humanoiden Nao-Roboters zu entwickeln, um damit Wizard-of-Oz-Studien durchführen zu können. Bei diesen Studien macht der Roboter den Eindruck eines autonomen Roboters, während er in Wirklichkeit ferngesteuert wird. Folgende Fragen sollen untersucht werden:

- Wie ist der erste Eindruck und die Reaktion wenn Menschen in die ungewohnte Situation kommen mit einem humanoiden Roboter zu kommunizieren?
- Wie menschlich sollen Roboter werden oder sollen Roboter Roboter bleiben? (z.B. bezüglich der Sprache oder ihrem Aussehen)
- Welche Mechanismen zur Kommunikation sind besonders wichtig?

Das Thema ist sehr relevant, weil es in Zukunft immer mehr Roboter geben wird, die uns Menschen im alltäglichen Leben unterstützen. Die Studien können Ingenieure und Entwickler von humanoiden Robotern nutzen um optimal auf Bedürfnisse und Ängste der Menschen einzugehen.

Das Paper hat folgenden Aufbau: Zu Beginn wird auf verwandte Arbeiten eingegangen. Danach werden die geplanten Use Cases aufgeführt und beschrieben wie sie technisch implementiert wurden. Zuletzt stellen wir unsere Evaluationsergebnisse vor.

2. VERWANDTE ARBEITEN
Inspiriert wurden wir vor allem von Robotern, die sich aus dem Internet steuern lassen. Ein Klassiker ist „Xavier“ von Reid Simmons [4], der am Robotikinstitut der Pittsburgher Carnegie-Mellon-Universität entwickelt wurde. Xavier steht jeden Tag für zwei Stunden zur Verfügung, um die Wünsche zu erfüllen, die den Tag über aus dem Internet eingetroffen sind. Damit der Roboter nicht den Laborbetrieb durcheinanderbringt, darf er nur bestimmte Dinge auf Fernkommando tun: an die Türen der Mitarbeiter klopfen, Bilder aufnehmen und sogenannte "knock-knock jokes" erzählen. Da Xaviers Zuverlässigkeit noch knapp unter hundert Prozent liegt, folgt ihm ein Techniker auf Schritt und Tritt.

Einen wissenschaftlicheren Ansatz bietet das Paper „Multimodal Metric Study for Human-Robot Collaboration“ von Green et al. [6]. Das Ziel der Autoren ist es, ein System zu entwickeln, in dem menschliche Mitglieder eines Teams auf eine natürlichen Art und Weise mit Robotern kommunzieren. In einer Studie wird versucht herauszufinden, welche Elemente der menschlichen Kommunikation besonders häufig genutzt werden. Testpersonen sollten einen Roboter durch ein Labyrinth hindurchführen und dabei Sprache und Gesten verwenden. Die Testpersonen nahmen

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Conference'04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

diese Aufgabenstellung gut an und nutzen unter Anderem auch ausführliche Armbewegungen.

Auch Raph et al. [7] suchen in ihrer Studie nach den bevorzugten Kommunikationsmechanismen bei der Interaktion mit Robotern, konzentrieren sich aber ausschließlich auf die Sprache. Nutzer sollten einem Roboter, der an einem Tisch fixiert war, Anweisungen geben, um fünf verschiedene Objekte aufzuheben. Dabei waren sie angewiesen, so ausführlich wie möglich in ihren Beschreibungen zu sein. Die Anweisungen wurde dann im Hintergrund von einem menschlichen Angestellten in Roboterbewegungen übersetzt. Die Testpersonen waren der Meinung, dass die natürliche menschliche Sprache gut geeignet ist um mit dem Roboter zu kommunizieren. Sie tendierten jedoch oft zu Kurzanweisungen in einer mechanischen Art und Weise.

Im Gegensatz zu den angesprochenen Arbeiten konzentrieren wir uns explizit auf die Interaktion mit einem humanoiden Roboter. Dieser hat eine ganz andere Wirkung auf die Menschen als ein grafisches Interface oder ein Roboter mit stark mechanischer Anmutung. Trotz ähnlicher Fragestellungen sind andere Ergebnisse zu erwarten.

Außerdem ist unser System stark skalierbar. Schon jetzt sind viele wichtige Bewegungen direkt ansteuerbar, weitere Verhalten des Roboters werden dynamisch geladen. Dadurch eröffnen sich viele Möglichkeiten für verschiedenste Studien und zum Untersuchen komplexer Fragestellungen.

3. ANWENDUNGSSZENARIO & USE CASES

3.1 Szenario

Nutzer können über ein graphisches User Interface einen humanoiden Roboter steuern sowie mit der Außenwelt kommunizieren. Der Nutzer, der den Roboter steuert, wird im Folgenden “Pilot” genannt.

3.2 Definition der Use Cases

Use-Case 1: Der Pilot läuft mit dem Roboter

- Ziel: Der vom Piloten gesteuerte Roboter soll sich überall hin bewegen können.
- Akteure: Pilot; Mensch, der mit dem Roboter interagiert
- Bedienablauf: Der Pilot startet die Aktion mit einem Maus-Klick auf den Pfeil-Button. Der Roboter läuft solange in die entsprechende Richtung wie der Button gedrückt wird.
- Kontextbedingungen: Der Roboter hat eine Verbindung zum Server aufgebaut, die Batterie des Roboters ist aufgeladen.

Use-Case 2: Der Pilot sieht die Kamera-Sicht des Roboters und hört die aufgenommenen Audio-Signale

- Ziel: Der Pilot soll Feedback darüber erhalten, ob die Operationen geglückt sind und wie die Reaktionen in der Umwelt sind.
- Akteure: Pilot; Mensch, der mit dem Roboter interagiert
- Bedienablauf: GUI stellt automatisch Kamerabild zur Verfügung gibt und Audiosignal aus.
- Kontextbedingungen: Der Roboter hat eine Verbindung zum Server aufgebaut, die Batterie des Roboters ist aufgeladen.

Use-Case 3: Der Pilot dreht den Kopf des Roboters

- Ziel: Der Pilot soll überall hin blicken können und der Roboter soll durch direkten Blickkontakt noch realistischer wirken.
- Akteure: Pilot; Mensch, der mit dem Roboter interagiert
- Bedienablauf: Der Pilot betätigt die Buttons zum Drehen des Kopfes mit einem Maus-Klick. Der Kopf dreht sich in einzelnen Schritten, so dass die Maus immer nur kurz geklickt werden sollte.
- Kontextbedingungen: Der Roboter hat eine Verbindung zum Server aufgebaut, die Batterie des Roboters ist aufgeladen.

Use-Case 4: Der Pilot kommuniziert über Text-to-Speech

- Ziel: Der Roboter soll mit den Menschen interagieren und vor allem kommunizieren können.
- Akteure: Pilot; Mensch, der mit dem Roboter interagiert
- Bedienablauf: Der Pilot gibt einen beliebigen Text in ein Textfeld ein und klickt auf den Button zum Abschicken.
- Kontextbedingungen: Der Roboter hat eine Verbindung zum Server aufgebaut, die Batterie des Roboters ist aufgeladen.

Use-Case 5: Der Pilot startet bzw. stoppt vorprogrammiertes Verhalten des Roboters

- Ziel: Der Roboter soll zu den grundlegenden Verhalten zusätzliche komplexere Verhalten zeigen können, damit die Interaktion nicht eindimensional bleibt.
- Akteure: Pilot; Mensch, der mit dem Roboter interagiert
- Bedienablauf: Der Pilot wählt das entsprechende Verhalten aus einer Drop-Down-Liste aus und startet es mit einem Button. Das Verhalten wird vorzeitig abgebrochen wenn der Stop-Button geklickt wird oder stoppt automatisch sobald es zu Ende ist.
- Kontextbedingungen: Der Roboter hat eine Verbindung zum Server aufgebaut, die Batterie des Roboters ist aufgeladen.

4. PROTOTYP & IMPLEMENTIERUNG

4.1 Technologie & Softwareumgebung

Der Roboter ist vom Modell Nao und wird hergestellt von der französischen Firma Aldebaran [1]. Eigentlich ist er ein autonomer Roboter, dessen Sensoren und Eingabegeräte so programmiert werden können, dass er von alleine reagiert. Doch augrund der komfortablen API sind dem Entwickler keine

Grenzen gesetzt, auch nicht-autonome Steuerungsmechanismen sind möglich. Die API ist in Python und C++ verfügbar.

Das Interface war von Anfang als Web-Applikation geplant, das über den Browser aufrufbar ist. Um uns auf das Wesentliche konzentrieren zu können, arbeiteten wir mit dem Python-Framework Web2Py [2]. Es wurde von Massimo de Piero entwickelt und bietet eine saubere Softwarearchitektur auf Grundlage des Model-View-Controller-Modells.

Als alternative Eingabemöglichkeit verwenden wir einen Xbox360-Controller [3]. Die Bedienung wirkt damit natürlicher; in Verbindung mit dem grafischen Interface bietet der Controller die optimale Steuerung. Zur Ansteuerung nutzten wir die Library pygame [4].

Da die Kamera des Roboters nur etwa 3 Frames pro Sekunde bei externem Zugriff leisten konnte, brachten wir eine externe Kamera der Firma Technaxx auf dem Roboter an. Zum Bereitstellen der Webcam-Bilder als Webstream verwendeten wir WebcamXP.

4.2 Aufbau & Umfang des Prototypen

4.2.1 Roboter

Auf dem Roboter läuft Linux und darauf das Betriebssystem des Roboters "NaoQi". Außerdem befinden sich die so genannten lokalen Module auf dem Roboter; beispielweise ein Modul für Bewegungen oder ein Modul für TextToSpeech.

4.2.2 Schnittstelle

Schnittstelle zwischen Roboter und externen Programmen wie unserem Server ist WLAN. Voraussetzung zur Kommunikation ist also, dass Roboter und Server sich im selben Netz befinden.

4.2.3 Server

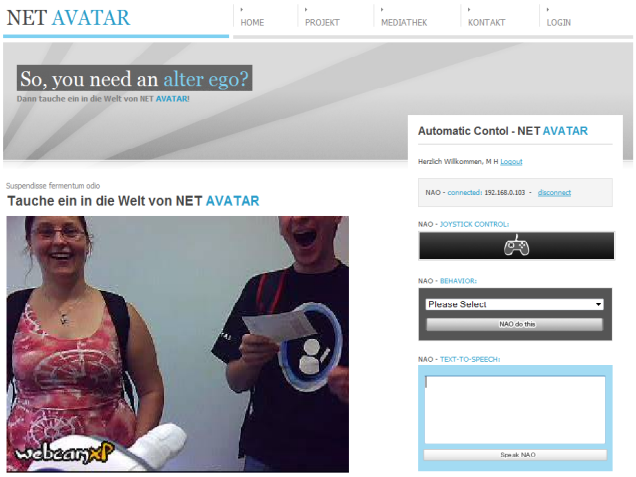
Der Server stellt das graphische User-Interface zur Verfügung. Er verbindet sich über IP mit dem Roboter. Dann führt er so genannte Remote Moduls aus. Das sind Module, die nicht auf dem Roboter, sondern extern laufen. Sie können lokale Module ansprechen und den Roboter so steuern

Welche Use Cases wurden umgesetzt?

Fast alle geplanten Use Cases sind implementiert worden. Lediglich die Aufnahme von Audiosignalen vom Roboter und deren Ausgabe im Interface ist zu diesem Zeitpunkt noch nicht realisiert.

Im Laufe der Zeit wurden einige Verbesserungen am Interface durchgeführt,

- Einige Use Case - Bedienabläufe wurden optimiert, außerdem wird nun der Xbox-360-Controller als primäre Eingabemöglichkeit für bestimmte Use Cases verwendet.
- Ein Login bietet zusätzliche Sicherheit.
- Über die Eingabe einer IP-Adresse kann man verschiedene Roboter schnell austauschen und einsetzen.
- "Views" bieten die Möglichkeit verschiedene Sichten auf die Steuerung zu definieren.



4.3 Beispiel: Laufen

Dies ist ein sehr vereinfachtes Beispiel und stellt nur einen Auszug aus dem Zusammenspiel der Komponenten des Interfaces dar. Es zeigt aber wie einfach ein Befehl an den Roboter übermittelt werden kann.

Über einen so genannten Proxy kann man auf die lokalen Module des Roboters zugreifen; hier das Modul für Bewegungen. Durch die Angabe von IP-Adresse und Port kann sich der Proxy mit dem Roboter verbinden.

motionProxy = ALProxy („ALMotion"„192.168.0.103", 9559)

Die eigentliche Steuerung ist in unserem System in Callback-Methoden gekapselt, welche aufgerufen werden sobald ein Button geklickt wird. Anschließend startet ein einfacher Methodenaufruf auf dem Proxy-Objekt die Aktion des Roboters; hier wird die Geschwindigkeit des Roboters gesetzt.

```
def setSpeed(x, y, theta, frequency):
        motionProxy.setWalkTargetVelocity
        (x, y, theta, frequency)
```

5. EVALUATION

5.1 Ziele der Evaluation und Methodik

Den ersten Einsatz findet das System auf der 40-Jahr-Feier der Universität Augsburg. Dort dürfen Besucher abwechselnd den Roboter steuern und Süßigkeiten an andere Besucher verteilen. Es geht hierbei nicht um eine exakte wissenschaftliche Studie. Vielmehr wollen wir im Rahmen dieses Events erste Erfahrungen mit dem System sammeln.

Ablauf und Aufbau:

Insgesamt sind für die Durchführung vier Slots à 45 Minuten eingeplant. In dieser Zeit wechseln sich interessierte Besucher als Piloten ab, wobei ein Pilot etwa 10 bis 15 Minuten „am Steuer ist".

Die Piloten sitzen in einem Raum an einem Tisch. Der Roboter wird im Gang daneben oder im selben Raum platziert, so dass der Pilot ihn zusätzlich zur Ego-Kamerasicht auch direkt im Blickfeld hat. Über bestimmte Buttons auf dem Controller kann der Pilot mit dem Roboter Süßigkeiten aufnehmen, um sie anschließend an

Besucher weiterzugeben. Außerdem kann der Pilot über Text-to-Speech mit ihnen kommunizieren.



Die Durchführung wird mit einer Kamera dokumentiert.

Abgesehen davon, dass die Aktion zur Unterhaltung dienen und zur Auseinandersetzung mit dem Roboter führen soll, interessieren uns bei diesem Event zwei Aspekte:

Evaluation der Roboter-Steuerung:

Zunächst interessiert uns wie gut das System im laufenden Betrieb funktioniert: Ist das System stabil und funktioniert die Hardware? Gibt es bestimmte Anwendungsfälle, die Probleme bereiten? Kann man den Roboter so flüssig steuern, dass er den Eindruck eines autonomen Roboters macht? usw.

Studie zur Interaktion mit einem humanoiden Roboter:

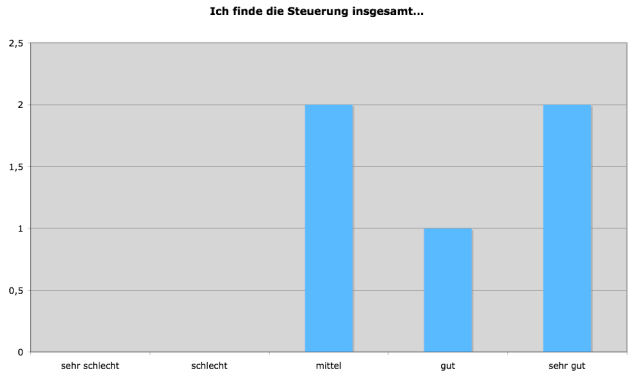
Außerdem wollen wir erste Eindrücke zu den Fragestellungen der Wizard-of-Oz-Studien bekommen: Wie ist die spontane Reaktion beim Erstkontakt mit dem Roboter? Welche Mechanismen zur Kommunikation sind besonders wichtig? usw.

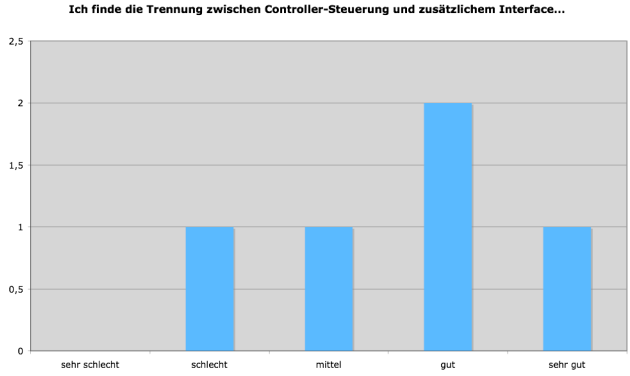
Beide Untersuchungen wurden durch einen Fragebogen gestützt, siehe 5.2.

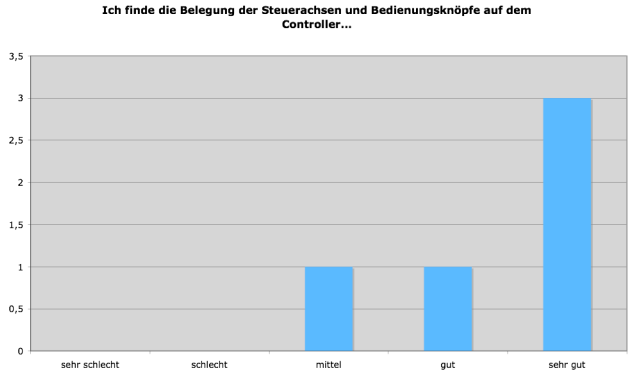
5.2 Durchführung der Evaluation

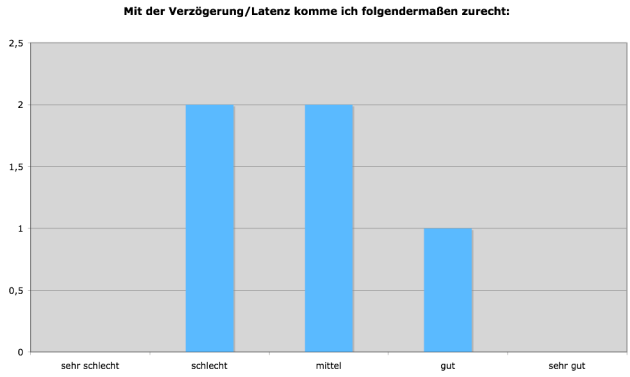
Fragebogen 1: Evaluation der Roboter-Steuerung

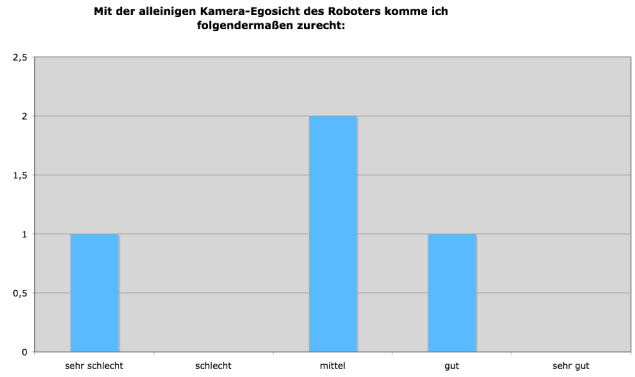
Hierbei wurden 5 Personen befragt, die die Roboter-Steuerung ausprobiert haben. Darunter waren zwei Schüler, zwei Stundenten und ein Betriebsingenieur – ihr durchschnittliches Alter beträgt 24 Jahre.

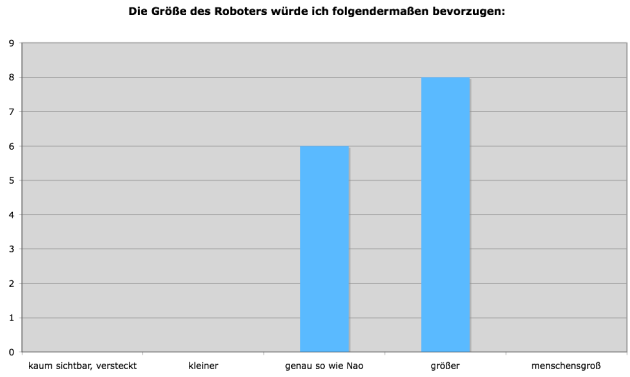






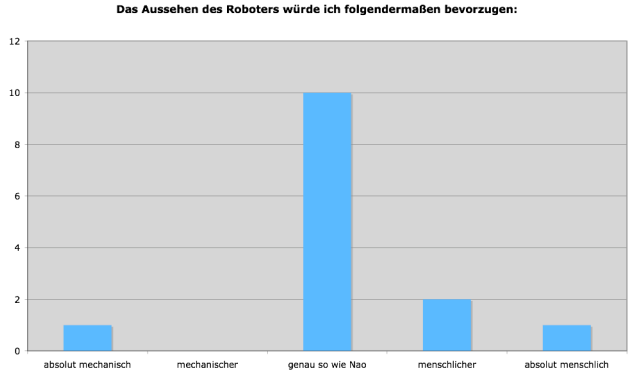


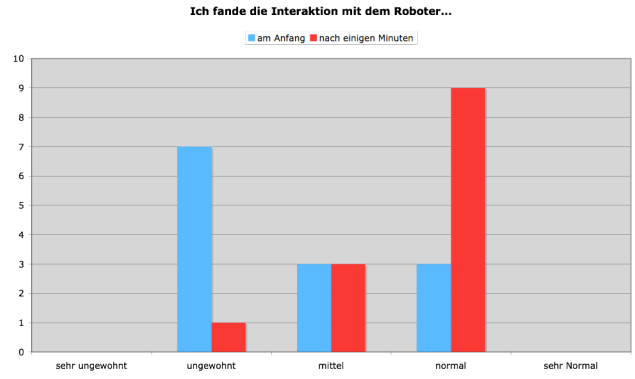


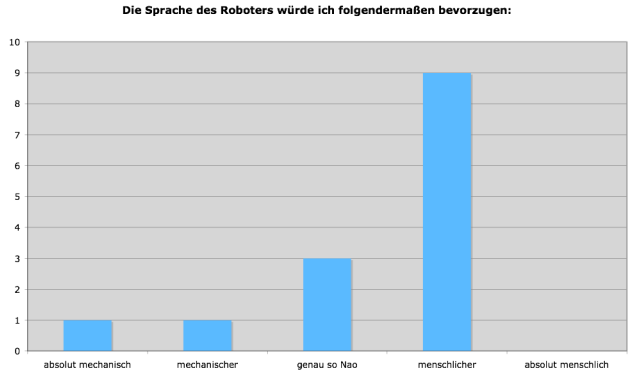


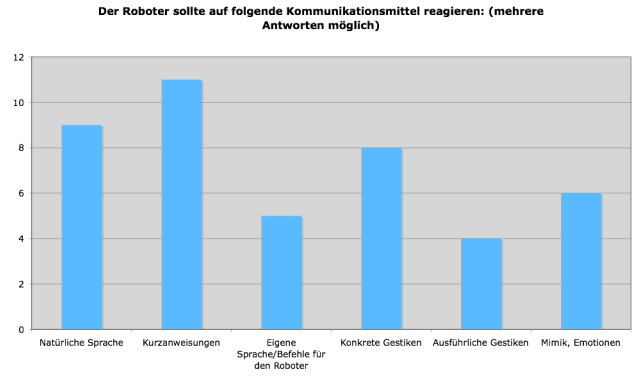
Fragebogen 2: Studie zur Interaktion mit einem humanoiden Roboter(Nao).

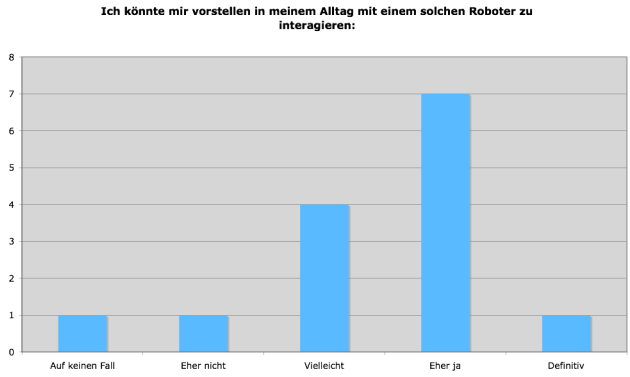
An dieser Studie haben insgesamt 14 Personen teilgenommen, welche mit dem Roboter interagierten. Das Teilnehmerfeld war bunt gemischt: Von Abiturienten über Studenten, Ingenieuren, Angestellten bis hin zu Lehren waren verschiedenste Nutzergruppen vertreten. Ihr durchschnittliches Alter beträgt 32 Jahre.











5.3 Interpretation der Evaluationsergebnisse

Evaluation der Roboter-Steuerung:

Insgesamt sind die Testergebnisse erfolgreich zu bewerten. Nach einer Eingewöhnungsphase von etwa 10 Minuten bekamen die Testpersonen aber ein Gefühl für den Roboter und konnten ihn flüssig steuern.

Etwas zu schaffen machte den Testpersonen die Verzögerung bei der Übermittlung der Befehle. Es ist zu hoffen, dass die nächsten Versionen des Nao-Roboters eine bessere WLAN-Verbindung und stärkere Prozessoren mit sich bringen.

Außerdem ist aufgefallen, dass die meisten Testpersonen nicht die Egosicht des Roboters verfolgten, sondern den Roboter von außen betrachteten. Die Ursache liegt darin, dass die Kamera zu schlechte Bilder lieferte. Zudem wären externe Kamerasichten auf den Roboter sinvoll; der Pilot hätte so einen besseren Überblick über die Szene.

Studie zur Interaktion mit einem humanoiden Roboter:

Die Meinungserhebung zeigt, dass viele Menschen die Interaktion mit dem Roboter am Anfang noch ungewohnt finden, sich aber sehr schnell daran gewöhnen.

Bei den Kommunikationsmitteln gibt es sehr verschiedene Meinungen. Eine Tendenz geht zur natürlichen Sprache, zu Kurzanweisungen und zu konkreten Gestiken. Gegenüber hochintelligenten Mechanismen wie dem Erkennen von Emotionen ist das Misstrauen noch zu hoch.

Klarere Meinungen gibt es zum Aussehen und der Sprache des Roboters. Etwa die Hälfte wünscht sich den Roboter größer, aber nicht menschensgroß. Der Nao-Roboter ist faktisch so klein, dass man ihn beinahe übersieht oder ihn überläuft. Roboter, die so groß wie Menschen sind würden uns in unserem Alltag aber nur verängstigen, ähnlich wie das bei großen Tieren der Fall ist.

Die humanoide Anmutung des Roboters wird gut angenommen. Eine noch menschlicheres Aussehen wird kaum gewünscht. Der potentielle Grund: Rein menschliche Züge dürften als Mantel eines mechanischen Systems „gruselig“ wirken und lösen ein unbehagliches Gefühl aus.

Die Sprache des Roboters ist den befragten Personen noch zu mechanisch, sie wünschen sich eine menschlichere Aussprache. Das dürfte damit zusammenhängen, dass die Text-to-Speech-Ausgabe nicht immer ganz leicht zu verstehen ist, sie wirkt manchmal etwa nuschelig.

6. ZUSAMMENFASSUNG

6.1 Erzielte Ergebnisse

Eine erste Version des Interfaces wurde erfolgreich fertig gestellt. Zusätzlich zu den geplanten Use Cases haben wir mit dem X-Box-360-Controller eine alternative Eingabemöglichkeit geschaffen.

Auf der 40-Jahr-Feier der Universität Augsburg sammelten wir bereits erste Erfahrungen mit dem System. Dort durften Besucher den Roboter steuern und mit ihm Süßigkeiten an andere Besucher verteilen. Die Evaluation der Steuerung war durchweg positiv, die Verzögerung bei der Übertragung der Befehle müsste jedoch minimiert werden. Fragebögen zur Interaktion mit dem Roboter ergaben unter Anderem, dass ein optimaler Roboter etwa hüftgroß ist, etwas menschlicher als der Nao redet, ihm vom Aussehen her aber gleichen sollte (nicht mechanischer und nicht menschlicher).

6.2 Erweiterungsmöglichkeiten / Ausblick

Die Tests haben gezeigt, dass das Interface noch etwas optimiert werden sollte. Besonders ein besseres Kamerabild ist von Nöten. Der fehlende Sound-Input vom Mikrophon des Roboters und die Ausgabe im Interface ist ebenfalls noch zu realisieren.

Anschließend können methodisch geplante Wizard-of-Oz-Studien oder andere Events folgen. Auch über einen Film mit den Nao-Robotern wird nachgedacht.

7. DANKSAGUNG

Vielen Dank an Herrn Rist, der uns den Kontakt zur Universität Augsburg hergestellt hat, sowie an Frau Andre und Herrn Häring, die uns die Arbeit mit den Nao-Robotern ermöglicht haben.

8. REFERENZEN

[1] Aldebaran Robotics, the creators of Nao http://www.aldebaran-robotics.com/en

[2] web2py Enterprise Web Framework, Massimo Di Pierro www.web2py.com

[3] Xbox 360 Controller for Windows http://www.microsoft.com/hardware/gaming/ProductDetails.aspx?pid=091&active_tab=overview

[4] Pygame – python game development, www.pygame.org/

[5] Reid Simmons, „Xavier“ http://www.ri.cmu.edu/research_project_detail.html?project_id=91&menu_id=261

[6] Scott A. Green, Scott M. Richardson, Randy J. Stiles, Mark Billinghurst, J. Geoffrey Chase , “Multimodal Metric Study for Human-Robot Collaboration”

[7] M. Ralph and M. Moussa, "Human-Robot, Interaction for Robotic Grasping: A Pilot Study", 2005, http://www.uoguelph.ca/~mralph/IROS05_general_paper.pdf