



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Asmatullah Noor

**Entwicklung eines autonomen Agenten zur
Mensch-Maschine-Interaktion auf Basis der Nao Plattform**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Asmatullah Noor

**Entwicklung eines autonomen Agenten zur
Mensch-Maschine-Interaktion auf Basis der Nao Plattform**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Olaf Zukunft
Zweitgutachter: Prof. Dr. Stephan Pareigis

Eingereicht am: 7. Oktober 2016

Asmatullah Noor

Thema der Arbeit

Entwicklung eines autonomen Agenten zur Mensch-Maschine-Interaktion auf Basis der Nao Plattform

Stichworte

NAO, Humanoider Roboter, Mensch-Maschine-Interaktion, Choregraphie

Kurzzusammenfassung

Im Rahmen dieser Bachelorarbeit wurde ein Softwaresystem zur Mensch-Maschine-Interaktion auf Basis der NAO Plattform entwickelt. Das Softwaresystem beinhaltet vier verschiedene Szenarien, dass zur Außendarstellung der Hochschule für Angewandte Wissenschaften dient. Eines der vier Szenarien sollte als interaktives Szenario umgesetzt werden, dass dem Nutzer um eine Erweiterung des Szenarios ermöglicht. Hierzu wurden grundlegende Technologien, Hardwarekomponente sowie Schlüsselkonzepte zur Ansteuerung des NAO erörtert. Um die Szenarien als Erfolgreich anzusehen, ist im Abschluss eine Systemvalidierung, auf die gestellten Anforderungen hin getestet wurden.

Asmatullah Noor

Title of the paper

Developing an autonomous agent to Human-machine interaction on the basis of NAO platform

Keywords

NAO, Humanoid robot, Human-Machine interaction, Choregraphie

Abstract

This thesis discusses an softwaresystem that interact between human and machine based on the nao plattform. The softwaresystem includes four different scenario, they will be used for prresentation purposes of the university of applied sciene. One scenario should have an interactive element. The first chapter explains the basics of the nao technologies, which is important to develop NAO applications. All scenario need to pass a systemvalidation, to the given requirements.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung dieser Arbeit	2
2. Grundlagen	3
2.1. Humanoider Roboter	3
2.1.1. NAO	3
2.2. Technische Daten des NAO	4
2.2.1. Computersystem	4
2.2.2. Kinematik und Motorik	6
2.2.3. Sensorik	7
2.3. Entwicklungssoftware des NAO	8
2.3.1. NAOqi	9
2.3.2. Programmierung	12
3. Analyse	14
3.1. Systemvision	14
3.2. Anwendungsfälle	16
3.3. Anforderungen	17
3.4. Anforderungsspezifikation - Szenario 1	18
3.4.1. Tanz	19
3.4.2. Ablauf	19
3.4.3. Aktivitätsdiagramm	19
3.4.4. Anforderungen	22
3.5. Anforderungsspezifikation - Szenario 2	22
3.5.1. Schere, Stein, Papier	23
3.5.2. Ablauf	25
3.5.3. Aktivitätsdiagramm	25
3.5.4. Anforderungen	27
3.6. Anforderungsspezifikation – Szenario 3	28
3.6.1. Quizzes-Show	28
3.6.2. Ablauf	28
3.6.3. Aktivitätsdiagramm	28
3.6.4. Anforderungen	30
3.7. Anforderungsspezifikation – Szenario 4	31
3.7.1. Tanzszenario erweitern	31

3.7.2.	Ablauf	32
3.7.3.	Aktivitätsdiagramm	32
3.7.4.	Anforderungen	33
4.	Konzept	35
4.1.	Evaluation zur Modulprogrammierung	35
4.1.1.	Programmierung anhand Choregraphie	36
4.1.2.	Programmierung anhand SDK	37
4.1.3.	Auswertung der Evaluation	40
4.2.	Systemarchitektur	41
4.3.	Technischer Ansatz	42
4.4.	Konzeption - Szenario 1	43
4.4.1.	Bewegung	44
4.4.2.	Technischer Ansatz	45
4.4.3.	Interaktionsdiagramm	45
4.5.	Konzeption - Szenario 2	47
4.5.1.	Symbol	47
4.5.2.	Strategie	47
4.5.3.	Technischer Ansatz	48
4.5.4.	Fachliches Datenmodell	49
4.5.5.	Architektur	50
4.5.6.	Interaktionsdiagramm	52
4.6.	Konzeption - Szenario 3	54
4.6.1.	Fachliches Datenmodell	54
4.6.2.	Architektur	55
4.6.3.	Interaktionsdiagramm	56
4.7.	Konzeption - Szenario 4	58
4.7.1.	Architektur	58
4.7.2.	Interaktionsdiagramm	59
5.	Realisierung	61
5.1.	Umsetzung - Szenario 1	61
5.2.	Umsetzung - Szenario 2	65
5.3.	Umsetzung - Szenario 3	70
5.4.	Umsetzung - Szenario 4	74
6.	Systemvalidierung	76
6.1.	Validierung - Szenario 1	76
6.2.	Validierung - Szenario 2	78
6.3.	Validierung - Szenario 3	80
6.4.	Validierung - Szenario 4	82
6.5.	Anforderungsabgleich	83
6.6.	Hardwarebeschränkung des NAO	85

6.7. Umfeld	86
7. Zusammenfassung und Ausblick	88
Anhang	i
A. Aktivitätsdiagramm	i
B. Anwendungsfälle	iv
C. Evaluation	vi
D. Interaktionsdiagramm	vii
E. Hauptschritte des Szenario 1	viii
Abbildungsverzeichnis	x
Tabellenverzeichnis	xiii
Listings	xv
Abkürzungsverzeichnis	xvi
Glossar	xvii

1. Einleitung

1.1. Motivation

Roboter sind in der heutigen Zeit Maschinen, die entwickelt werden, um den Menschen bei seinen alltäglichen Tätigkeiten zu unterstützen und ihn (dadurch) zu entlasten: Sie navigieren Personen zum Ziel, pflegen ältere Menschen, helfen Kindern beim Lernen, spielen Fußball und entschärfen sogar Bomben. Auch in der Industrie sind Roboter kein fremder Anblick mehr: Sie werden für eine konkrete Aufgabe konzipiert und hergestellt, wie die Montage oder Bearbeitung von Werkstücken. Unermüdlich führen Industrieroboter monotone Tätigkeiten durch, mit konstanter Präzision und ohne Leistungsabfall.

Roboter, die dem Menschen nachempfunden sind, werden als humanoide Roboter bezeichnet. Die Anatomie eines solchen Roboters ist an die des Menschen angepasst: Neben einem Kopf besitzt er zwei Arme und zwei Beine, somit ist die Körperstruktur ideal, um die gleichen Bewegungen wie ein Mensch auszuführen. Anhand der Beine kann sich der Roboter im Raum fortbewegen. Die Arme helfen ihm beim Greifen von Gegenständen. Durch Integration weiterer Hardware wie Kameras wird das maschinelle Sehen erschaffen und somit ist ihm auch eine visuelle Wahrnehmung seiner Umgebung möglich. Der humanoide Roboter wurde nicht entwickelt, um nur auf eine Tätigkeit beschränkt zu sein. Ganz im Gegenteil; er bewegt sich in der Umgebung des Menschen und somit hilft er bei der Ausübung von dessen Tätigkeiten. Sobald humanoide Roboter in das Umfeld von Menschen integriert werden, spielt die Mensch-Maschine-Interaktion eine zentrale Rolle. Die für verwendete Technik zählt nicht als einzige wichtige Komponente für einen erfolgreichen Einsatz. Daneben spielen die Interaktionsmöglichkeiten und die Verhaltensweise, die der humanoide Roboter mit sich bringt, eine große Rolle. Im Rahmen dieser Bachelorarbeit wird eine Mensch-Maschine-Interaktion zur Außendarstellung des Fachbereichs Informatik an der HAW Hamburg entwickelt.

1.2. Zielsetzung dieser Arbeit

Das Ziel der vorliegenden Bachelorarbeit ist es, vier verschiedene Mensch-Maschine-Interaktionen zu entwickeln, die letztlich der Außendarstellung der Hochschule für Angewandte Wissenschaften Hamburg dienen sollen. Dem Nutzer soll eine Schnittstelle, mit der nötigen Hilfestellung, zur Verfügung stehen, um jeweils interaktiv ein spezielles Szenario zu nutzen und zu erweitern. Die vier Szenarien fokussieren auf Mensch-Mensch-Interaktionen, die sich im Zusammenspiel aufbauen, was sich auf eine Mensch-Maschine-Interaktion übertragen lässt.

Stellvertretend für die Rolle der Maschine soll ein humanoider Roboter namens „NAO“ von der Firma Aldebaran Robotics deren Platz einnehmen und mit dem interagieren. Die vier Szenarien werden im folgenden Abschnitt **3.1** näher vorgestellt.

2. Grundlagen

Nachfolgend erfolgt zuerst eine Einführung in die NAO-Roboter. Anschließend werden die technischen Komponenten des NAO-H25 im Einzelnen vorgestellt, welche in diese Thesis einfließen. Abschließend wird die Entwicklungssoftware NAOqi mit dem Schwerpunkt des Schlüsselkonzeptes vorgestellt, um das Ansprechen der Modul-Methoden zu erörtern.

2.1. Humanoider Roboter

2.1.1. NAO

Die Firma Aldebaran Robotics beschäftigt sich seit ihrer Gründung im Jahr 2005 in Paris mit der Entwicklung humanoider Roboter. Schon im darauffolgenden Jahr präsentierte das zwölfköpfige Team den ersten NAO-Prototyp aus der NAO-Roboter-Familie. Die NAO-Roboter-Familie beinhaltet verschiedene Ausführungsmodelle (T2, T14, H21 und H25), die sich in den Freiheitsgrad (DoF) unterscheiden, was bereits anhand des Akronyms im Namen jeweils gut identifiziert werden kann.

NAO Torso

Die T-Modelle sind Ausführungen des NAO für einfach strukturierte Umgebungen, da sie aufgrund der geringen DoF nur ortsgebunden einsetzbar sind. Die Grundkörperteile sind fest definiert, bestehend aus Torso und Kopf; diese minimale Ausführung stellt der NAO T2 dar (siehe Abbildung 2.1). Eine Erweiterung des DoF durch das Hinzufügen von zwei Armen repräsentiert der NAO T14 (siehe Abbildung 2.1). Die T-Modelle sind ideal für den Einstieg in die Entwicklung des NAO geeignet, das Akronym „T“ steht hierbei für Torso [1].

NAO Humanoid

Die dem Menschen anatomisch ähnlichste Gestalt gelingt Aldebaran Robotics mit den H-Modellen des NAO, das Akronym „H“ steht für humanoid. Die H-Modelle sind eine Erweiterung der T-Modelle durch die Ergänzung um zwei Beine. Aktuell ist bereits die fünfte Version des

Modells NAO H25 (siehe Abbildung 2.1) entwickelt worden. Die stetige Weiterentwicklung neuerer Generationen konzentriert sich auf die Verbesserung der Sensoren, die Optimierung der Interaktionsfunktionen sowie einen robusteren Körperbau und ein leistungsfähigeres Betriebssystem [2]. Diese Entwicklungen machen den NAO nicht nur für die Forschung und Bildung attraktiv, er ist seit 2008 der Diego Maradona beim Fußballspielen in der RoboCup-Liga, da er interaktiv, autonom und vollständig programmierbar ist [3].

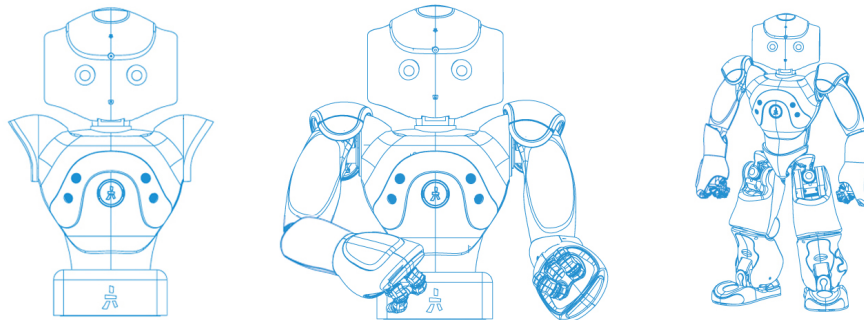


Abbildung 2.1.: NAO-Roboter Familie: Die Torso-Modelle T2 (Links) und T14 (Mittig), die Humanoide-Modelle H21 und H25 (Rechts) [4].

2.2. Technische Daten des NAO

Die interaktive Präsentation dieser Bachelorarbeit wird anhand des Modells H25 veranschaulicht, dieser ist mit der aktuellen Hardware ausgestattet. Der NAO H25 hat in der aufrechten Position vom Scheitel bis zur Sohle eine Körperhöhe von 57,4 cm bei einem Gesamtgewicht von 5,2 kg. Durch seine 25 DoF ist er dazu in der Lage, eine Vielzahl von Bewegungen auszuführen.

In diesem Abschnitt werden nur die wichtigsten Komponenten des NAO beleuchtet, die in die vorliegende Thesis einfließen. Eine detailliertere Auflistung der Hardware des NAO befindet sich zur Ansicht im Anhang.

2.2.1. Computersystem

Das Mainboard, welches sich im Kopf des NAO als gut platziert erwiesen hat, setzt sich aus verschiedenen Hardwarekomponenten zusammen. Aus dem Hause Intel ist der Hauptprozessor mit dem Modell Atom Z530 integriert. Er leistet eine Grundtaktfrequenz von 1,6 GHz und besitzt einen Cache-Speicher von 512 KB. Weitere wichtige Hardwarekomponenten sind der 1-GB-Arbeitsspeicher und der Flash-Speicher, der mit einer Größe von 2 GB ausgestattet und an

2. Grundlagen

die Northbridge angebunden ist. Die Front Side Bus (FSB)-Taktfrequenz hat 533 MHz, sie dient als Schnittstelle des Hauptprozessors zur Northbridge. Die Hauptaufgabe des Hauptprozessors besteht darin, fortgeschrittene Module wie die Kommunikationswege über Wireless Local Area Network (WLAN) oder RS 232 sowie das Audio- und das visuelle System sorgfältig zu verwalten. Die Abbildung 2.2 zeigt die Computerarchitektur des NAO.

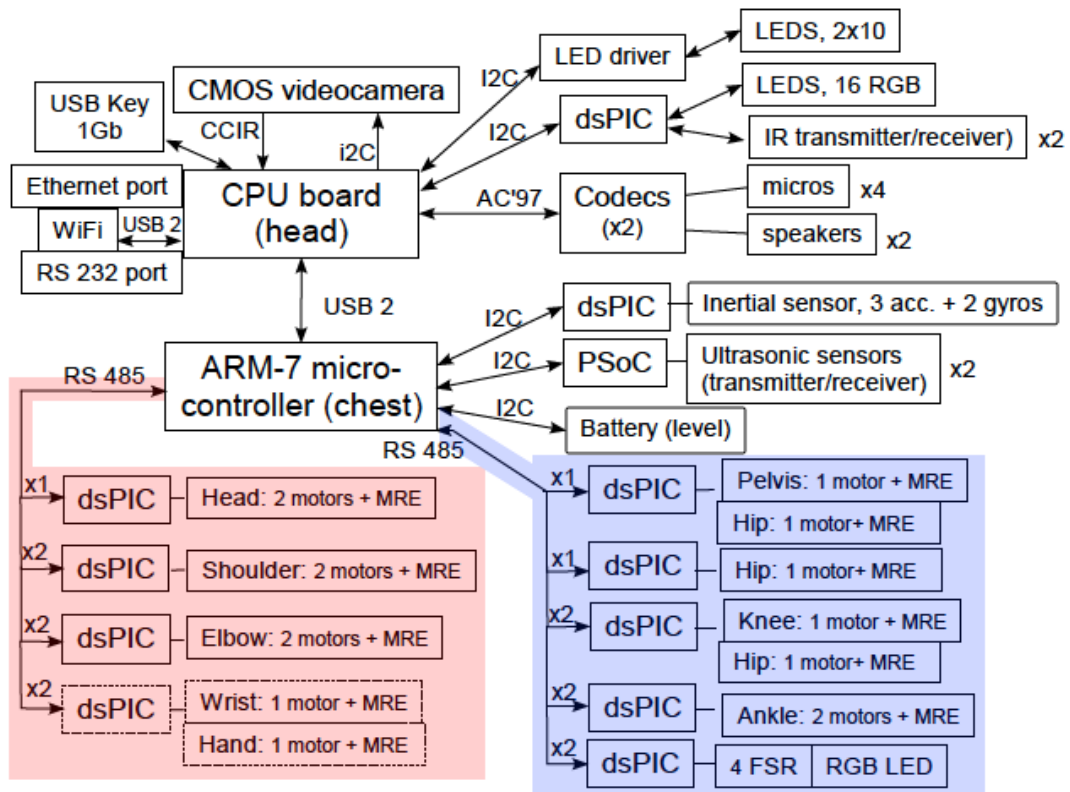


Abbildung 2.2.: Computerarchitektur des NAO [5].

Ein weiteres Computersystem mit einer Taktfrequenz von 60 MHz ist der ARM-7-Mikrocontroller. Er sorgt für die Kommunikation sowie Steuerung der Motoren und wurde im Torso des NAO verbaut. Steuerungsinformationen werden vom ARM-7 an die jeweiligen Körpermotoren delegiert, die von 16-Bit-dsPICS-Mikrochips entgegengenommen werden. Die Übertragung erfolgt über das Bussystem RS 485. Der Körper ist in zwei Ebenen unterteilt, die obere und untere Körperhälfte. Jede dieser Ebenen ist mit einer separaten Schnittstelle RS 485 anzusprechen. Die Kommunikation des ARM-7 zum Hauptprozessor ist unumgänglich, da sie z. B. für die Stabilität des Roboters zuständig ist [5].

2.2.2. Kinematik und Motorik

Den mechanischen Aufbau der humanoiden Roboter beschreibt innerhalb der Robotik die Kinematik. Der NAO besteht aus einzelnen Körperteilen, die jeweils mit komplexen Gelenken versehen sind, angetrieben werden die Gelenke durch Motoren. Dadurch bilden sich Ketten von gekoppelten Körperteilen. Der Torso bildet die Wurzel für jegliche Verkettung. Um den Roboter in Gang zu setzen, spielen die in Kapitel 2.2.2 erwähnten 25 DoF eine entscheidende Rolle, diese variable Größe dient zur Steuerung des NAO. Die Abbildung 2.3 stellt alle Gelenkpositionen dar, die auf der X-, Y- und Z-Achse einstellbar sind.

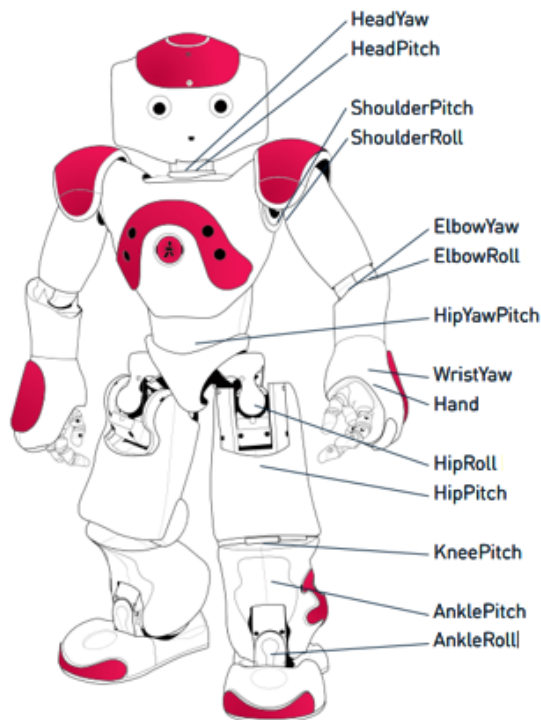


Abbildung 2.3.: Gelenkpositionierung des NAO [6].

Um den NAO fortzubewegen, bedarf es Aktuatoren, sie setzen die elektrischen Impulse in mechanische Bewegungen um. Die mechanischen Bewegungen werden mithilfe von 25 einzelnen elektrischen Motoren kontrolliert, sie setzen sich aus vier verschiedenen Motoren-Modellen zusammen: 22NT82213P, 12N88208E, 16GT83210E und DCX16S01GBKL651 [6]. Sie werden in zwei Reduktionsverhältnisse kategorisiert, die sich hinsichtlich der technischen Angaben bei der Beschleunigung und der Bremsfähigkeit unterscheiden. Genau ein Aktuator verändert somit einen Freiheitsgrad, der genau ein Gelenk bewegt. Informativ ist das Hüftgelenk, es

verändert zeitgleich das linke und das rechte Hüftgelenk mittels eines elektrischen Motors. In der folgenden Tabelle 2.1 ist die Unterbringung der 25 DoF im kinematischen Design des NAO ersichtlich.

Gelenkposition	Anzahl (DoF)	Gelenkname
Kopf	2	HeadYaw, HeadPitch
Arm (jeweils)	5	ShoulderPitch, ShoulderRoll, ElbowYaw, ElbowRoll, WristYaw
Becken	1	HipYawPitch
Bein (jeweils)	5	HipRoll, Hip Pitch, KneePitch, AnklePitch, AnkleRoll
Hand (jeweils)	1	Hand

Tabelle 2.1.: Verteilung der DoF im Körperbau des NAO

2.2.3. Sensorik

Die Sensorik ist mit den menschlichen Sinnesorganen gleichzusetzen und für das gesamte System von großer Bedeutung. Ohne jegliche Ansammlung von Informationen würde der NAO quasi im Dunkeln tappen. Um ein solches Szenario zu vermeiden, orientiert sich der NAO in seiner Umgebung, indem er durch seine Vielzahl von unterschiedlichen Sensoren jegliche relevanten Informationen sammelt sowie auswertet und zugleich seinen eigenen Zustand erfasst.

Kamera Der NAO erhält sein Sehvermögen durch zwei baugleiche VGA-Digitalkameras, welche sich an der Stirn (Blick nach vorn) und im Mund (Blick zum Boden) befinden. Die Digitalkameras sind vom Modell MT9M114 und liefern HD-auflösende Bilder von 720 p mit einer maximalen Auflösung von 1.280 x 960 px bei 30 fps. Die Digitalkameras liefern laut Datenblatt ab einem Abstand von 30 cm zum Gegenstand die schärfsten Bilder. Das Datenformat der Bilddateien wird als YUV422 abgespeichert. Mittels Konvertierung lassen sich die Bilder an ein gewünschtes Farbfeld anpassen. Anders sieht dies beim simultanen Zugriff der Digitalkameras aus. Dem NAO ist der Zugriff auf nur eine Digitalkamera zum selben Zeitpunkt gestattet. Die Auswahl der Digitalkameras wird entsprechend dem geeigneten Sichtfeld getroffen und dies schließt somit Stereo-Bilder aus [7].

Mikrophone Die Akustik der Umwelt wird über vier Mikrophone aufgezeichnet, die am Kopf des NAO installiert sind. Zwei der Mikrophone sind vorn angebracht und die

anderen beiden hinter den Ohrmuscheln. Die Anordnung der Mikrophone ist ideal für die Positionsbestimmung von Geräuschquellen durch eine Laufzeitunterscheidung der empfangenen Geräusche geeignet. Die Geräuschfrequenz liegt bei 20 Hz bis 20 kHz. Dieses ermöglicht dem NAO, aus bis zu 2 m Entfernung vollständige Sätze zu hören und zu orten [7].

Lautsprecher Um sich verbal auszudrücken oder eine Audiodatei abzuspielen, besitzt der NAO an beiden Ohren ein Stereosystem, bestehend aus zwei Lautsprechern [7].

Trägheitsmesssystem Das Trägheitsmesssystem besteht aus zwei Hardwarekomponenten: zum einen aus zwei Gyrometern, verteilt auf zwei Achsen, zum anderen aus einem Bewegungssensor mit drei Achsen. Die Hauptaufgabe des Trägheitsmesssystems besteht darin, den kinematischen Körper zu überwachen und diesen vor Stürzen zu schützen, dass durch Einfluss der momentanen Krafteinwirkung in den Körper einfließt. Über den Bewegungssensor kann die Geschwindigkeit z. B. aus dem Torso eingesehen werden. Das Gyrometer erbringt dahingehend Einsicht, ob sich der NAO in einer stabilen oder instabilen Haltung befindet, ob er liegt oder steht und ob er seine aktuelle Position verlässt [7].

2.3. Entwicklungssoftware des NAO

Um Applikationen für den NAO zu konzipieren, bedarf es zunächst eines Verständnisses der entsprechenden Softwarearchitektur (siehe Abbildung 2.4). NAOqi OS ist ein speziell angepasstes embedded x86-Linux-Betriebssystem, das unter der GNU-Lizenz stehende Gentoo-Derivat wurde von Aldebaran Robotics direkt auf den NAO zugeschnitten. Unter diesem Betriebssystem läuft die Software NAOqi. Während jedes Einschaltvorgangs des NAO startet NAOqi nach dem Booten des Betriebssystems automatisch, welches durch ein Skript initiiert wird, das unter dem Dateipfad `/etc/init.d/naoqi` zu finden ist [7].

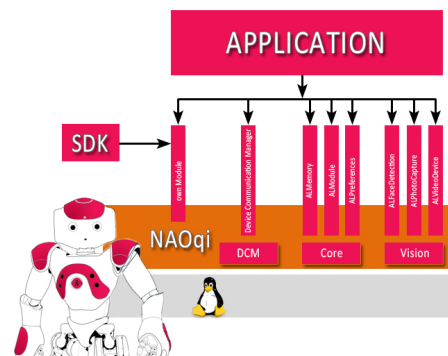


Abbildung 2.4.: Softwarearchitektur des NAO.

2.3.1. NAOqi

NAOqi ist die Software, die zur Steuerung und Kontrolle des NAO dient. Diese Software bringt die Ansprüche der Roboterprogrammierung mit sich. NAOqi stellt ein Event-, Ressourcen-, Synchronisierungs- sowie Parallelisierungsmanagement dar. Das Betriebssystem betreibt und spielt eine Vielzahl von Programmen und Bibliotheken ab. Darin sind auch diejenigen Programme und Bibliotheken enthalten, die für NAOqi erforderlich sind. Das NAOqi-Framework kann auf einem Computer installiert werden und bietet die Rahmenbedingungen, um Applikationen für den NAO zu implementieren und den Quellcode an einem simulierten NAO zu testen [8].

Die Nutzung der Methoden wird mithilfe des Frameworks von NAOqi mit verschiedenen Programmiersprachen und auf verschiedene Art und Weise angesprochen, genutzt und erweitert. Während des Initialisierungsprozesses von NAOqi wird die ausführende Instanz „Broker“ zur Verfügung gestellt. Ein Broker ist ein Objekt, das für das Auffinden von angeforderten Modulen zuständig ist. Der Broker liest beim Starten die Einstellungsdatei `autoload.ini` aus. Dem Broker sind nach erfolgreichem Auslesen sämtliche Bibliotheken und die darin enthaltenen Module bekannt. Folglich verwaltet der Broker einen Verzeichnisbaum von Modulen. Auf diesen Verzeichnisbaum kann intern oder extern über das Netzwerk zugegriffen werden. Die Abbildung 2.5 verdeutlicht die Einstellungsdatei von NAOqi.

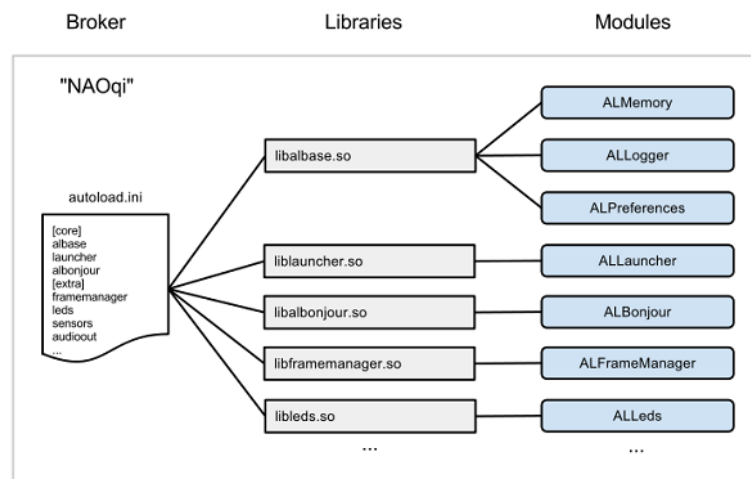


Abbildung 2.5.: Auslesung der `autoload.ini` Datei und die damit verbundene Instanziierung der Module [9].

In der nachfolgenden Abbildung 2.6 wird der Verzeichnisbaum, nach dem erfolgreichen Laden der Bibliothek, mit instanziierten Modulen dargestellt. Hieraus wird die Wichtigkeit des Instanz-Brokers deutlich, der als Wurzelknoten des Verzeichnisbaums mit den einzelnen Modulen verbunden ist und den Aufruf der Methoden delegiert, sei es lokal oder **Remote** über das Netzwerk.

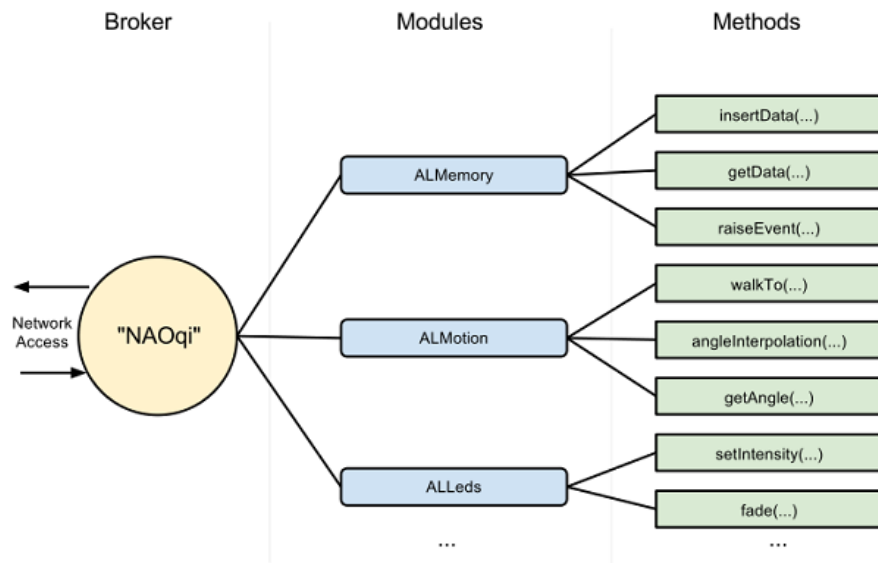


Abbildung 2.6.: Verzeichnisbaum mit der Instanz Broker als Wurzelknoten [9].

Eine Bibliothek besteht aus einem oder mehreren Modulen. Jedes Modul, das wiederum aus einer Ansammlung von Methoden besteht, ist somit nach dem erfolgreichen Starten des Brokers bekannt. So kann durch das NAOqi-Framework das Modul `ALTextToSpeech` benutzt werden. Dieses Modul ermöglicht dem NAO, verbale Kommunikation aufzunehmen, indem er über das Soundsystem einen vordefinierten Text in einer ihm bekannten Sprache ausgibt. Die Module wurden speziell für die Ansteuerung von einem Teil des Robotersystems entwickelt. Zur besseren Übersicht sind folgende Kategorien entstanden, in denen die Module eine geordnete Struktur bilden: *Core*, *Motion*, *Audio*, *Vision*, *People Perception*, *Sensor*, *Tracker*, *Diagnosis*, *DCM* und *Deprecated*. Eine Übersicht über diese Module wird in der Tabelle 2.2 festgehalten und bei der Konzeption der Szenarien in Kapitel 4.3 vertieft.

2. Grundlagen

Core	Vision	Audio
ALAutonomousLife	ALBacklightingDetection	ALAnimatedSpeech
ALBehaviorManager	ALBarcodeReader	ALAudioDevice
ALConnectionManager	ALCloseObjectDetection	ALAudioPlayer
ALExtractor	ALColorBlobDetection	ALAudioRecorder
ALMemory	ALDarknessDetection	ALDialog
ALModule	ALLandmarkDetection	ALSoundDetection
ALNotificationManager	ALLocalization	ALSoundLocalization
ALPreferenceManager	ALMovementDetection	ALSpeechRecognition
ALResourceManager	ALPhotoCapture	ALTextToSpeech
ALStore	ALRedBallDetection	ALVoiceEmotionAnalysis
ALSystem	ALSegmentation3D	
ALVisionExtractors	ALVideoDevice	People Perception
ALTabletService	ALVideoRecorder	ALBasicAwareness
ALUserSession	ALVisionRecognition	ALEngagementZones
ALWorldRepresentation	ALVisualCompass	ALFaceCharacteristics
PackageManager	ALVisualSpaceHistory	ALFaceDetection
		ALGazeAnalysis
Sensor	Deprecated	ALPeoplePerception
ALBattery	ALAudioSourceLocalization	ALSittingPeopleDetection
ALBodyTemperature	ALBonjour	ALWavingDetection
ALChestButton	ALFaceTracker	
ALFsr	ALLogger	Motion
ALInfrared	ALLauncher	ALAutonomousMoves
ALLaser	ALMotionRecorder	ALMotion
ALLeds	ALPreferences	ALNavigation
ALSensors	ALRedBallTracker	ALRecharge
ALSonar	ALRobotPose	ALRobotPosture
ALTouch		
	Diagnosis	DCM
Trackers	ALDiagnosis	DCM
ALTracker		

Tabelle 2.2.: Übersicht der gestellten Module im NAOqi API.

Um die Module zu nutzen, wird das Schlüsselkonzept vorgestellt. Es bedarf eines Einstiegspunkts beim Instanz-Broker, damit die Module genutzt werden können. Hierfür wird ein Proxy-Objekt erzeugt, welches das Verhalten eines Moduls festhalten soll. Mit den Eigenschaften meldet sich der Proxy beim Broker an und verlangt das Objekt, das der Proxy zu imitieren hat. Über das Proxy-Objekt kann nun jede Methode des Moduls angesprochen und genutzt werden. Abstrakt gesehen handelt es sich bei einem Proxy-Objekt um eine Art Voodoo-Puppe, d. h. eine Puppe, die das Verhalten eines Moduls imitieren kann.

2.3.2. Programmierung

Das NAOqi-Framework ist plattformübergreifend, was die Möglichkeit bietet, unter Linux, Mac oder in Windows-Betriebssystemen entwickelt und genutzt zu werden. Weiterhin ist sie sprachübergreifend. Eigens entwickelte Module können in das NAOqi-Framework integriert werden, wobei die Module ausschließlich in Python, C++ und JavaScript programmiert sind. Darüber hinaus bietet das NAOqi-Framework eine API, die es ermöglicht, über einen Wrapper C++ oder Python, Modulmethoden aus den Entwicklungssprachen Java und JavaScript aufzurufen (siehe Abbildung 2.7).

Programming Languages	Bindings running on		Choregraphe support	
	Computer	Robot	Build Apps	Edit code
Python	✓	✓	✓	✓
C++	✓	✓	⊘	⊘
Java	✓	⊘	⊘	⊘
JavaScript	✓	✓	✓	⊘

✓	OK
⊘	Not available

Abbildung 2.7.: Übersicht der Unterstützten Programmiersprachen [8].

In der Abbildung 2.7 ist ersichtlich, dass Python aufgrund der vollen Unterstützung, um einen Quellcode zu implementieren, und von der Ausführung her eine der leistungsstärksten Programmiersprachen ist. Mit Python kann der Quellcode remote über das Netzwerk oder lokal auf dem NAO ausgeführt werden. Es kann mit der mitgelieferten Entwicklungsumgebung „Choregraphe“ oder einer seiner Favoriten-Integrierte Entwicklungsumgebung (IDE), eigenständigen Applikationen und einer vorgefertigten Applikation in Python editiert werden.

Ein Modul ist eine Klasse mit Ansammlungen von Bibliotheken. Das Aufrufen eines Moduls kann remote oder lokal geschehen. Nachfolgend wird kurz auf Module, die sich remote oder lokal befinden, eingegangen.

Computer (remote): Module, die sich außerhalb des NAO befinden, können nur über das Netzwerk kommunizieren. Sie werden in einer ausführbaren Datei kompiliert und befinden sich außerhalb des NAO. Ein entfernt situiertes Modul kann sich mit anderen Modulen austauschen, indem eine Verbindung vom Main-Broker zum Broker des entfernt situierten Moduls stattfindet. Die Möglichkeit besteht darin, eine Broker-zu-Broker-Verbindung (gegenseitige Kommunikation) oder eine Broker-zu-Proxy-Verbindung (einseitige Kommunikation) aufzubauen.

Roboter (lokal): Bis auf Java lassen sich Module implementieren, was dem Entwickler erlaubt, diese lokal auf dem NAO zu instanzieren und auszuführen (siehe Abbildung 2.7). Der Quellcode wird fest in NAOqi integriert, indem er in Bibliotheken erstellt und beim Starten von NAOqi aus der Einstellungsdatei instanziiert wird. Dadurch, dass lokale Module im selben Prozess laufen, können deren Eigenschaften und Methoden untereinander eingesehen und genutzt werden.

Vorteil: Hohe Performance-Gewinnung, da die Netzwerkkommunikation entfällt.

Nachteil: Kommt es zum Absturz des Prozesses, was den Main-Broker betrifft, wird der NAO in seiner Bewegung erstarren oder umfallen, was zu seiner Beschädigung führen kann. Beim Absturz eines entfernten Prozesses dagegen, was den Broker des entfernten Moduls betrifft, wären die roboterspezifischen Module weiterhin ansprechbar, da die Verbindung zum Main-Broker konstant intakt ist.

Im folgenden Kapitel werden die Szenarien konkret vorgestellt und der Spielablauf im Detail spezifiziert. Die Szenarien ermöglichen in Kapitel 3 im Hinblick auf die funktionalen, technischen und nicht-funktionalen Anforderungen an eine Realisierung der vier Szenarien.

3. Analyse

Die im Grundlagenkapitel 2 dargelegten technischen Details des NAO dienen dazu, die Szenarien für die Realisierung zu bestimmen. Um die Anforderungen für die Präsentation des NAO zu formulieren, wird zu Beginn dieses Kapitels die Systemvision mit den dazugehörigen Szenarien vorgestellt. Im Anschluss daran werden die Anwendungsfälle ermittelt und der jeweilige Systemkontext wird bestimmt. Es folgt eine Analyse der konkreten Anwendungsfälle, zudem werden zum Schluss die Anforderungen identifiziert und spezifiziert.

3.1. Systemvision

Die interaktive Präsentation des NAO wird durch die Hochschule für Angewandte Wissenschaften Hamburg für verschiedene Arten der Außendarstellung genutzt. Die Zielgruppe entsprechen der Veranstaltungen umfasst Nutzer diverser Altersgruppen, welche überwiegend keine technischen Vorkenntnisse besitzen. Die interaktive Präsentation des NAO trägt zur Unterhaltung des Nutzers bei, woraus sich folgende Systemvision ergibt:

Das interaktive Präsentieren der Möglichkeiten der NAO-Plattform. Dazu gehören akustische und visuelle Verarbeitungen sowie die Durchführung komplexer Bewegungsabläufe.

Eine der Hauptveranstaltungen zur Präsentation des NAO stellt der jährlich stattfindende Girls' Day (Mädchen-Zukunftstag) dar, an dem die Hochschule für Angewandte Wissenschaften Hamburg regelmäßig teilnimmt. Da der Girls' Day dazu dient, Schülerinnen einen Einblick in eher durch Männer dominierte Branchen zu verschaffen, ist es somit von hoher Wichtigkeit diesen technische Informatik bzw. Technik möglichst auf spielerische Art und Weise zu vermitteln. Daraus ergibt sich eine weitere wichtige Systemvision:

Durch den Umgang und die Interaktivität mit dem NAO soll bei Schülerinnen das Interesse an technischen Berufen gezielt geweckt bzw. verstärkt werden.

Die zu konzeptionierende Anwendung wird auf den beiden Systemvisionen sowie der reichhaltigen Ausstattung und den vielseitigen Fähigkeiten des NAO (vgl. Kapitel 2.2) als idealen

3. Analyse

Grundlagen basieren. Die zu entwickelnde Anwendung soll eine Interaktion anhand von Kommunikation und Gesten ermöglichen, weshalb sich Spiele hierfür sehr gut eignen. Aus diesem Grund werden zwei der vier geplanten Szenarien als Spiele realisiert. Das erste Spielszenario ist ein gestenbasiertes Spiel, auch als „Schere, Stein, Papier“ bekannt. Das zweite Spielszenario soll im Gegensatz zum ersten Spiel den kommunikativen Aspekt hervorheben und wird deshalb als „Quiz-Spiel“ realisiert. Dabei wird der NAO die Rolle eines Moderators einnehmen und dem Nutzer verbal Fragen mit den damit verbundenen Antwortmöglichkeiten stellen. Als drittes Szenario wird ein Tanzszenario gewählt. Mithilfe von verschiedenen vom NAO vorgeführten Tänzen kann dies zur Unterhaltung der Nutzer beitragen. Als viertes und letztes Szenario wird ein Interaktivitätsszenario realisiert, das dem Nutzer erlaubt, das bestehende Tanzszenario zu erweitern. Dies soll dem Nutzer einen Einblick in die praktische Anwendungsentwicklung bezüglich des NAO gewähren. Während die ersten drei Szenarien dazu dienen, dem Nutzer die Möglichkeiten des NAO aufzuzeigen, dient das vierte Szenario dazu, dem Nutzer zu ermöglichen, die Tanzabläufe aus dem Szenario 1 individuell zu erweitern. Das vierte Szenario zielt somit darauf ab, durch die Möglichkeit zur eigenen praktischen Entwicklung das Interesse des Nutzers an technischen Berufen zu erhöhen. Ein grober Überblick über das System wird in Abbildung 3.1 dargestellt.

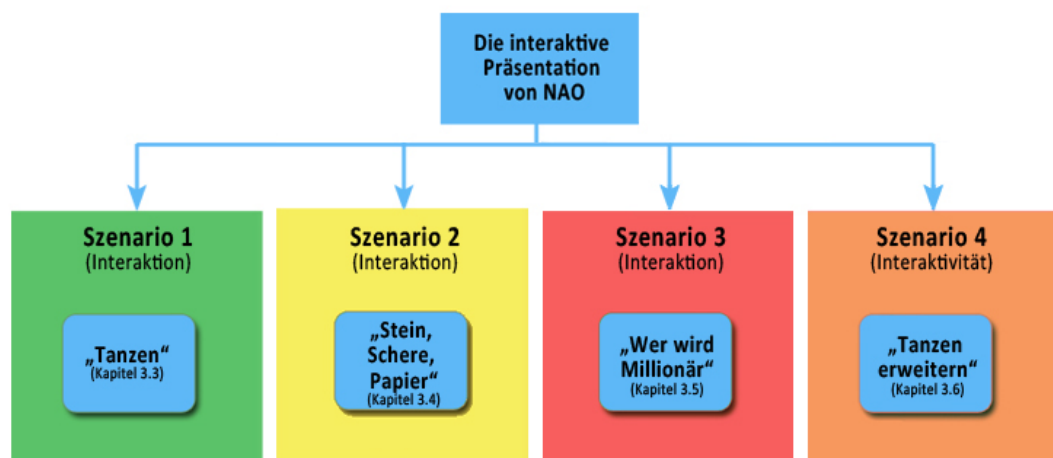


Abbildung 3.1.: Übersicht der zu entwickelnden Szenarien.

3.2. Anwendungsfälle

Anhand eines Unified Modeling Language (UML) Anwendungsfalldiagramms wird die Funktionalität des Systems festgehalten. Ersteres visualisiert die Präsentation des NAO mit den ermittelten Anwendungsfällen und den beteiligten Akteuren des Systems. Die Anwendungsfälle ergeben sich aus den einzelnen Szenarien aus den Abschnitten 3.4.1, 3.5.1, 3.6.1, 3.7.1. Die Stakeholder in diesem System sind die Nutzer. Sobald der NAO einen Nutzer erkannt hat, startet die interaktive Präsentation und der Nutzer bestimmt durch sein Verhalten den weiteren Verlauf.

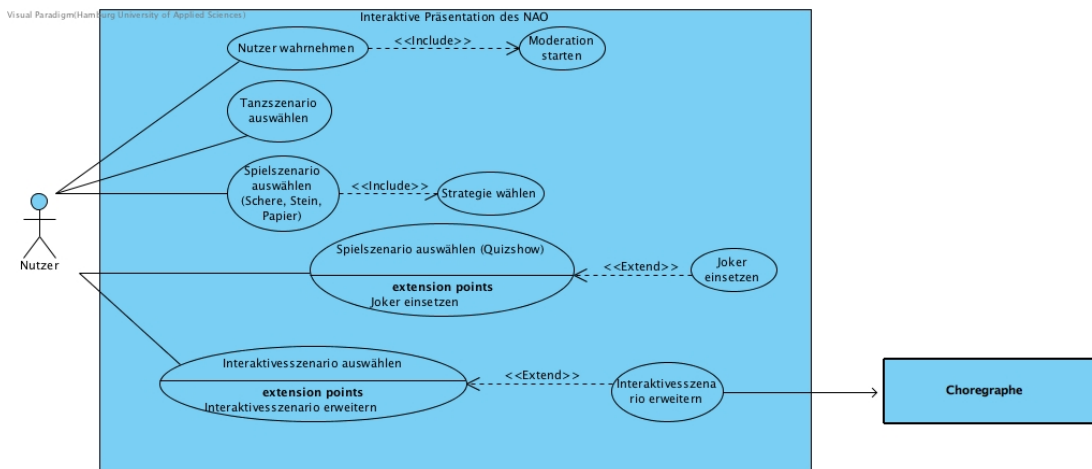


Abbildung 3.2.: Ermittelte Anwendungsfälle der interaktiven Präsentation des NAO.

In den kommenden Abschnitt werden die szenarienübergreifenden Anwendungsfallbeschreibungen tabellarisch erläutert. Sie ergeben sich aus den in Kapitel 3.2 ermittelten Anwendungsfällen, welche anhand eines Anwendungsfalldiagramms (siehe Abbildung 3.2) festgehalten wurden. In den einzelnen Abschnitten der Anforderungsspezifikationen werden speziell diejenigen Anwendungsfälle beschrieben, die für das jeweilige Szenario relevant sind.

Name	Nutzer wahrnehmen
Kurzbeschreibung	Der Nutzer startet die interaktive Präsentation von NAO.
Verwendete Anwendungsfälle	Moderation starten
Erweiterungspunkt	-
Akteur	Nutzer
Vorbedingung	Die interaktive Präsentation ist im Idle-Zustand.

3. Analyse

Ergebnis	Der Nutzer ist Erfolgreich registriert.
Nachbedingung	Die Moderation wird autonom gestartet.
Standard Ablaufschritte	<ol style="list-style-type: none">1. Der Nutzer schaut in die obere Kamera, um von NAO registriert zu werden.2. NAO begeben sich in die Initialstellung.3. NAO begrüßt den aktiven Nutzer.

Tabelle 3.1.: Anwendungsfallbeschreibung: "Nutzer wahrnehmen".

Name	Moderation starten
Kurzbeschreibung	NAO hält eine Einleitungsrede.
Verwendete Anwendungsfälle	-
Erweiterungspunkt	-
Akteur	NAO
Vorbedingung	Der Nutzer wurde erfolgreich vom NAO registriert.
Ergebnis	Die Moderation wurde beendet.
Nachbedingung	Ein Szenario kann vom Nutzer gewünscht werden.
Standard Ablaufschritte	<ol style="list-style-type: none">1. NAO trägt die Einleitungsrede mit den wichtigsten Regelungen und der Benutzung vor.2. NAO stellt die einzelnen Szenarien vor.

Tabelle 3.2.: Anwendungsfallbeschreibung: "Moderation starten".

3.3. Anforderungen

Auf Basis der konkreten Anwendungsfälle aus Abschnitt 3.2 werden in diesem Abschnitt die funktionalen, technischen und nicht-funktionalen Anforderungen, welche szenarioübergreifend sind, identifiziert.

Funktionale Anforderungen

- S0-FA-1** Nachdem die Anwendung gestartet worden ist, befindet sich der NAO im Idle-Zustand. Der Nutzer kann sich jetzt mithilfe der oberen Kamera (vgl. Kapitel 2.2.3) registrieren, indem der NAO dessen Gesicht wahrnimmt. Diese Registrierung verlangt kein Anlegen von Benutzerdaten.
- S0-FA-2** Registriert der NAO das Gesicht des Nutzers, beginnt eine Moderation. Im Rahmen dieser Moderation werden kurz alle vier Szenarien vorgestellt, die der Nutzer sprachlich auswählen kann.
- S0-FA-3** Das System identifiziert die Szenarien mit einer eindeutigen Identifikationsnummer. Sie bestehen aus den einstelligen Zahlen 1 bis 4.
- S0-FA-4** Wenn die Moderation beendet ist, fordert der NAO den Nutzer dazu auf, ihm die jeweilige Identifikationsnummer sprachlich mitzuteilen, um das gewünschte Szenario zu starten.

Technische Anforderungen

- S0-TA-1** Die Typenbezeichnung des NAO, mit der die Präsentation lauffähig ist, lautet H25, aufgrund der 25 DoF, die für das Tanzszenario benötigt werden. Zudem muss der NAO mit der Software NAOqi betrieben werden.

Nicht-funktionale Anforderungen

- S0-NFA-1** Zur Präsentation des NAO sind eine Netzwerkanbindung sowie ein Tisch mit den Maßen 200 x 200 cm mit einer Stromversorgung von mindestens drei Steckdosen notwendig.
- S0-NFA-2** Um das Wunschscenario auszuwählen, hat der Nutzer 30 Sekunden Zeit. Nach dem Ablauf dieser begibt sich der NAO in den Idle-Zustand zurück.

3.4. Anforderungsspezifikation - Szenario 1

Im kommenden Abschnitt wird eine gestenbasierte Interaktion anhand eines Tanzszenarios analysiert. Tanzen ist die Umsetzung von kontinuierlichen Bewegungen, passend zum Rhythmus der jeweiligen Musik. Die Tänze des NAO sollen der Unterhaltung des Nutzers dienen.

3.4.1. Tanz

Die Aufgabe beim ersten Szenario besteht darin, ein Tanzmedley mit dem NAO einzustudieren. Das Tanzmedley wird aus acht unterschiedlichen bekannten Tänzen (Macarena, Ententanz, Kung Fu, Hound Dog, I Feel Good, Uptown Funk, Loose Yourself und Gangnam Style) zusammengesetzt, daraus resultiert eine Ansammlung von Tänzen. Das Ziel des ersten Szenarios ist es, das Publikum durch sprachliche Interaktion in das Tanzmedley einzubinden. Der NAO bietet den Nutzern die Möglichkeit, den jeweils nächsten Tanz auszuwählen. Um dies zu gewährleisten, sind folgende Fragen zu klären:

- 1. Wie kann eine sprachliche Interaktion mit dem Nutzer des NAO sinnvoll verarbeitet werden?*
- 2. Wie schafft es der NAO, die unterschiedlichen Geschwindigkeiten eines Musiktitels taktgemäß durch rhythmische Körperbewegungen zu begleiten?*
- 3. Wie kann eine Choreographie ausgeführt werden, ohne dass der NAO aus dem Gleichgewicht kommt?*

3.4.2. Ablauf

Damit die Choreographie gestartet werden kann, muss der Nutzer beim NAO das gewünschte Tanzszenario anhand von dessen Identifikationsnummer verbal auswählen. Ist eine bestimmte Tanzeinlage der Choreographie erreicht, wird der Nutzer mit dem NAO interagieren. Ihm stehen zwei verschiedene Tänze (X, Y) zur Verfügung, damit der NAO die Choreographie wiederaufnimmt, um das Tanzszenario zu beenden. Sobald der erste Tanz (Y) durchgeführt worden ist, wird mit dem zweiten Tanz (X) fortgefahren. Daraus ergibt sich eine Variation der Choreographie von maximal zwei verschiedenen Abläufen.

3.4.3. Aktivitätsdiagramm

Im kommenden Abschnitt wird der Hauptanwendungsfall „Tanzszenario auswählen“ aus dem Systemkontext (siehe Abbildung 3.2) durch ein Aktivitätsdiagramm dargestellt (siehe Abbildung 3.3).

3. Analyse

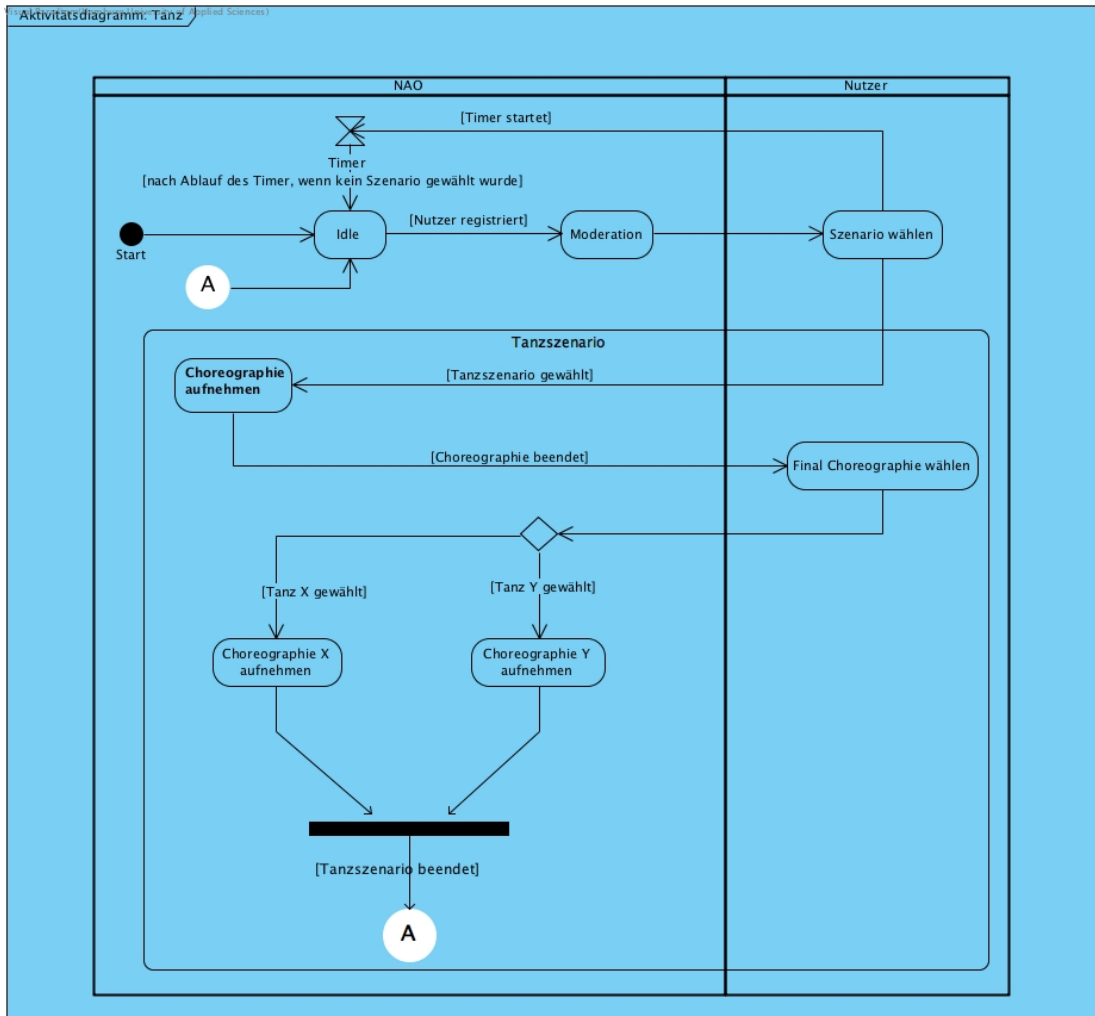


Abbildung 3.3.: Aktivitätsdiagramm: Tanzen.

Das Aktivitätsdiagramm enthält den Verlauf des Prozesses sowie Alternativabläufe des Szenarios „Tanz“. Aus dem Aktivitätsdiagramm gehen die Spezifikation der internen Abläufe hervor und die damit verbundene Funktionalität des Anwendungsfalls „Tanzszenario auswählen“. In der Tabelle 3.3 erfolgt im Anschluss die Beschreibung des Anwendungsfalls.

3. Analyse

Name	Tanzszenario auswählen
Kurzbeschreibung	NAO tanzt das Tanzmedley für den Nutzer vor. Nach Aufforderung vom NAO kann der Nutzer aus insgesamt zwei zu Verfügung stehenden Tänzen, die Schlusschoreographie wählen.
Verwendete Anwendungsfälle	-
Erweiterungspunkt	-
Akteur	Nutzer
Vorbedingung	Nach Aufforderung vom NAO, hat der Nutzer anhand der Identifikationsnummer "1" das Szenario gewählt.
Ergebnis	Die Choreographie wurde von NAO vorgetanzt und das Tanzszenario wird beendet.
Nachbedingung	Ein weiteres Szenario kann gestartet werden, wenn der Nutzer sich erneut bei NAO registriert hat.
Standard Ablaufschritte	<ol style="list-style-type: none"> 1. Der Nutzer wählt das Tanzszenario anhand der Identifikationsnummer "1". 2. Das Tanzszenario wird gestartet und NAO tanzt die Hauptchoreographie. 3. Nach Ablauf der Hauptchoreographie fordert NAO den Nutzer auf die Schlusschoreographie zu wählen. 4. Der Nutzer wählt die Schlusschoreographie anhand des Identifikationsbuchstaben. 5. NAO startet die Schlusschoreographie. 6. Nach Ablauf der Schlusschoreographie endet das Tanzszenario. 7. NAO begibt sich in den Idle-Zustand. 8. Der Nutzer kann sich erneut registrieren, um weitere Szenarien auszuwählen.
Alternative Ablaufschritte	<ol style="list-style-type: none"> 4.1. Der Nutzer wählt die Schlusschoreographie "A". <ol style="list-style-type: none"> 5. Die Schlusschoreographie "A" startet. 4.2. Der Nutzer wählt die Schlusschoreographie "B". <ol style="list-style-type: none"> 5. Die Schlusschoreographie "B" startet.

Tabelle 3.3.: Anwendungsfallbeschreibung: "Tanzszenario auswählen".

Die erreichte Grundlage dient der Spezifikation der funktionalen, technischen und nicht-funktionalen Anforderungen, was aus der Vertiefung des Anwendungsfalls anhand der Anwendungsfallbeschreibung und durch ein Aktivitätsdiagramm festgehalten wird.

3.4.4. Anforderungen

Im kommenden Abschnitt werden die funktionalen, technischen und nicht-funktionalen Anforderungen für das Tanzszenario aufgeführt.

Funktionale Anforderungen

- S1-FA-1** Das Tanzszenario wird im System anhand einer eindeutigen Identifikationsnummer repräsentiert. Mit der Identifikationsnummer „1“ kann der Nutzer nach Aufforderung die Nummer des gewünschten Tanzszenarios sprachlich angeben.
- S1-FA-2** Die Schlusschoreographien sind im System mit einem eindeutigen Buchstaben identifizierbar. Diese Identifikationsbuchstaben lauten „A“ und „B“. Wenn der NAO zur Aufforderung der Schlusschoreographien bittet, muss der Nutzer den entsprechenden Identifikationsbuchstaben sprachlich angeben.

Technische Anforderungen

- S1-TA-1** Der NAO muss dazu in der Lage sein, die Audiodatei über das integrierte Soundsystem abzuspielen.
- S2-TA-2** Die integrierten Mikrophone des NAO zeichnen die Identifikationsnummer und den Identifikationsbuchstaben des Nutzers auf.
- S1-TA-3** Der NAO soll im Verlauf des Tanzens alle 25 DoF genutzt haben.

Nicht-funktionale Anforderungen

- S1-NFA-1** Der NAO muss auf jede Anfrage des Nutzers innerhalb von 30 Sekunden reagieren.
- S2-NFA-2** Der NAO wartet nach Anfragen an den Nutzer bis zu 30 Sekunden. Nach dem Ablauf dieser Zeit versetzt sich der NAO in den Idle-Zustand.

3.5. Anforderungsspezifikation - Szenario 2

Im kommenden Abschnitt wird ein gestenbasiertes Spiel analysiert. Unter dem Namen „Schere, Stein, Papier“ ist es eines der bekanntesten statischen Spiele, die im Kindesalter gelehrt werden, um z. B. Entscheidungen zu treffen oder Konflikte friedlich zu lösen. Bei statischen Spielen wird zwischen den Spielern zeitgleich eine Entscheidung gewertet. Zum Zeitpunkt der Bestimmung einer Geste weiß der Mitspieler nicht, welche Wahl von seinem Kontrahent getroffen wurde, folglich ist dies ein Spiel mit unvollständiger Information.

3.5.1. Schere, Stein, Papier

Die Zahl der Spieler ist klar definiert und besagt, dass zwei Personen daran beteiligt sein müssen. Die teilnehmenden Spieler sprechen rhythmisch den Namen des Spieles und bewegen parallel dazu eine geballte Hand auf und ab. Die Festlegung eines Spielzuges bzw. der Geste darf bis zur Aussprache des letzten Wortes, „Papier“, unbeobachtet vom Kontrahenten, gewählt werden. Die Symbole, zwischen denen hierbei zu wählen ist, sind „Stein“, „Schere“, „Papier“. Im Anschluss daran wird das gewählte Symbol mit der Hand geformt und zur Veröffentlichung aller Mitspieler parallel zu ihrer Geste präsentiert. Während des Spiels kann dies zu einem Fehlstart führen. Dies wird instinktiv zum Vorteil des jeweiligen Mitspielers genutzt und blitzartig wird von diesem die jeweils überlegene Strategie bzw. das jeweils ausstechende Symbol gewählt. Um einen Fehlstart zu vermeiden, gibt es Synchronisierungshilfen wie das rhythmische Aussprechen des Namens des Spieles sowie die kontinuierliche Auf-und-ab-Bewegung des Armes. Dies sind Faktoren, die Spielbeginn und -ende sowie die Synchronisierung mehrerer Spieler zugleich ermöglichen. Die Auswertungsregelung besagt Folgendes: *Stein schleift Schere, Schere schneidet Papier und Papier wickelt Stein ein*. Wählen beide Spieler dasselbe Symbol während der Auswertung, so ist die Spielrunde als „unentschieden“ zu werten. Ein Spiel besteht aus mindestens drei Spielrunden oder es dauert so lange, bis ein klarer Sieger hervorgeht. Ein klarer Sieger zu sein, bedeutet somit genau drei Spielrunden zu gewinnen. In der Spieltheorie werden solche Spiele als Nullsummenspiele bezeichnet. Die Tabelle 3.4 verdeutlicht das Nullsummenspiel anhand der Auszählung.

Spieltheorie

Eine Spieltheorie stellt ein mathematisches Modell dar, das einer Entscheidungssituation ausgesetzt ist. Wessler [10] definiert im Hinblick auf die Spieltheorie das, was zu einem Spiel gehört, wie folgt:

1. Zwei oder mehr Spieler
2. Eine Ausgangs-Spielposition
3. Eine Menge sogenannter Strategien (Möglichkeiten, Aktionen auswählen)
4. Eine Festlegung der Reihenfolge, in der die Spieler ihre Strategie wählen dürfen
5. Resultierend aus der Strategieauswahl ist allen die Übersicht über jegliche Spielpositionen bekannt

Laut Wessler [10] wird die oben beschriebene Definition der Spieltheorie für das Spiel „Schere, Stein, Papier“ in folgendem Fall als erfüllt angesehen:

3. Analyse

1. Es gibt zwei einzelne Spieler.
2. Die Ausgangs-Spielposition ist „leer“.
3. Es ist zu jedem Zeitpunkt möglich, aus den drei Symbolen eine Auswahl zu treffen.
4. Die Spieler sind verpflichtet, gleichzeitig das Symbol zu wählen und dieses anzuzeigen.
5. Insgesamt neun verschiedene Tupel können aus der Strategiewahl der Symbole durch die Spieler entstehen.
6. Das Spielende wird bewirkt, wenn beide Spieler je ein Symbol gewählt haben.
7. Durch die Auswertungsregelung ist am Ende klar, wer als Sieger hervorgeht.

Strategie

Ein Spielzug erfordert die Auswahl eines der oben genannten drei Symbole durch beide Spieler. Die Entscheidung für ein Symbol in einer oder mehreren Spielrunden beschreibt die Strategie in der Spieltheorie. Wessler [10] teilt die möglichen Strategien in drei Kategorien ein:

Reine Strategie: Der Spieler trifft die Entscheidung für ein Symbol in jeder einzelnen Spielrunde neu.

Langzeit Strategie: Der Spieler trifft die Entscheidung für ein Symbol für mehrere Spielrunden oder den gesamten Spielverlauf.

Gemischte Strategie: Der Spieler trifft die Entscheidungen zufällig, beispielsweise mithilfe einer Wahrscheinlichkeitsverteilung durch einen Würfel.

Auszahlungsmatrix

		B		
		Schere	Stein	Papier
A	Schere	0/0	-1/1	1/-1
	Stein	1/-1	0/0	-1/1
	Papier	-1/1	1/-1	0/0

Tabelle 3.4.: Auszahlungsmatrix: Auswertungsregelung.

Die Auszahlungsmatrix 3.4 bewertet den Gewinner mit dem Wert „1“ und den Verlierer mit „-1“. Kommt es zu einem Unentschieden, werden beide Spieler mit „0“ bewertet. Die Matrix wird wie folgt gelesen: Der erste Wert gilt dem Zeilen-Spieler (A). Der zweite Wert gilt dem Spalten-Spieler (B). Wenn z. B. die Entscheidung eines Spielzuges „Stein“ für den Spieler A zutraf und

sich der Spieler B für „Papier“ entschieden hat, ist der Spieler B bei dieser Entscheidung dem Spieler A überlegen. Die Auswertungsmatrix hält dieses wie folgt fest „-1/1“. Ist einem Spieler die Strategie des anderen bekannt, so ist es möglich, eine Langzeit-Strategie für den gesamten Spielverlauf aufzubauen. Folglich kann der jeweilige Spieler seine Gewinnchancen erhöhen.

3.5.2. Ablauf

Das Starten des Spieles „Schere, Stein, Papier“ wird, wie das Szenario 1, nach der Aufforderung des Nutzers hierzu seitens des NAO, anhand der Identifikationsnummer verbal ausgewählt. Nach dem erfolgreichen Starten des Spieles erklärt der NAO seinem Kontrahenten zunächst die Spielregelungen und Randbedingungen, um einen erfolgreichen Ablauf zu garantieren. Anhand der ermittelten Strategien aus dem Abschnitt 3.5.1 kann der Nutzer den Schwierigkeitsgrad für den NAO setzen, welcher im Konzept detaillierter spezifiziert wird. Gespielt wird so lange, bis der NAO oder der Nutzer seinem Kontrahenten dreimal mit der von ihm ausgewählten Strategie überlegen ist. Der NAO bekleidet bei diesem Spiel zwei Rollen: Er ist zugleich einer der „Spieler“ und „Schiedsrichter“.

3.5.3. Aktivitätsdiagramm

Ein weiterer Hauptanwendungsfall, „Spielszenario auswählen (Stein, Schere, Papier)“, wird anhand eines Aktivitätsdiagramms dargestellt. Mithilfe dieses Diagramms (siehe Anhang A) werden der Verlauf des Prozesses und die Alternativabläufe des zweiten Szenarios verdeutlicht. Im Anschluss erfolgt die Anwendungsfallbeschreibung in Tabellenform.

Das Aktivitätsdiagramm (siehe Abbildung 1) besteht aus folgenden zwei Aktivitätsbereichen:

- Nutzer
- NAO

Der Beginn des Ablaufs startet an beiden Startpunkten, d. h. beim Nutzer und beim NAO, parallel. Trotzdem wartet der NAO auf das Eintreffen des Signales „NAO!“. Der Nutzer sucht so lange, bis er eine Einstellung gefunden und gewählt hat, mit der NAO gegen den Nutzer antritt. Somit wird das Signal „NAO!“ ausgesendet (der Nutzer kommuniziert sprachlich „Easy“, „Medium“ oder „Hard“ aus). Empfängt der NAO das Signal „NAO!“, nimmt er die Einstellung eines Nutzers auf und setzt diese für den gesamten Spielverlauf fest. Der NAO leitet daraufhin den Spielbeginn ein und fordert den Nutzer dazu auf, sich für ein Symbol zu entscheiden, woraufhin dieser wiederum eines aussucht. Kommt es zur Aufdeckung der Symbole, wird vom

3. Analyse

Nutzer das Signal „Auswertung!“ an den NAO gesendet, woraufhin dieser die Auswertung der Spielrunde vornimmt und dem Nutzer das Ergebnis dieser mitteilt. Ist nach dieser Spielrunde ein klarer Sieger hervorgegangen, wird das Spiel beendet, andernfalls kommt es zu einer neuen Spielrunde.

Name	Spielszenario auswählen (Schere, Stein, Papier)
Kurzbeschreibung	Der Nutzer spielt "Schere, Stein, Papier" gegen NAO, bis ein klarer Sieger hervorgeht.
Verwendete Anwendungsfälle	Strategie wählen
Erweiterungspunkt	-
Akteur	Nutzer, NAO
Vorbedingung	Der Nutzer hat das zweite Szenario "Schere, Stein, Papier" verbal gewählt, nach Aufforderung vom NAO.
Ergebnis	Das Spiel wurde beendet.
Nachbedingung	Ein weiteres Szenario kann gestartet werden, wenn der Nutzer sich erneut beim NAO registriert hat.
Standard Ablaufschritte	<ol style="list-style-type: none"> 1. Der Nutzer wählt das Spielszenario "Schere, Stein, Papier". 2. Das Spiel startet nach erfolgreicher Wahrnehmung vom Wunsches des Nutzers. 3. Der Nutzer stellt den Schwierigkeitsgrad beim NAO ein. 4. NAO teilt nach jeder Spielrunde den Gewinner mit. 5. Beide Spieler versuchen, das Ziel zu erreichen. 6. Das Spiel ist beendet, wenn ein Spieler drei Spielrunden gewonnen hat. 7. NAO begeben sich in den Idle-Zustand zurück. 8. Der Besucher kann sich erneut registrieren, um weitere Szenarien auszuwählen.
Alternativ Ablaufschritte	<ol style="list-style-type: none"> 3.1. Der Nutzer stellt die Langzeitstrategie beim NAO ein. <ol style="list-style-type: none"> 4. NAO wird fortlaufend das Spiel mit der Strategie "Dèjà-vu" spielen. 3.2. Der Nutzer stellt die gemischte Strategie beim NAO ein. <ol style="list-style-type: none"> 4. NAO wird fortlaufend das Spiel mit der Strategie "Random" spielen. 3.3. Der Nutzer stellt die Mogel-Strategie beim NAO ein. <ol style="list-style-type: none"> 4. NAO wird fortlaufend das Spiel mit der Strategie "Mogel" spielen.

Tabelle 3.5.: Anwendungsfallbeschreibung: "Spielszenario auswählen (Schere, Stein, Papier)".

3.5.4. Anforderungen

Ausgehend von den Anwendungsfällen in Abschnitt 3.2 werden nachfolgend die funktionalen, technischen und nicht-funktionalen Anforderungen für das Spiel „Schere, Stein, Papier“ identifiziert.

Funktionale Anforderungen

- S2-FA-1** Das Spielszenario „Schere, Stein, Papier“ wird im System anhand einer eindeutigen Identifikationsnummer repräsentiert. Mittels der Identifikationsnummer „2“ kann der Nutzer das Spielszenario „Schere, Stein, Papier“, nach der Aufforderung hierzu seitens des NAO starten.
- S2-FA-2** Die Strategien sind im System mit einem eindeutigen Buchstaben identifizierbar. Die Identifikationsbuchstaben lauten „A“, „B“ und „C“. Der Nutzer muss eine der in Kapitel 4.5.2 vorgestellten Strategien für den NAO durch den Identifikationsbuchstaben sprachlich auswählen.
- S2-FA-3** Dem Nutzer stehen drei Symbolbilder zur Verfügung, stellvertretend für die eigene Hand, um das getroffene Symbol dem des vom NAO gewählten gegenüberzustellen.
- S2-FA-4** Der NAO erfasst die Symbole des Nutzers über die obere Kamera (vgl. Kapitel 2.2.3). Die erfassten Symbole des Nutzers wertet der NAO anhand der Auszahlungsmatrix entsprechend aus und speichert das Ergebnis.
- S2-FA-5** Das Spiel wird vom NAO beendet, sobald die Ergebnisse der drei gewonnenen Spielrunden bei einem der Kontrahenten verzeichnet wurden.

Technische Anforderungen

- S1-TA-1** Die Symbole des Nutzers müssen mit der oberen Kamera des NAO aufgezeichnet werden.
- S2-TA-2** Die Einstellungsmöglichkeiten des Nutzers werden von den integrierten Mikrofonen vom NAO aufgenommen.

Nicht-funktionale Anforderungen

- S2-NFA-1** Der NAO wartet bei der Erfassung der Symbole des Nutzers bis zu 30 Sekunden. Nach dem Ablauf dieser Zeit verzeichnet der NAO einen Sieg für sich.

3.6. Anforderungsspezifikation – Szenario 3

Im kommenden Abschnitt wird die Anforderungsspezifikation des Szenarios 3 erörtert. Das Szenario soll die kommunikative Interaktion mit dem NAO anhand eines Quiz-Spieles repräsentieren. Dem Nutzer können bis zu acht Fragen hintereinander gestellt werden, was nur durch richtiges Beantworten der vorherigen Frage erreicht wird. Scheitert der Nutzer – auch nach dem Gebrauch des Jokers – an einer Frage, so wird das Spiel beendet.

3.6.1. Quiz-Show

Das Spiel benötigt einen Kandidaten und einen Moderator. Der Kandidat ist der Nutzer und die Rolle des Moderators übernimmt der NAO. Im Spielverlauf gibt es maximal acht Multiple-choice-Fragen mit steigendem Schwierigkeitsgrad. Zu jeder Frage stehen vier Antwortmöglichkeiten zur Verfügung, von denen jeweils nur eine Antwort richtig ist. Das Spiel ist beendet, sobald der Nutzer alle acht Fragen hintereinander korrekt beantwortet hat oder an einer scheitert. Dabei stehen dem Kandidaten pro Spiel drei Joker zur Verfügung. Bei der Wahl des Jokers werden zwei falsche Antworten eliminiert, sodass die Auswahl um 50 % reduziert wird. Zu jeder gestellten Frage kann ein Joker genutzt werden, bis alle drei Joker in Anspruch genommen worden sind.

3.6.2. Ablauf

Einen Spielverlauf startet der Nutzer durch Spracheingabe der Identifikationsnummer beim NAO. Nach dem erfolgreichen Starten des Spieles erklärt der NAO seinem Kandidaten die Spielregeln und die dazu notwendigen Randbedingungen, um einen erfolgreichen Spielverlauf zu garantieren. Der NAO stellt seinem Kandidaten im Anschluss jeweils eine Frage mit vier Antwortmöglichkeiten. Der Nutzer muss dem NAO seine Antwort verbal mitteilen. Kommt es zu einem Engpass, hat der Nutzer die Möglichkeit, einen von insgesamt drei Jokern auf die vom NAO gestellte Frage anzuwenden. Gespielt wird so lange, bis der Besucher an einer Frage scheitert, die zeitliche Grenze zum Antworten (30 Sekunden) überschreitet oder durch das richtige Beantworten aller acht Fragen das Ziel erreicht. Der NAO bekleidet im Spiel zwei Rollen: Er ist der „Moderator“, der die Fragen stellt, und zugleich der „Schiedsrichter“.

3.6.3. Aktivitätsdiagramm

Anhand eines Aktivitätsdiagramms wird der Hauptanwendungsfall „Spielszenario auswählen (Quiz-Show)“ aus dem Systemkontext 3.2 dargestellt (siehe Anhang A). Mithilfe des Akti-

3. Analyse

vitätsdiagramms werden die Standards und Alternativabläufe des Spieles festgehalten. Die dazugehörige Anwendungsfallbeschreibung wird in Form einer Tabelle 3.6 dargestellt.

Das Aktivitätsdiagramm (siehe Abbildung 2) besteht aus zwei Aktivitätsbereichen:

- Nutzer
- NAO

Parallel an beiden Startpunkten beginnt der Ablauf beim Nutzer und beim NAO. Der Nutzer wartet zunächst auf das Signal „Nutzer!“. Erhält der Nutzer das Signal „Nutzer!“ vom NAO, so nimmt er die aktuell gestellte Frage mit den vier Antwortmöglichkeiten auf. Der Nutzer muss die Frage korrekt beantworten, damit die nächste Frage gestellt wird. Hierfür hat er die Möglichkeit, bei Engpässen einen von insgesamt drei Jokern zu wählen. Kennt der Nutzer die Antwort auf die Frage, wird dieses dem NAO mitgeteilt und üblicherweise wird dann das Signal „Antwort!“ vom Nutzer gesendet, das den Beginn einer Aktivität anregt. Die Aktivität führt zur Auswertung sowie zur Bekanntgabe der abgegebenen Antwort. Wurde sie als korrekt ausgewertet, stellt der NAO die nächste Frage, andernfalls wird das Spiel beendet. Kam es zum Ziehen eines Jokers, startet eine weitere Aktivität, was die Antwortmöglichkeiten von vier auf zwei reduziert und dieses dem Nutzer mitteilt. Vorab wird geprüft, ob das Nutzen eines Jokers noch erlaubt ist, falls ja, wird die Aktivität ausgeführt, andernfalls erwartet der NAO eine Antwort.

Name	Spielszenario auswählen (Quiz-Sow)
Kurzbeschreibung	Der Nutzer spielt das Szenario "Quiz-Sow" mit NAO.
Verwendete Anwendungsfälle	
Erweiterungspunkt	Joker einsetzen
Akteur	Nutzer, NAO
Vorbedingung	Nach Aufforderung vom NAO, hat der Nutzer das dritte Szenario "Quiz-Show" gewählt.
Ergebnis	Das Spiel wurde beendet.
Nachbedingung	Ein weiteres Szenario kann gestartet werden, wenn der Nutzer sich erneut beim NAO registrieren hat.

Standard Ablaufschritte	<ol style="list-style-type: none"> 1. Der Nutzer wählt das Spielszenario "Quiz-Show". 2. Das Spiel wird autonom gestartet. 3. Der Nutzer wird vom NAO in die Spielregelungen eingeweiht. 4. NAO stellt die Frage mit den vier dazugehörigen Antwortmöglichkeiten an den Nutzer. 5. Der Nutzer wählt eine Antwort aus den Antwortmöglichkeiten und teilt diese verbal dem NAO mit. 6. NAO wertet die Antwort des Nutzers aus und teilt die Bewertung mit. 7. Das Spiel ist beendet sobald alle zehn Fragen korrekt beantwortet sind.
Alternativ Ablaufschritte	<ol style="list-style-type: none"> 6.1. Der Nutzer hat die gestellten Fragen korrekt beantwortet. <ol style="list-style-type: none"> 7. NAO gratuliert zum Sieg. Das Spiel wird beendet und NAO begibt sich in den Idle-Zustand zurück. 6.2. Der Nutzer scheitert an der aktuellen Frage. <ol style="list-style-type: none"> 7. NAO beendet das Spiel an dieser Stelle und begeben sich in den Idle-Zustand zurück. 4.1. Der Nutzer wählt für die aktuelle Frage einen Joker <ol style="list-style-type: none"> 4.2. NAO eliminiert zwei von vier Antwortmöglichkeiten

Tabelle 3.6.: Anwendungsfallbeschreibung: "Spielszenario auswählen (Quiz-Show)".

3.6.4. Anforderungen

Es folgt eine Darstellung der funktionalen, technischen und nicht-funktionalen Anforderungen für das Spiel „Quiz-Show“, welche spezifisch für das Szenario gelten.

Funktionale Anforderungen

- S3-FA-1** Das Spielszenario „Quiz-Show“ wird im System anhand einer eindeutigen Identifikationsnummer repräsentiert. Mit der Identifikationsnummer „3“ kann der Nutzer das Spielszenario „Quiz-Show“ nach der Aufforderung hierzu vom NAO starten.
- S3-FA-2** Die Antwortmöglichkeiten sind im System mit einem eindeutigen Buchstaben identifizierbar. Die Identifikationsbuchstaben lauten „A“, „B“, „C“ und „D“. Der Nutzer muss eine der vier Antwortmöglichkeiten durch den Identifikationsbuchstaben sprachlich auswählen.

- S3-FA-3** Der NAO muss dazu in der Lage sein, maximal acht Fragen mit jeweils vier Antwortmöglichkeiten zu stellen. Eine neue Frage wird gestellt, sobald die vorherige Frage korrekt beantwortet wurde.
- S3-FA-4** Das System muss dem Nutzer drei Joker zur Verfügung stellen. Beim Nutzen eines Jokers fallen bei einer Frage jeweils zwei von vier Antwortmöglichkeiten weg.
- S3-FA-5** Das Spiel wird beendet, wenn alle acht Fragen korrekt beantwortet worden sind oder eine Frage falsch beantwortet worden ist oder der Nutzer den NAO länger als 30 Sekunden auf eine Antwort hat warten lassen.

Technische Anforderungen

- S3-TA-1** Das Auffassen der Antworten und die Nutzung der Joker werden mit den integrierten Mikrofonen vom NAO aufgenommen.

Nicht-funktionale Anforderungen

- S3-NFA-1** Der NAO wartet jeweils 30 Sekunden auf die Antwort des Nutzers. Ist nach dem Ablauf dieser Zeit keine Antwort aufgezeichnet worden, wird das Spiel beendet.

3.7. Anforderungsspezifikation – Szenario 4

Der kommende Abschnitt verfolgt das Ziel, das Interaktivitätsszenario zu analysieren. Als Framework dient das gesamte Szenario 1 als Grundbasis für den Aufbau des Szenarios 4 „Tanzszenario erweitern“.

3.7.1. Tanzszenario erweitern

Im ersten Szenario (vgl. Kapitel 3.4.1) wurden verschiedenartige Tanzeinlagen durch ein Tanzmedley vom NAO vorgeführt. Das kommende Szenario beinhaltet einen einzigen Tanz, den der NAO vorzuführen hat. Dieser Tanz wurde in der Choreographie von Szenario 1 nun unvollständig realisiert. Der Erweiterungspunkt bei diesem Szenario bezieht sich auf die Vervollständigung des Tanzes, welche interaktiv vom Nutzer vorgenommen wird. Die Vervollständigung der Tanzeinlage wird mit der mitgelieferten Entwicklungsumgebung Choregraphie durchgeführt. Es handelt sich hierbei um den Tanz „Macarena“. Dieser scheint hierfür bestens geeignet, da das Zielpublikum die einzelnen Tanzschritte wahrscheinlich noch in Erinnerung haben.

3.7.2. Ablauf

Das Starten dieses Szenarios wird, wie bei allen anderen Szenarien auch, vom Nutzer nach Aufforderung durch den NAO anhand der Identifikationsnummer verbal ausgewählt. Nach dem erfolgreichen Starten des unvollständigen Szenarios klärt der NAO den Besucher zunächst über das spezielle Szenario auf. Der Nutzer wird gebeten, die Tanzschritte genau zu beobachten und die fehlenden Tanzschritte zu notieren. Die Haupttanzschritte werden dem Besucher als Vorlage mitgegeben, um sie leichter zu identifizieren. Nach erfolgreicher Wahrnehmung der fehlenden Tanzschritte kann der Nutzer versuchen, die Tanzschritte mit Choregraphie zu vervollständigen und anschließend zu testen.

3.7.3. Aktivitätsdiagramm

Im kommenden Abschnitt wird der Hauptanwendungsfall „Interaktives Szenario aus dem Systemkontext“ von der Abbildung 3.2 durch ein Aktivitätsdiagramm (siehe Anhang A) vertieft und schriftlich festgehalten. Das Diagramm hält den Verlauf des Prozesses fest und verdeutlicht Alternativabläufe des Szenarios 4.

Das Aktivitätsdiagramm (siehe Abbildung 3) besteht aus folgenden zwei Aktivitätsbereichen:

- Nutzer
- NAO

Der Ablauf beginnt am Startpunkt beim NAO. Der NAO fängt an, den unvollständigen Tanz vorzuführen. Sobald die Endposition eingenommen worden ist, fordert der NAO den Nutzer dazu auf, das Szenario interaktiv zu vollenden, und sendet hierzu das Signal „Choregraphie!“. Empfängt der Nutzer das Signal „Choregraphie!“, nimmt er die Herausforderung zur Vollendung der Choreographie (im Idealfall) auf. Hat der Nutzer die fehlenden Tanzbewegungen mithilfe der IDE aufgenommen, wird das Szenario auf dem NAO erneut installiert. Der Nutzer wird erneut das Szenario nach dem Installieren auswählen, um das vollendete Szenario zu betrachten. Der Nutzer kann das Szenario beliebig oft erweitern.

Name	Tanzszenario erweitern
Kurzbeschreibung	NAO tanzt das unvollständige Tanzszenario für den Nutzer vor. Nach Aufforderung vom NAO kann der Nutzer anhand der Entwicklungsumgebung das Tanzszenario erweitern.
Verwendete Anwendungsfälle	-

3. Analyse

Erweiterungspunkt	Interaktivesszenario erweitern
Akteur	Nutzer, NAO
Vorbedingung	Der Nutzer hat das vierte Szenario "Tanzszenario erweitern" verbal nach Aufforderung vom NAO gewählt.
Ergebnis	Das unvollständige Tanzszenario wurde beendet.
Nachbedingung	Der Nutzer soll das Tanzszenario erweitern.
Standard Ablaufschritte	<ol style="list-style-type: none"> 1. Der Nutzer wählt das unvollständige Tanzszenario aus. 2. Das unvollständige Tanzszenario wird gestartet. 3. NAO erklärt vor dem Tanz das Szenario und was der Nutzer Anschließend für Optionen hat. 4. NAO startet den Tanz. 5. Das Szenario ist beendet sobald NAO den unvollständigen Tanz vorgeführt hat. 6. NAO fordert dass der Nutzer die fehlenden Tanzschritte implementiert. 7. Der Nutzer soll anhand der Entwicklungsumgebung Choreographie die fehlenden Tanzschritte implementieren 8. Die fehlenden Tanzschritte sollen vom Nutzer anhand des Animationsmodus in Keyframes aufgenommen werden. 9. Der Nutzer muss die Keyframes in der Zeitleiste anordnen. 10. Sind die fehlenden Tanzschritte aufgenommen kann das neue Szenario auf dem NAO geladen werden und anschließend das Szenario erneut ausgewählt werden. 11. NAO tanzt die neue Choreographie vor. 12. Der NAO begeben sich in die Initialstellung zurück sobald das neue Tanzszenario beendet ist.

Tabelle 3.7.: Anwendungsfallbeschreibung: "Interaktivesszenario auswählen".

3.7.4. Anforderungen

Ausgehend von den Anwendungsfällen in Abschnitt 3.2 werden die funktionalen, technischen und nicht-funktionalen Anforderungen für das interaktive Szenario „Tanzszenario erweitern“ identifiziert.

Funktionale Anforderungen

- S3-FA-1** Das interaktive Szenario „Tanzszenario erweitern“ ist im System mit einer eindeutigen Identifikationsnummer repräsentiert. Anhand der Identifikationsnummer „4“ kann der Nutzer das interaktive Szenario „Tanzszenario erweitern“ nach der Aufforderung hierzu vom NAO starten.
- S3-FA-2** Der Nutzer muss mit der Entwicklungsumgebung Choregraphie die fehlenden Tanzschritte aufnehmen.

Technische Anforderungen

- S4-TA-1** Dem Nutzer muss ein lauffähiges Computersystem mit der vorinstallierten Entwicklungsumgebung Choregraphie geboten werden.
- S4-TA-2** Die Version „Choregraphie 2.1.3.3“ sollte auf dem Computersystem installiert sein..
- S4-TA-3** Das Betriebssystem des Computersystems muss Windows, Mac oder Linux sein.

Nicht-funktionale Anforderungen

- S4-NFA-1** Dem Nutzer werden die Hauptschritte als Bildvorlagen bereitgestellt.

Auf Basis der vorgestellten Szenarien in der Systemvision wurde in diesem Kapitel die Anforderungen spezifiziert und analysiert.

4. Konzept

Auf der Grundlage der Analyse in Kapitel 3 beschreibt dieses Kapitel die notwendigen Konzepte zur Realisierung der Präsentation des NAO. Um die Konzepte darzustellen, muss zuerst eine Evaluierung zur Modulprogrammierung stattfinden. Diese Evaluierung hilft bei der Auswahl der Programmiersprache für die spätere Implementierung der einzelnen Szenarien. Danach folgt im Abschnitt 4.2 ein Überblick über das System. Szenarienübergreifend folgt im Anschluss ein technischer Abschnitt mit den identifizierten Hardwarekomponenten. Zudem können die NAOqi-Module der betroffenen Hardware im Grundlagenkapitel 2 aus der Tabelle 2.2 erörtert werden. Darauf aufbauend werden die einzelnen Konzepte der vier Szenarien beschrieben.

4.1. Evaluation zur Modulprogrammierung

Um die gewünschten Module (vgl. Kapitel 2.3.1) konsequent und effektiv zu implementieren, stellt Aldebaran Robotics folgende zwei Wege zur Umsetzung vor:

- über die mitgelieferte Entwicklungsumgebung Choregraphe (graphische Programmierung)
- über die SDK der unterstützten Programmiersprachen (Quellcode implementieren)

Im Grundlagenabschnitt 2.3.2 ist ersichtlich, dass die Umsetzung der Präsentation ein passendes Framework benötigt, um den Kriterien der Evaluation gerecht zu werden. Anhand der Tabelle 4.1 werden diese Kriterien aufgelistet und am Ende des Kapitels bewertet. Anschließend kann für das Konzept ein geeignetes Framework ausgewählt werden.

Kriterium	Beschreibung
Modul lokal ausführbar	Der erzeugte Quellcode der Präsentation kann direkt auf dem NAO aufgespielt und autonom ausgeführt werden.
Module remote ausführbar	Das erzeugte Behavior liegt auf dem Entwicklungscomputer. Beim Abspielen wird das Behavior immer über das Netzwerk auf dem NAO ausgeführt.
Modul Ausführungsgeschwindigkeit	Die Ausführungsgeschwindigkeit ist fließend, egal ob lokal oder remote Ausführungen der Module.
Choreographie entwickeln	Bewegungsabläufen werden bei der Implementierung visuell unterstützt

Tabelle 4.1.: Kriterien für die Implementierung.

4.1.1. Programmierung anhand Choregraphie

Aldebaran Robotics bietet eine speziell angefertigte **IDE** namens „Choregraphie“ sowie eine hilfreiche Dokumentation für die Programmierung des NAO. Die **IDE** stellt ein Bindeglied zwischen dem NAOqi-Framework sowie dem NAO-Simulator dar und unterstützt die Entwicklung von Animationen und Behaviors (deutsch: Verhalten) durch eine graphische Schnittstelle. Das Testen der Module sollte bevorzugt mit dem Simulator durchgeführt werden, um die Hardware zu schützen. Da allerdings nicht alle Systembestandteile (z. B. Kamera) durch den Simulator abgedeckt werden, müssen bestimmte Funktionen direkt beim Roboter getestet werden.

Die **IDE** stellt vordefinierte Behaviors als Boxen dar. Durch eine Aneinanderreihung der Boxen in einem Flussdiagramm können komplexe Bewegungsabläufe zusammengestellt werden. Durch Schleifen und Bedingungen können Behaviors manipuliert werden. Außerdem erlaubt die **IDE** das Erweitern eines Behaviors durch das Hinzufügen von geschriebenen Python-Skripten. Im Anhang C ist eine Veranschaulichung der **IDE** mit einem kleinen Beispielprogramm beschrieben.

Bei einer remote Ausführung des Moduls auf dem NAO mittels Netzwerkübertragung entstehen hohe Übertragungszeiten, was sich negativ auf das Kriterium der Ausführungsgeschwindigkeit auswirkt. Die hohen Übertragungszeiten kommen durch die unterschiedliche Menge an Daten (z. B. Audiodateien) jedes erstelltes Behaviors zustande.

4.1.2. Programmierung anhand SDK

Da das NAOqi-Framework für verschiedene Programmiersprachen (Abb. 2.7) zu Verfügung steht, wird auf diese in den kommenden Abschnitt (4.1.2 näher eingegangen).

Modul Programmierung (Proxy)

Bei der Modul-Programmierung über Proxys werden bereits vorgefertigte lokale Methoden verwendet, um Behaviors für den NAO zu erstellen. Bei den bereits vorgefertigten lokalen Methoden handelt es sich um Methoden, die von NAOqi mitgeliefert beziehungsweise vom Entwickler zuvor definiert und in NAOqi integriert wurden. Die Nutzung dieser Methoden für das Erstellen von Behaviors wird über das Schlüsselkonzept realisiert, das im Grundlagenabschnitt 2.3.1 vorgestellt wurde. Zunächst muss ein Proxy-Objekt instanziiert werden, das genau ein Modul von NAOqi repräsentiert. Das Proxy-Objekt kennt nach der Instanziierung sämtliche Modulmethoden in ihrem Verhalten und ihren Eigenschaften. Bei Methodenaufrufen erweckt es den Anschein, sie lokal ausgeführt zu haben.

Mit der Modul-Programmierung über ein Proxy wird erzielt, dass ein schneller Zugriff auf die Methoden der Module des NAOqi stattfindet. Trotzdem sind diese Module nicht sehr flexibel, da sie sich nur auf die Ansteuerung des Roboters auswirken. Das Broker-zu-Proxy-Konzept wird anhand eines Beispiel-Python-Skripts und visuell im Anhang C beschrieben.

```
1 from naoqi import ALProxy
2 tts = ALProxy("ALTextToSpeech", "<IP_of_your_NAO>", 9559)
3 tts.say("Hello,_world!")
```

Listing 4.1: Quellcode "Hello World": Methoden aufruf anhand eines Proxy-Objekt

Die zweite Zeile des Quellcodes erstellt ein Objekt namens „tts“. Mit dem erstellten Objekt werden Aufrufe an NAOqi gesendet. Das ALProxy()-Objekt ermöglicht den Zugang zu den Methoden der Module in NAOqi. Das Modul im Skript ist das ALTextToSpeech, in der ersten Parameter-Anweisung. Somit wird ein Proxy-Objekt für das Ansprechen der Methode des Moduls ALTextToSpeech erstellt. Zur Vollendung eines Proxy-Objekts fehlt die Kennzeichnung eines NAO. Dies wird mit den restlichen Parametern IP und der Portnummer übergeben. In der dritten Zeile des Quellcodes erfolgt ein Aufruf der Methode say mit entsprechender Parameter-Übergabe. Das geschriebene Hello-World-Python-Skript 4.1 erlaubt dem NAO das Sprechen in seiner Umgebung.

Modul Programmierung (Broker)

Das Konzept „Broker zu Broker“ ermöglicht, eigenständige Module zu entwickeln, die das Verhalten des NAO in NAOqi nicht beinhaltet, und somit die Funktionalität des NAOqi zu erweitern. Dazu wird ein eigenständiger Broker benötigt, der als Client dient, um sich mit der übergeordneten Instanz „Main-Broker“ zu verbinden. Das Modul zur Erweiterung von NAOqi besteht aus einer Anzahl von Methoden, welche eine eigene Logik beinhalten. Über die Verbindung vom Broker zum Main-Broker wird die Öffentlichkeit des Moduls hergestellt. Somit können die kommenden Aufrufe an den Broker delegiert werden. Zudem entsteht eine Verknüpfung zu den bestehenden Modulen von NAOqi und den Modulen des Brokers. Es ermöglicht, dass die Module untereinander Variablen und Methoden anbieten.

Über den erstellten Broker wird eine zweite Laufzeitumgebung geschaffen, die unabhängig von NAOqi existiert. Beide Instanzen können unabhängig voneinander Kommunikation aufbauen, was die Stabilität des Gesamtsystems nicht gefährdet. Ein Absturz des erstellten Brokers kann durch fehlerhaftes Programmieren erfolgen. Das Konzept ermöglicht einen orts-unabhängigen Aufenthalt des Moduls. Ein Methodenaufruf des erstellten Moduls kann sowohl aus dem NAOqi des NAO oder auf dem NAOqi-Framework eines Entwicklungscomputers stattfinden. Im Anhang C wird das Broker-zu-Broker-Konzept dargestellt, wo sich das Modul im NAOqi-Framework des Entwicklungscomputers befindet. Ein beispielhaftes Python-Skript könnte folgendermaßen aussehen:

```
1 from naoqi import ALProxy
2 from naoqi import ALBroker
3 from naoqi import ALModule
4
5 NAO_IP = "nao.local"
6
7 class NewModule(ALModule):
8
9     def __init__(self, name):
10         ALModule.__init__(self, name)
11
12         # Ein Proxy Objekt vom Modul ALTextToSpeech erstellt
13         self.tts = ALProxy("ALTextToSpeech")
14
15         # Erweiterungs Methode.
```

```
16 # Ruft die say Methode des ALProxy auf.
17 def sayHello(self):
18 self.tts.say("Hello,_folks")
19
20 def main():
21 #pip = "Die IP des NAO"
22 #pport = "Die Portnummer des NAO"
23 parser.set_defaults(pip=NAO_IP, pport=9559)
24
25 (opts, args_) = parser.parse_args()
26 pip = opts.pip
27 pport = opts.pport
28
29 #Erstellung eines neuen Broker.
30 #Ueber den Broker koennen neue Module
31 #fuer das NAOqi konstruiert werden.
32 #Zudem koennen Anfragen zu internen Module statt finden.
33 myBroker = ALBroker("myBroker",
34 "0.0.0.0", # listen to anyone
35 0, # find a free port and use it
36 pip, # parent broker IP
37 pport) # parent broker port
```

Listing 4.2: Quellcode "Hello World": Broker zu Main-Broker Konzept.

Relevant sind die Zeilen 33 bis 37 im Quellcode. Hier findet die Erstellung eines neuen Brokers mit der dazugehörigen Verbindung zum Main-Broker statt. Mit dem Erstellen des Brokers können neue Module für NAOqi konstruiert werden. Um die neuen Logiken zu nutzen, kann das Proxy-Konzept verwendet werden (siehe Listing 4.1).

Modul Programmierung (Main-Broker)

Das Konzept für die Implementierung eigener Module mit der Nutzung des Main-Brokers ist ein Spezialfall und mit Vorsicht zu gebrauchen, denn für die Module wird keine zweite Laufzeitumgebung erschaffen. Der Broker wird direkt mit der übergeordneten Instanz Main-Broker verknüpft und dient ihm als alleinige Kontrolle. Alle Module befinden sich lokal auf dem NAOqi des NAO. Die neu erstellten Module sind als auf derselben Ebenenhöhe wie die vorgefertigten Module vom Hersteller befindlich anzusehen und anzusprechen. Dieses Konzept eignet sich gut für Module, die eine höhere Belastung an rechenintensive Aufgaben an die CPU verursachen, weiterhin besitzen sie dieselben Hardwarezugriffsrechte.

Erst nach intensiver Überlegung sollte über den Einsatz dieses Konzeptes entschieden werden. Bei einer falschen Implementierung oder im Falle eines Speicherfehlers führt dies zur sofortigen Beendigung des NAOqi-Prozesses. Außerdem kann eine Auslastung der CPU bei einem Methodenaufruf des neu erstellten Moduls zur selben Reaktion führen. Da es zur Beendigung der Ansteuerung des kinematischen Körpers kommt, wird der NAO umfallen. Dies kann wiederum zu einer Beschädigung des NAO führen.

4.1.3. Auswertung der Evaluation

Die Tabelle 4.1 wird nun kritisch im Hinblick auf die Entwicklungsmöglichkeiten, d. h. auf deren Eigenschaften und Umsetzung, hin bewertet. Die Bewertung ist aus den Abschnitten 4.1.1 und 4.1.2 entstanden. Der Autor hat hierbei Punkte von keinem Punkt (-) bis maximal zwei Punkte (++) vergeben.

Kriterien	Choregraphie	SDK
Modul lokal ausführbar	++	+
Module remote ausführbar	++	++
Ausführungsgeschwindigkeit	+	+
Choreographie entwickeln	++	-

Tabelle 4.2.: Auswertung der Evaluation anhand der ermittelten Kriterien.

Chregraphie Das Programm bietet ein schnelles Verständnis für die Modulprogrammierung des NAO und ist für den Einstieg bestens geeignet. Die Bereitschaft der vordefinierten Module vereinfacht den Aufbau des zu entwickelnden Moduls. Zudem ermöglichen die Modi des NAO und die Visualisierung bei der Programmierung schnell neue Bewegungsabläufe. Eine Erweiterung der erstellten Behaviors zu komplexen Programmabläufen durch Python-Skripte ist auch in Choregraphie möglich.

Software Development Kit (SDK) - Das Implementieren von Modulen über Entwicklungssprachen kann bevorzugt werden, wenn Module erstellt werden, die viel Rechenzeit brauchen, hardwarenah sein sollen oder bezüglich der Ausführungsgeschwindigkeit punkten müssen. Hierzu wurden in Abschnitt 4.1.2 weitere Konzepte vorgestellt, die bei der Auswahl helfen, vor allem muss eine Entscheidung zwischen einem lokalen oder Remote-Modul getroffen werden.

Die ganze Präsentation wird in Choregraphie entwickelt. Abgesehen davon, dass die Tabelle 4.2 bei der Bewertung für Choregraphie spricht, wurde während der Einarbeitung in die Modulpro-

grammierung ersichtlich, dass die IDE für ein erfolgreiches Erreichen des Ziels gut geeignet ist. Die Präsentation an sich nutzt ständig die intern vorgefertigten Module des NAOqi. Es ist nicht nötig, neu definierte Module, die ein neues Verhalten für den NAO aufweisen, zu entwickeln. Folglich muss kein weiterer Broker konstruiert werden und das Aufrufen der Module kann über das Proxy-Konzept (vgl. Kapitel 4.1.2) erfolgen. In Kombination mit Python-Skripten werden die Logiken der Szenarien entwickelt.

4.2. Systemarchitektur

Das System (Abb. 4.1) besteht aus drei Komponenten, welche von einem Akteur bedient werden. Der NAO stellt hierbei die erste Komponente dar. Die zweite Komponente stellt der Rechner dar, auf welchem die IDE läuft. Die dritte Komponente besteht aus dem Netzwerk, welches den ersten beiden Komponenten miteinander zu kommunizieren ermöglicht. Der Akteur leitet bei allen Szenarien den Ablauf des Systems mittels einer Interaktion mit dem NAO ein. Ausschließlich beim Szenario 4 ist nach der Interaktion mit dem NAO der Nutzer gefordert, Eingaben am Rechner vorzunehmen. Da beim Szenario 4 weitere Eingaben erforderlich sind, wie in Kapitel 3.7 beschrieben, muss der Nutzer nach der Interaktion mit dem NAO weitere Eingaben am Rechner vornehmen. Die in den Rechner eingegebenen Daten werden über das Netzwerk in den NAO integriert. Daher stellen der NAO und der Rechner die Schnittstellen des Systems für den Nutzer dar.

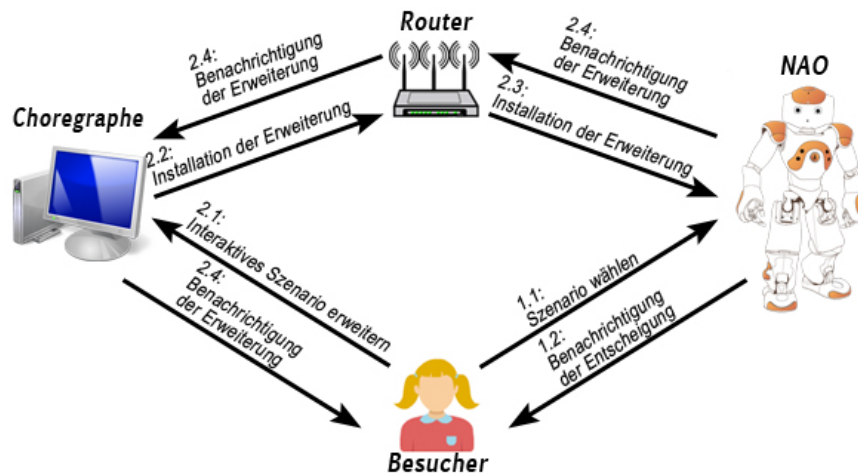


Abbildung 4.1.: Übersicht der Systemarchitektur.

4.3. Technischer Ansatz

Hardwarekomponente

Für die Realisierung aller vier Szenarien sind verschiedene Hardwarekomponenten erforderlich. Die Tabelle 4.3 listet die Hardwarekomponenten auf, die bei der Nutzung aller vier Szenarien benötigt werden. Zu jeder Hardwarekomponente wird der Nutzen erläutert, den diese bei der Anwendung erfüllt.

Identifikation	Hardware	Nutzen
/HW01	Lautsprecher	Begrüßung, Aufforderung, Moderation
/HW02	Mikrofone	Erfassung der Szenarioauswahl und Antworten
/HW03	Kamera	Gesicht-, Gesten- und Symbolerkennung

Tabelle 4.3.: Übersicht der genutzten Hardwarekomponenten.

Für die verbale Kommunikation werden die Komponenten (/HW01) und (/HW02) des Soundsystems verwendet. Die Komponente (/HW01) verarbeitet eingehende Soundsignale, wie die Antworten des Nutzers. Die Komponente (/HW02) gibt die Antworten des NAO akustisch wieder. Zur Gesichts-, Gesten- und Symbolerkennung verwendet der NAO die Kamera der Komponente (/HW03).

NAOqi Modul

Im Grundlagenkapitel 2.3.1 wurden sämtliche Module, die NAOqi beinhaltet, anhand der Tabelle 2.2 aufgelistet. In der folgenden Tabelle 4.4 werden die Komponenten farblich markiert um sie den jeweiligen Zielszenarien zuzuweisen.

Anhand von Stichpunkten werden die Module verdeutlicht.

ALBasicAwareness Modul, um Augenkontakt zum Nutzer aufzubauen und aufrechtzuerhalten. ALFaceDetection: Modul zur Erkennung von Gesichtern. Damit können Gesichter gespeichert und wiedererkannt werden.

ALMemory Zentrale Speicherstelle im NAO, nutzbar zum Speichern oder Auslesen von angelegten Daten sowie eventbasierte Kommunikation zwischen Modulen des NAOqi, um z. B. eine Begrüßung zu starten, sobald der NAO ein Gesicht wiedererkennt.

ALModule Der Entwickler ist zur Nutzung der Basisklasse zum Erstellen eigener Module verpflichtet, welche NAOqi nicht standardmäßig beinhaltet. Dies hilft bei der Bereitstellung und Bekanntmachung des entwickelten Moduls.

4. Konzept

Core	Vision	Audio
ALMemory	ALPhotoCapture	ALAudioDevice
ALModule	ALVideoDevice	ALAudioPlayer
	ALVisionRecognition	ALSpeechRecognition
Motion		ALTextToSpeech
ALMotion		
		People Perception
DCM		ALBasicAwareness
DCM		ALFaceDetection

Tabelle 4.4.: Übersicht der gestellten Module im NAOqi API. Die blau Markierten Module sind szenarioübergreifende Module. Die grün Markierten Module werden ausschließlich im Szenario 1 verwendet. Die gelb Markierten Module werden im Szenario 2 genutzt.

ALSpeechRecognition Spracherkennungssystem des NAO, damit können Interaktionen des Nutzers entgegengenommen werden.

ALTextToSpeech Sprachsynthese-Modul, damit kann der NAO in seiner Umgebung kommunizieren. Es eignet sich hervorragend, um die Szenarien oder Moderationen vorzustellen.

DCM Softwaremodule des NAOqi. Bis auf das Soundsystem und die Mikrophone hat das DCM Zugriff auf jede weitere Hardware. Jedes gesendete Kommando wird vom DCM zum jeweiligen Aktor gesendet oder die Werte des Aktor werden gelesen.

4.4. Konzeption - Szenario 1

Die Systemvision im Abschnitt 3.1 enthält eine Beschreibung dessen, eine interaktive Präsentation zu entwickeln, die bei der Nutzung eine Atmosphäre aufbaut, die bei Nutzung positive Gefühle bewirkt. Eine Kombination aus angenehmer Musik und rhythmischen Körperbewegungen kann eine positive Atmosphäre aufbauen. Des Weiteren können mit Musik schöne sowie unangenehme Situationen assoziiert werden: Ein angenehmes Musikstück kann Freude bereiten, während ein unangenehmes Musikstück zu negativen Gefühlen führen kann. Im kommenden Abschnitt wird das Konzept des Szenarios 1 erläutert, das für die spätere Realisierung des Szenarios in Kapitel 5.1 notwendig ist.

4.4.1. Bewegung

Aus dem Analysekapitel 3.4.1 resultieren Fragen zur Erörterung der Beweglichkeit des NAO, worauf nachfolgend kurz eingegangen wird.

Zeitleiste

Der Mensch verdankt der Muskulatur eine flüssige Bewegung des Körpers. Die Steuerung des Roboters wird von Motoren und von DoF hingegen in einer robotertypischen Art und Weise ausgeübt. Diese Bewegung entsteht durch die Impulsvergabe und besteht aus zwei Impulsen: Der erste Impuls regt an, den DoF zur maximalen Geschwindigkeit in die Position X zu bewegen. Der zweite Impuls regt zum sofortigen Stoppen des DoF an, sobald von X die Position erreicht worden ist. Hieraus resultiert folgende Frage:

Wie schafft es der NAO, die unterschiedlichen Geschwindigkeiten eines Musiktitels durch rhythmische Körperbewegungen taktgemäß zu begleiten?

Der NAO besitzt die Möglichkeit, fließendere Körperbewegungen auszuüben: mittels Choregraphie und der Timeline. Die Timeline erlaubt, Bewegungsebenen für den NAO zu kreieren. Solche Bewegungen werden in Keyframes festgehalten. Der Keyframe ist die Einheit der Timeline. Ein Keyframe hält die Winkel der einzelnen DoF des Körpers fest. Somit repräsentiert jedes einzelne Keyframe eine neue Position des NAO. In der Timeline findet zwischen allen Keyframes eine Interpolationsberechnung der gespeicherten Werte statt. Die Interpolationen bezwecken Übergänge zwischen Bewegungen, damit eine robotertypische Bewegung vermieden wird. Durch die Nutzung der Timeline können sanfte und fließende Bewegungen für den NAO kreiert werden.

Balance

Jeder Körper besitzt einen Massenmittelpunkt. Dadurch wird die Stabilität einer Position des Körpers erzielt. Befindet sich der Schwerpunkt in der Basis des Körpers, bedeutet dies eine Stabilität für starre Körper. Die für die Realisierung einer Choreographie zu klärende Frage lautet somit wie folgt:

Wie kann eine komplexe Tanzposition eingenommen werden, ohne den NAO aus dem Gleichgewicht zu bringen?

Auch der Körper des NAO besitzt einen Massenmittelpunkt, der sich ungefähr im Torso lokalisieren lässt. Steht der NAO auf beiden Beinen in einer aufrechten Position, befindet sich

die Basis zwischen beiden Füßen. Steht der NAO auf einem Bein, liegt die Basis nur auf dem entsprechenden Fuß. Das Verschieben des Schwerpunktes im Körper wird dann erzielt, wenn eine Körperhaltung gefunden worden ist und sich nur die Position der Arme kontinuierlich interpoliert. Komplexe Körperpositionen können erschaffen werden, wenn man dabei den Massenmittelpunkt des NAO berücksichtigt.

4.4.2. Technischer Ansatz

Hardwarekomponente

Die folgende Tabelle listet speziell die Hardwarekomponenten auf, welche für eine erfolgreiche Realisierung zusätzlich zu den in der Tabelle 4.3 genannten benötigt werden.

Identifikation	Hardware	Fähigkeit
S1/HW01	Motoren und Gelenke	Bewegung des Körper

Tabelle 4.5.: Übersicht der speziellen Hardwarekomponenten im Szenarien 1.

Durch die Nutzung der Motoren und Gelenke (S1/HW01) kann der NAO seinen Körper in Bewegung bringen. Die Hardwarenutzung muss ins Szenario aufgenommen werden, damit der NAO tanzen kann.

NAOqi Modul

Für eine Realisierung des Szenarios 1 wurden weitere Hardwarekomponenten identifiziert. Aldebaran Robotics hat zur Ansteuerung dessen spezielle Module bereitgestellt, welche nachfolgend anhand von Stichpunkten verdeutlicht werden.

ALAudioDevice Das Modul regelt die Ein- und Ausgabe von akustischen Signalen. Um von anderen Modulen verwendet zu werden, stellt es öffentliche Funktionen zur Verfügung.

ALMotion Modul, das sämtliche Methoden der Bewegung des NAO abdeckt. Die Methoden sind in vier Hauptgruppen unterteilt. Beispielsweise animiert die Methode `moveTo()` den NAO zum Gehen.

4.4.3. Interaktionsdiagramm

Im kommenden Abschnitt wird mithilfe des Sequenzdiagramms UML 2.5 [11] die Interaktion des Szenarios 1 zwischen dem Nutzer und dem NAO modelliert. Das Verhalten zur Laufzeit wird durch die Gewinnung des Sequenzdiagramms erzielt. Das Sequenzdiagramm konzentriert

4. Konzept

sich speziell auf den Ablauf der Interaktion (siehe Abbildung 4.2). An der Interaktion sind zwei Lebenslinien von Teilnehmern, die an der Interaktion teilnehmen:

- Ein Tänzer, dessen Tanz vom NAO übernommen wird
- Ein Nutzer, der die Schlusschoreographie auswählt

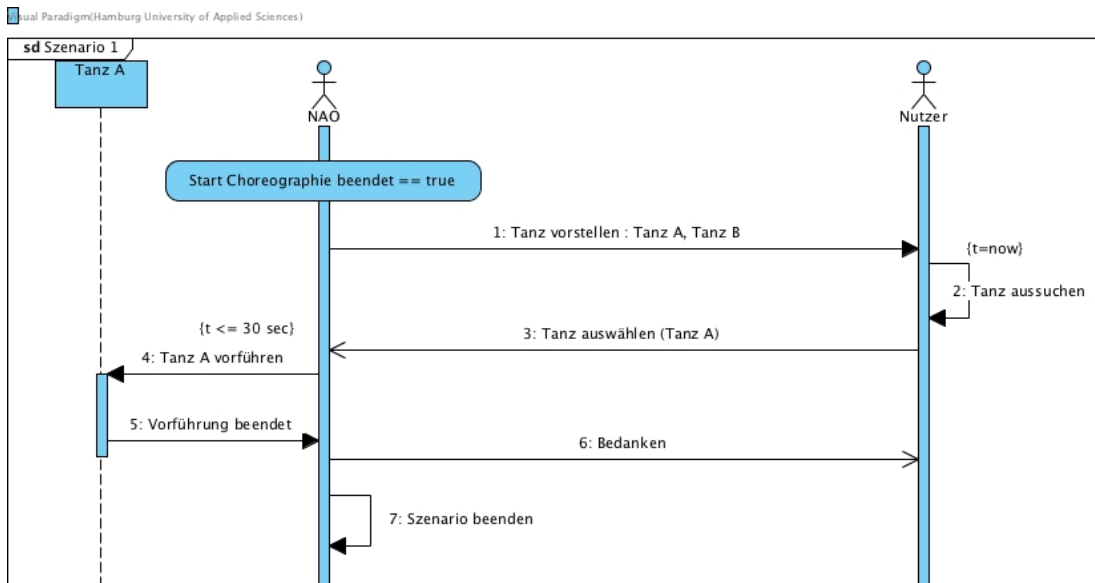


Abbildung 4.2.: Sequenzdiagramm: Auswahl der Schlusschoreographie.

Der Ausgangspunkt ist, dass der Nutzer die Auswahl des Szenarios 1 getroffen hat. Sobald der NAO die Endposition der Standard-Choreographie erreicht hat, stellt er dem Nutzer eine Auswahl von Tänzen vor (1). Die Auswahlmöglichkeit besteht aus zwei verschiedenen Tänzen. Das Vorstellen der Tänze erfolgt synchron (ausgefüllte Pfeilspitze), d. h., der NAO wartet auf die Auswahl des Nutzers. Der Nutzer trifft eine Entscheidung, mit welcher der NAO die Choreographie fortzuführen hat (2). Die Entscheidung des Nutzers wird dem NAO verbal mitgeteilt (3), es ist eine asynchrone Nachricht. Registriert der NAO die Nachricht vor dem Ablaufenden des hierfür vorgegebenen Zeitfensters von 30 Sekunden, kann das System den Wunsch des Nutzers zuordnen und die Vorführung mit der entsprechenden Choreographie wiederaufnehmen (4, 5). Nach der Beendigung der Schlusschoreographie bedankt sich der NAO beim Nutzer (6). Der NAO begibt sich zurück in den Idle-Zustand, wodurch sich der Nutzer erneut registrieren lassen kann, womit das Szenario endet (7).

4.5. Konzeption - Szenario 2

4.5.1. Symbol

Für das kommende Spielszenario müssen Gestiken gespielt und erkannt werden. Der NAO muss die Gestiken so formen, dass der Nutzer sie erkennen kann. Das Problem ist dabei die Hand des NAO: Dass die Hand des NAO aus drei Fingern besteht und er diese nur simultan bewegen kann, beschränkt den NAO beim Entwerfen der Gestiken. Die zu implementierende Spiellogik besteht aus insgesamt drei Symbolen: Stein, Schere und Papier. All diese Symbole kann der NAO für seinen Kontrahenten deutlich darstellen.

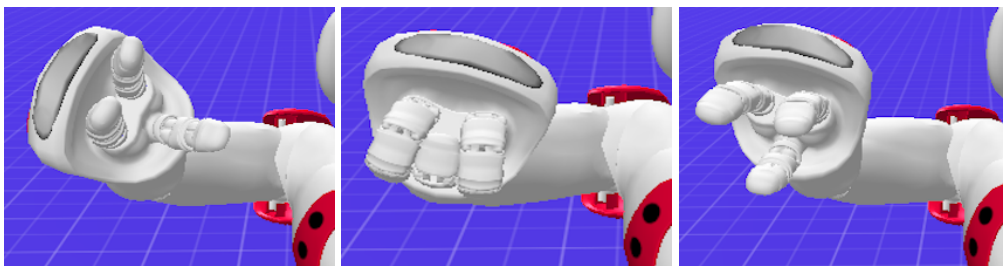


Abbildung 4.3.: Schere (Links), Stein (Mittig), Papier (Rechts).

4.5.2. Strategie

Bei diesem Spiel neigt jeder Spieler dazu, den strategisch besten Zug zu wählen, um aus dem Spiel als Gewinner hervorzugehen. Das Spiel „Schere, Stein, Papier“ ist ein Nullsummenspiel, der Verlust des einen Spielers ist folglich der Gewinn des Kontrahenten. In einem Spiel sind die komplexen Algorithmen eines Computersystems für den nächsten Zug verantwortlich. Wessler [10] teilt die Strategien in reine, Langzeit- und gemischte Strategie ein (siehe Abschnitt 3.5.1). Auf Basis der Strategieaufteilung von Wessler beleuchtet der folgende Abschnitt die konzeptionellen Strategien des NAO während eines Spielverlaufs.

Random Bei dieser Strategie ist das Verhalten des NAO zufällig und somit für das den Mitspieler unberechenbar. Das Erscheinungsbild des NAO wirkt mit dieser Strategie menschlicher und dies kann beim Spielen zu einer positiven Spieleinstellung führen. Der zu realisierende Algorithmus hat die Eigenschaft, dass der NAO bei jeder Spielrunde ein zufälliges Symbol wählt. Das Aufbauen einer reinen Strategie ist somit ausgeschlossen, der Nutzer weiß nie zu 100 %, welche Strategie der NAO zufällig wählt.

Déjà-vu Beim Spielen mit der Déjà-vu-Strategie verfällt der NAO in ein Musterverhalten, das sich ständig wiederholt. Auf das Spiel bezogen ist der Algorithmus wie folgt zu beschreiben: Bei der ersten Spielrunde wird zufällig eines der drei zu spielenden Symbole (Schere, Stein, Papier) gewählt. In den nachfolgenden Spielrunden wird kein weiteres Symbol generiert. Der NAO spielt von nun an immer das gespeicherte Symbol, bis aus der Partie ein klarer Sieger hervorgeht. Dadurch, dass der NAO ein bestimmtes Muster vorweist, ist er berechenbar. Dies eignet sich somit hervorragend für den Nutzer, um eine Langzeit-Strategie gegenüber dem NAO aufzubauen. Das Verhalten des NAO weist bei dieser Strategie die Verlässlichkeit und Qualität eines Industrieroboters aus. Industrieroboter sind so programmiert und konzeptioniert, dass sie bei ihrer Tätigkeit immer denselben Bewegungsablauf ausüben.

Mogel Die Motivation für eine Strategie, bei welcher der NAO mogelt, damit er gewinnt, ist durch eine Roboter-Hand namens „Janken Robot“ entstanden. Diese Roboter-Hand besiegt ihre Gegner beim Spiel „Schere, Stein, Papier“ immer zu null. Bei der Mogelstrategie wird der NAO zu 100% gegen den Nutzer gewinnen. Der NAO wird zugleich ständig mogeln, denn er wählt ein Symbol immer erst dann, wenn das Symbol des Kontrahenten bereits vom System identifiziert wurde. Anhand der Identifizierung kann durch die Auswertungsmatrix 3.4 dasjenige Symbol gewählt werden, welches dem seines Kontrahenten überlegen ist.

4.5.3. Technischer Ansatz

Hardwarekomponente

Das Szenario 2 benötigt zur Auswertung der gespielten Symbole des Nutzers eine visuelle Möglichkeit. Anhand der Tabelle 4.6 werden szenariospezifische Hardwarekomponenten aufgelistet und im Anschluss erörtert.

Identifikation	Hardware	Fähigkeit
S2/HW01	Kamera	Visuelle Wahrnehmung

Tabelle 4.6.: Übersicht der speziellen Hardwarekomponenten im Szenarien 2.

Für eine visuelle Wahrnehmung der vom Nutzer gespielten Symbole dienen als Hardware die integrierten Kameras (S2/HW01). Für die Nutzung beim Szenario 2 eignet sich die obere Kamera des NAO. Anhand dieser Kamera erhält der NAO einen Blick nach vorn. Dies ermöglicht ihm,

einen Schnappschuss der Symbolbilder des Nutzers zu schießen, der später in die Auswertung des Spieles eingeht.

NAOqi

In Stichpunkten werden nachfolgend die NAOqi-Module erörtert, die für das Kamerasystem nötig sind, um die Kamera zu bedienen, sowie zur Aufnahme eines Bildes relevant sind.

ALPhotoCapture Modul, um Bilder mit der Kamera aufzunehmen und auf der Festplatte zu speichern.

ALVideoDevice Modul, das den Zugriff auf das Kamerasystem des NAO ermöglicht. Es stellt die Bilder der Videoaufnahme für weitere Module bereit wie **ALVisionRecognition**.

ALVisionRecognition Das Bilderkennungsmodul im NAOqi System. Der NAO versucht, Bilder, Objekte, Seiten und sogar Standorte wiederzuerkennen, welche ihm vorher gelehrt werden. Dies eignet sich für die spätere Realisierung (vgl. Kapitel 5.2), um die gespielten Symbole des NAO wiederzuerkennen.

4.5.4. Fachliches Datenmodell

Aus den Anforderungen und der Analyse wird entsprechend im Detail das Szenario 2 „Schere, Stein, Papier“ in Form eines **UML 2.5-Klassendiagramms** beschrieben. Die Abbildung 4.4 zeigt das Klassendiagramm mit seinen Eigenschaften und den Assoziationen zueinander.

Nutzer - Klasse Der Nutzer besitzt einen Namen, mit dem er angesprochen wird. Das Starten eines Spieles ist der Rolle des Nutzers gestattet. Der Nutzer kann vor dem Beginn des Spieles die Strategie einstellen, mit welcher der NAO spielt.

Runde - Klasse Eine Runde gehört zu genau einem Spiel. Es gibt maximal fünf Spielrunden, bis ein Gewinner ermittelt wird. Eine Spielrunde erreicht das Ende beim erfolgreichen Feststellen des Gewinners.

Symbol - Enumeration Die zu spielende Statistik erfordert eine Menge an Symbolen, die dem Abschnitt 4.5.1 zu entnehmen sind. Es repräsentiert die zu spielende Gestik, die der NAO je nach Vorauswahl bei der Statistik annimmt.

Einstellung-Klasse Die Schwierigkeitseinstellung gehört zu genau einem Spiel. Es hält die Algorithmen der Strategien aus dem Abschnitt 4.5.2 fest.

4. Konzept

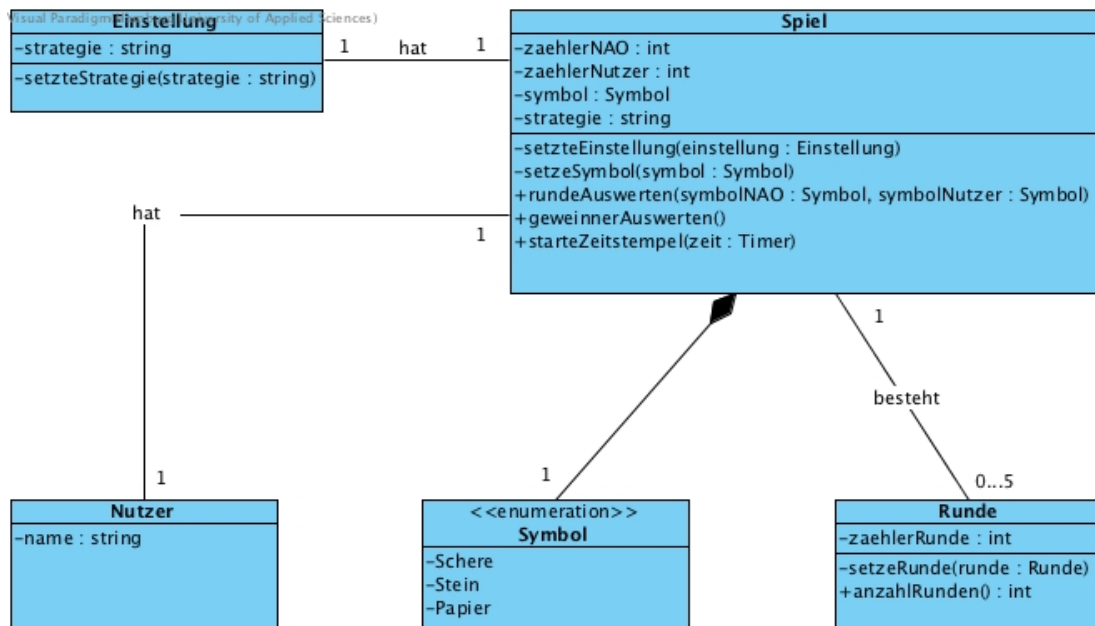


Abbildung 4.4.: Klassendiagramm des Szenario 2 "Schere, Stein, Papier".

Spiel-Klasse Die zentrale Klasse des Systems ist die Spiel-Klasse. Sie besteht aus genau einem Nutzer, der die Rolle des Spielers annimmt. Ein Spiel umfasst maximal fünf Spielrunden, bis ein Gewinner ermittelt wird. Symbole werden dem Spiel zugeordnet, damit das Spielen einer Geste visuell dem Nutzer des NAO vorgestellt wird. Die Spiel-Klasse hält die Logik des Spieles fest. Es wertet zudem jede Spielrunde aus und ermittelt den Gewinner anhand der Auswertungsmatrix (siehe Tabelle 3.4).

4.5.5. Architektur

Im Folgenden wird die Architektur des Szenarios 2 aus dem fachlichen Datenmodell (siehe Abbildung 4.4) durch ein Komponentendiagramm dargestellt. Eine Komponente besteht aus sechs Merkmalen, so beschreibt es Siedersleben [12]. Die folgenden sechs Merkmale leiten das Komponentendiagramm mit seinen Beziehungen und den öffentlichen Schnittstellen aus dem fachlichen Datenmodell her:

1. Eine Komponente exportiert eine oder mehr Schnittstellen. Jede Komponente, welche die Schnittstelle S exportiert, ist zu einer Implementierung von S verpflichtet.
2. Eine Komponente importiert eine oder mehr Schnittstellen. Durch den Import einer Schnittstelle hat die Komponente Zugriff auf die Methoden der importierten Schnittstelle.

4. Konzept

Der Importeur ist zur Bereitstellung der Schnittstellen verpflichtet, um sie lauffähig zu implementieren.

3. Komponenten haben die Eigenschaft, ihre Implementierung nach außen geheim zu halten. Dies ermöglicht das Ersetzen von Komponenten durch eine andere Komponente, die dieselben Schnittstellen exportiert.
4. Die Komponente kennt die Umgebung nicht, in der sie sich aufhält, dadurch eignet sich Erstere als Einheit zur Wiederverwendung.
5. Eine Komponente kann aus weiteren Komponenten bestehen. Eine Zusammensetzung aus Komponenten führt zu neuen Komponenten.
6. Die Komponente ist für den Entwurf, die Implementierung und die dazugehörige Planung eine der wesentlichen Einheiten.

Die oben beschriebenen Merkmale von Siedersleben [12] tragen zum Verständnis der Komponente bei. Mithilfe dieser Merkmale werden die Komponenten mit ihren öffentlichen Schnittstellen aus dem Klassendiagramm entworfen.

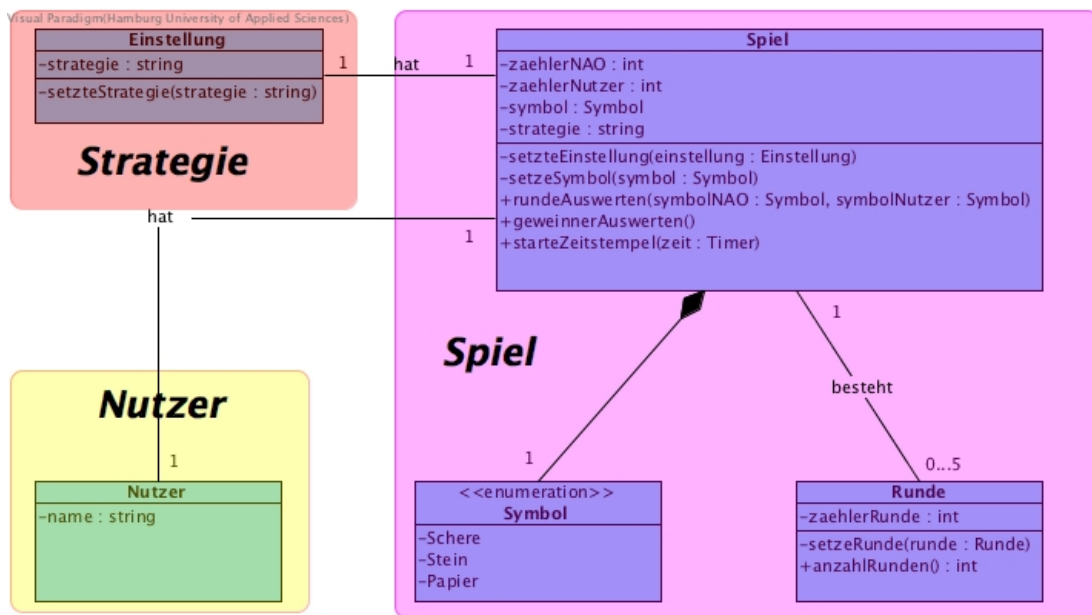


Abbildung 4.5.: Abbildung des Klassendiagramm "Schere, Stein, Papier" auf Komponenten.

Durch den Komponentenschnitt sind folgende drei Komponenten mit ihren Beziehungen und öffentlichen Schnittstellen entstanden:

Nutzer-Komponente: verwaltet den Besucher.

Spiel-Komponente: verwaltet das Spiel und hält die darin enthaltene Logik sowie den Spielablauf fest.

Strategien-Komponente: verwaltet die zu spielenden Algorithmen der Strategien.

Das Komponentendiagramm (siehe Abbildung 4.6) dient der Strukturierung des Systems und der Abhängigkeiten zwischen den einzelnen Komponenten. Die Schnittstellen einer Komponente zeigen die Außenansicht der jeweiligen Komponente, die nach außen öffentlich sichtbar sind.

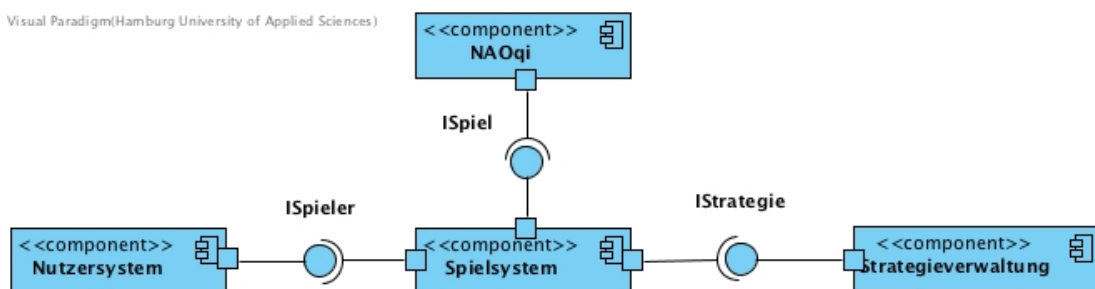


Abbildung 4.6.: Komponentendiagramm: "Schere, Stein, Papier".

4.5.6. Interaktionsdiagramm

Das Szenario 2 wurde von einem registrierten Nutzer aus vier möglichen Szenarien ausgewählt (vgl. Kapitel 3.5.4). Dem Nutzer stehen Einstellungsmöglichkeiten zur Verfügung, die den Schwierigkeitsgrad beim NAO ändern. Die Auswahl besteht aus drei Strategien (siehe Abschnitt 4.5.2). Die folgenden Sequenzdiagramme halten die Interaktion sowie die unterschiedlichen Strategien in einem Spielablauf fest. Hierfür wird die Random-Strategie als Beispiel gewählt. Im Anhang D sind die Alternativstrategien dargestellt und beschrieben. An der Interaktion sind zwei Lebenslinien von Teilnehmern, die an der Interaktion teilnehmen, beteiligt:

- Spieler A, die Position wird vom NAO eingenommen
- Spieler B, die Position wird vom Nutzer eingenommen

Random-Strategie

Die Ausgangssituation im Sequenzdiagramm ist, dass der Nutzer das Szenario 2 anhand der Identifikationsnummer vom NAO gewählt hat. Bevor der Spielverlauf beginnt, wartet das

4. Konzept

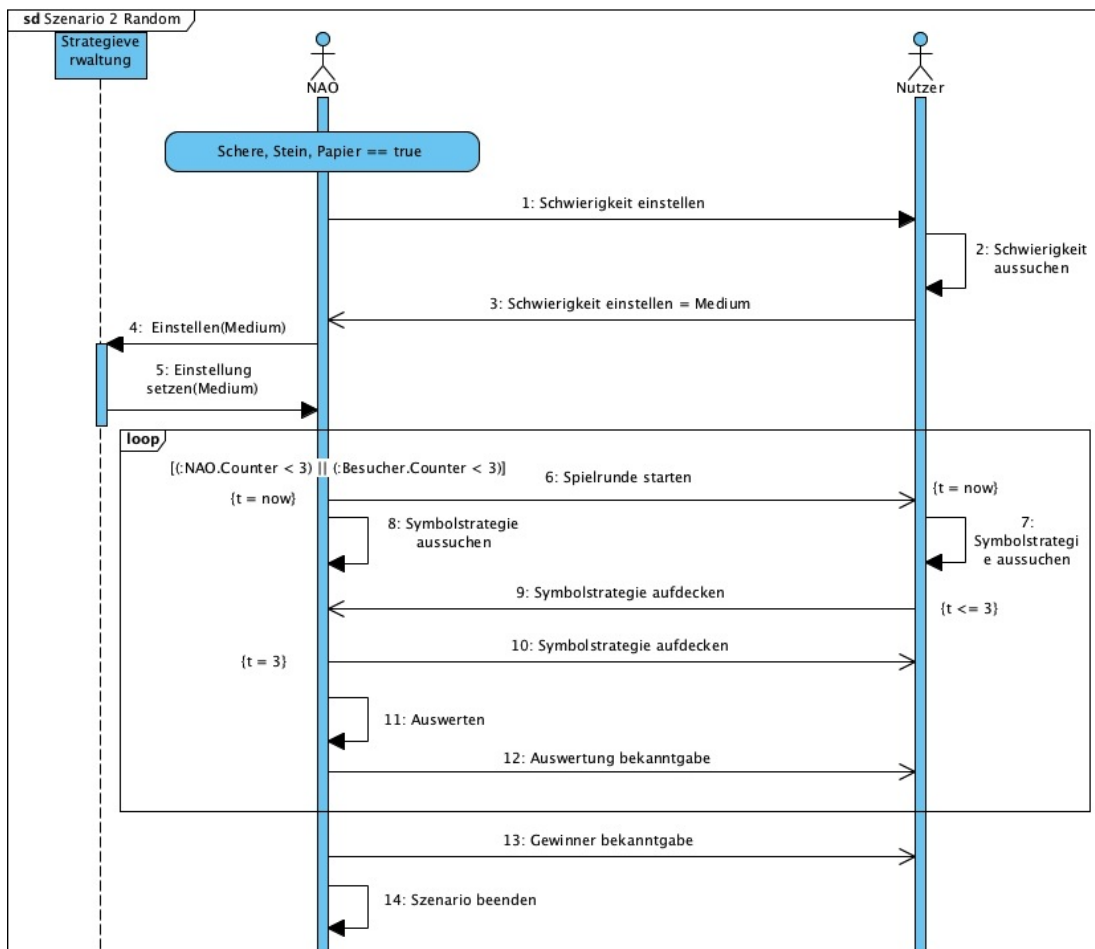


Abbildung 4.7.: Sequenzdiagramm: "Random-Strategie".

System auf eine Eingabe des Nutzers. Die Eingabe bestimmt den Schwierigkeitsgrad des NAO für den Spielverlauf (1). Der Nutzer entscheidet sich für einen Spielverlauf, bei dem der NAO mit einem mittleren Schwierigkeitsgrad antritt (2). Die getroffene Entscheidung wird dem NAO verbal mitgeteilt (3). Das System setzt für den NAO daraufhin die Random-Strategie ein (4, 5). Nach erfolgreichem Festlegen des Schwierigkeitsgrads beginnt das Spiel (6). Unabhängig voneinander wählen beide Spieler ein Symbol (7, 8). Sobald der Zeitpunkt des Aufdeckens da ist, zeigen beide die zu spielenden Symbole (9, 10).

Nach erfolgreicher Identifizierung des Symbols des Nutzers findet die Auswertung statt. Anhand der Auswertungsmatrix (siehe Tabelle 3.4) wird der Gewinner der Spielrunde bestimmt (11). Der NAO teilt dem Nutzer die Auswertung der Spielrunde mit (12). Sobald ein Spieler

drei Spielrunden gewonnen hat, findet das Spiel ein Ende und der NAO kann den Gewinner ernennen (13). Andernfalls wird eine neue Spielrunde aufgenommen (6).

4.6. Konzeption - Szenario 3

4.6.1. Fachliches Datenmodell

Für das Szenario 3 „Quiz-Show“ wird entsprechend den Anforderungen und der Analyse das Klassendiagramm beschrieben. Die Abbildung 4.8 zeigt das Klassendiagramm mit seinen Eigenschaften und den Assoziationen zueinander.

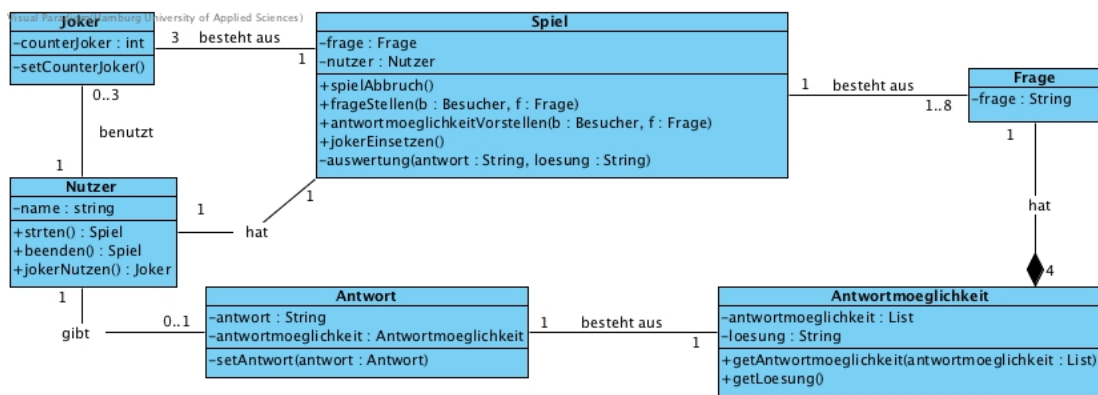


Abbildung 4.8.: Klassendiagramm des Szenario 3 „Quiz-Show“.

Nutzer-Klasse: Die Nutzer besitzen einen Namen, mit dem sie im Spielverlauf angesprochen werden. Das Starten eines Spielverlaufes ist der Rolle des Nutzers gestattet. Ein Nutzer kann auf die vom NAO gestellte Frage antworten und, wenn es zu einem Engpass kommt, einen Joker zur Hilfe einsetzen.

Frage-Klasse: Diese hält die Fragen fest, welche der NAO dem Nutzer während des Spieles stellt.

Antwortmöglichkeit-Klasse: Jede Frage-Klasse besitzt vier Antwortmöglichkeiten, die beim Stellen der Frage vom NAO mit vorgestellt werden. Sie hält separat die Lösung der Frage fest und dient zur Überprüfung der Antwort des Nutzers, die in der Antwort-Klasse gespeichert wird.

Spiel-Klasse: Die zentrale Klasse des Systems ist die Spiel-Klasse. Sie besteht aus genau einem Nutzer, der die Rolle als Spieler einnimmt. Ein Spiel benötigt maximal acht gestellte

4. Konzept

Fragen, bis der Nutzer gewinnt. Die drei zu Verfügung gestellten Joker werden beim Starten eines Spielverlaufs zurückgesetzt. Die Spiel-Klasse hält die Logik des Spieles fest. Es wertet zudem jede Frage aus und ermittelt den Gewinner.

4.6.2. Architekt

Durch das Komponentendiagramm wird die Architektur des Szenarios 3 beschrieben. Mithilfe der sechs Merkmale von Siedersleben [12] und durch die Analyse der Abhängigkeiten einzelner Klassen voneinander wird anhand des fachlichen Datenmodells ein Komponentenschnitt vorgestellt.

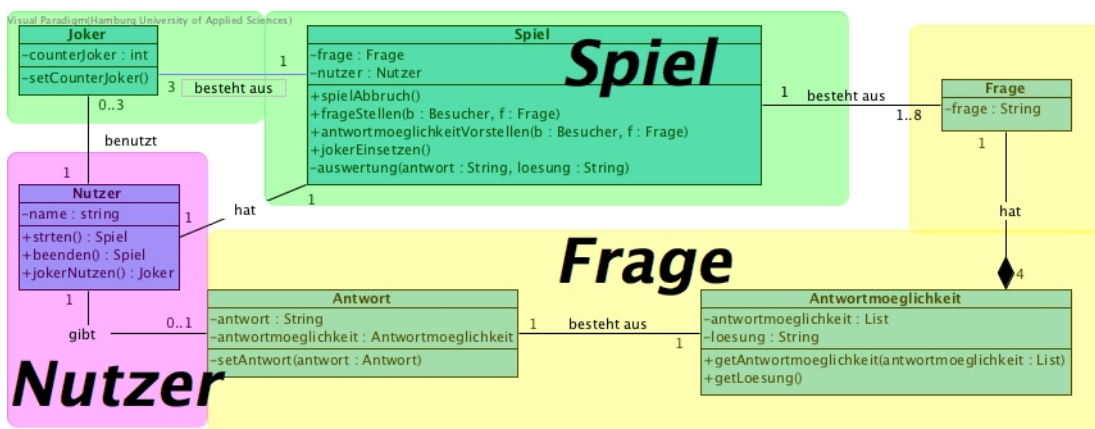


Abbildung 4.9.: Abbildung des Klassendiagramm "Quiz-Show" auf Komponenten.

Aus dem Komponentenschnitt des fachlichen Datenmodells sind drei Komponenten mit ihren Beziehungen und öffentlichen Schnittstellen entstanden.

Nutzer-Komponente: verwaltet den Nutzer.

Spiel-Komponente: verwaltet das Spiel und hält die darin enthaltene Logik sowie den Spielablauf fest.

Frage-Komponente: verwaltet die Fragen mit den dazugehörigen Antwortmöglichkeiten und Antworten des Nutzers.

Das Komponentendiagramm (siehe Abbildung 4.10) dient der Strukturierung des Systems und der Abhängigkeiten zwischen den einzelnen Komponenten. Es zeigt deren bereitgestellte und verwendete Schnittstellen.

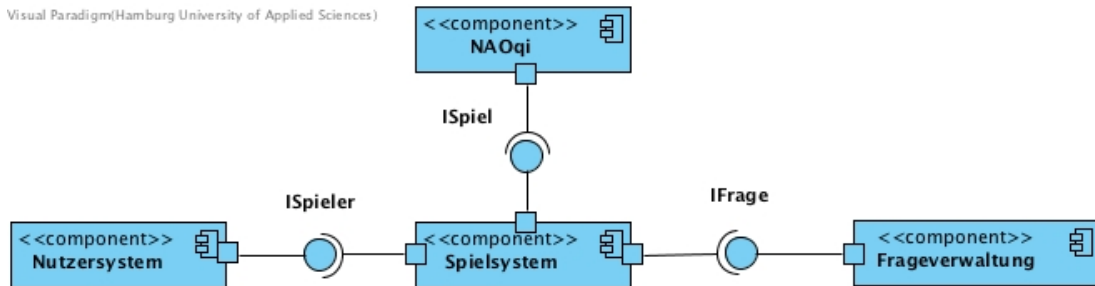


Abbildung 4.10.: Komponentendiagramm: "Quiz-Show".

4.6.3. Interaktionsdiagramm

Nachfolgend werden die betroffenen Interaktionen beim Szenario 3 anhand des Sequenzdiagramms beschrieben. Es beleuchtet die Interaktionen zwischen Objekten und Systemen, um das Verhalten zur Laufzeit zu analysieren.

Die Eigenschaft für das Szenario 3 bezieht sich auf ein kommunikatives Szenario. Durch das ständige Beantworten von Fragen ist der Nutzer in eine Interaktion mit dem NAO involviert. Bei Engpässen besteht die Möglichkeit, sich anhand eines Jokers die Antwortmöglichkeiten von vier auf zwei zu verringern. Im folgenden Sequenzdiagramm werden die Interaktion sowie die Nutzung eines Jokers in einem Spielablauf beschrieben.

Spielablauf

Das Sequenzdiagramm modelliert einen typischen Ablauf der Interaktion während des Spielverlaufs im Spielszenario „Quiz-Show“. An dieser Interaktion sind zwei Lebenslinien von Teilnehmern, die an der Interaktion teilnehmen, beteiligt:

- Eine Rolle, welche vom NAO übernommen wird, um die Show zu moderieren
- Ein Besucher, der die Fragen beantwortet

Damit das Spiel beginnen kann, muss sich der Nutzer in der Hauptauswahl das Szenario verbal aussuchen. Hat das Spiel begonnen, stellt der NAO dem Nutzer so lange Fragen mit den dazugehörigen Antwortmöglichkeiten, bis eine Antwort falsch war oder alle Fragen richtig beantwortet wurden (1, 2). Sobald die Antwortmöglichkeiten aufgezählt worden sind, wird im System eine Stoppuhr gestartet, die bei 0 beginnt und bis 30 Sekunden zählt. In dieser Zeitspanne hat der Nutzer Zeit, um eine Antwort aus den Antwortmöglichkeiten auszusuchen

4. Konzept

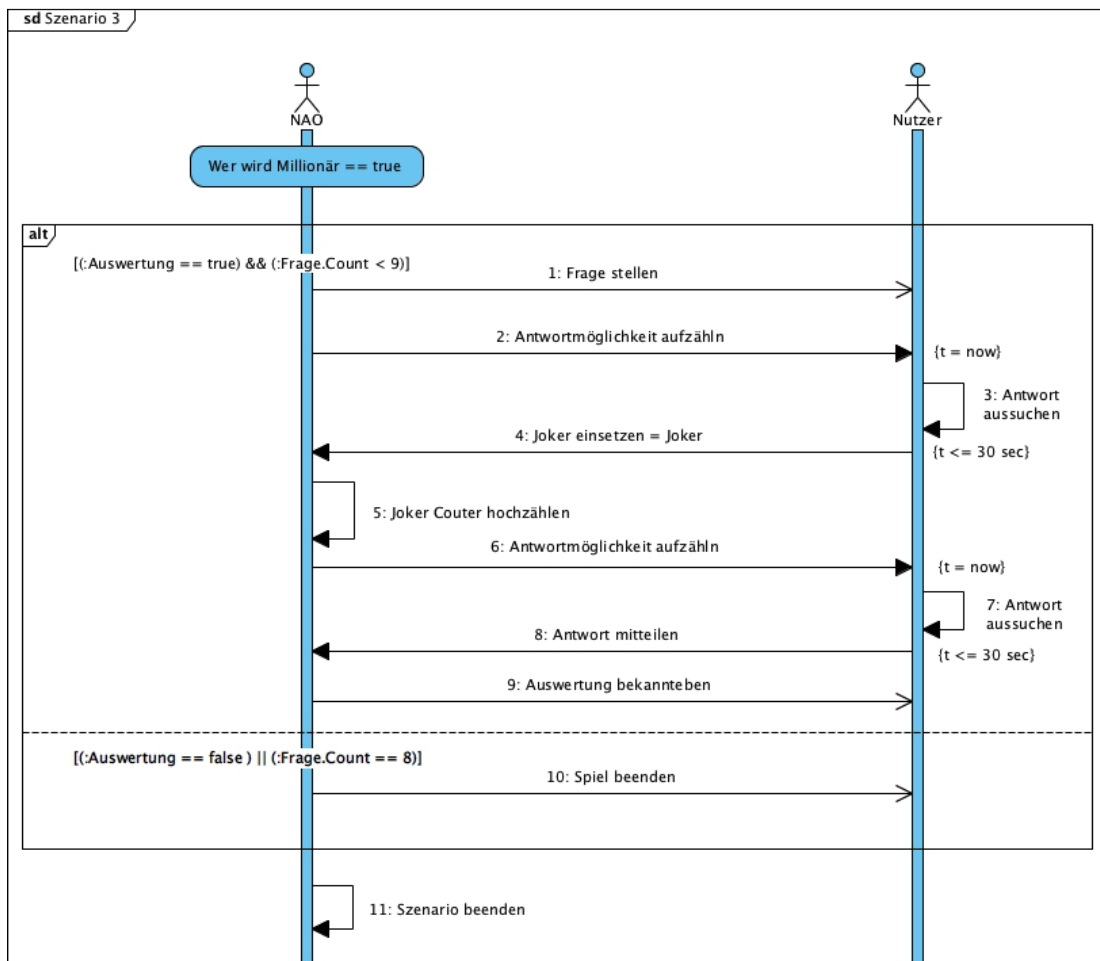


Abbildung 4.11.: Sequenzdiagramm: Szenario 3 "Wer wird Millionär" mit Nutzung eines Joker.

oder einen Joker zu nutzen, und dem Moderator (NAO) die Antwort mitzuteilen (3 – 8). Ist die Zeit abgelaufen, wertet der Moderator die Antwort auf die aktuelle Frage als falsch. Die Antwort des Nutzers erfolgt synchron (ausgefüllte Pfeilspitze), d. h., der Nutzer wartet auf die Auswertung seitens des Moderators (6). Unmittelbar nach der Auswertung sendet der Moderator eine asynchrone Nachricht an den Nutzer, mit dem Ergebnis der Auswertung der Frage (9). Somit wartet der Moderator nicht auf eine Antwort-Nachricht vom Nutzer, sondern fährt sofort mit seiner weiteren internen Auswertung fort, um eine neue Frage zu stellen oder das Spiel zu beenden (10). Ist das Spiel beendet, begibt sich der NAO zurück in die Hauptauswahl, wo der Nutzer ein neues Szenario wählen kann, womit die Interaktion endet (11).

4.7. Konzeption - Szenario 4

Nachfolgend wird das Konzept des Szenarios 4 erörtert. Das Szenario 4 ist eine Erweiterung des Szenarios 1 (vgl. Kapitel 4.4) um die Interaktivität, die durch den Besucher vorgenommen wird. Die Interaktivität erfolgt durch die IDE „Choregraphie“.

4.7.1. Architektur

Die Architektur beschreibt die anwendungsspezifischen Komponenten des Szenarios 4 mittels eines Komponentendiagramms (siehe Abbildung 4.12). Die Architektur besteht aus zwei anwendungsspezifischen Haupt-Komponenten, NAOqi und Choregraphie.

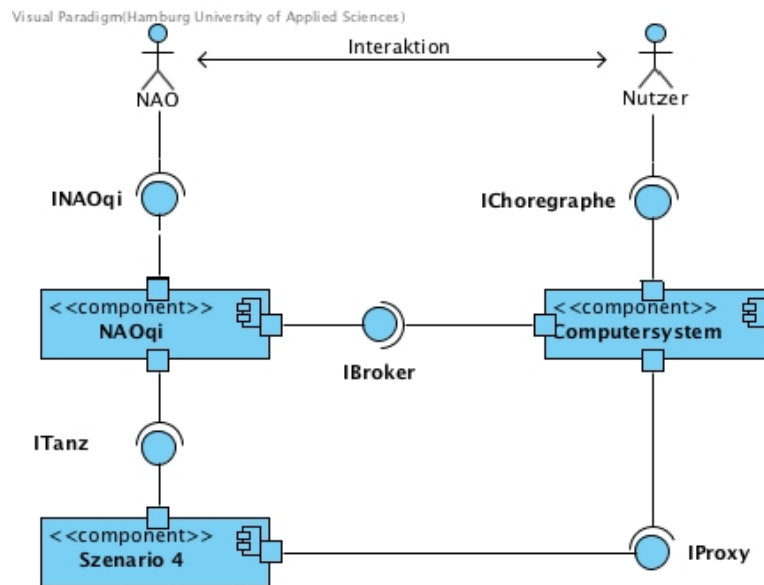


Abbildung 4.12.: Komponentendiagramm: "Tanz erweitern".

NAOqi: Die NAOqi-Komponente hält die Module mit ihren Methoden fest. Dies erlaubt es, den Zustand des NAO abzufragen, und ist für die Einhaltung der Bewegungsabläufe des NAO verantwortlich. Mit der übergeordneten Instanz Broker kann die IDE Choregraphie, mithilfe der Port- und der IP-Adresse des NAO, eine Netzwerkverbindung aufbauen. Der Broker delegiert jegliche Anfragen an das jeweilige Modul.

Choregraphie: Die Komponente Choregraphie hält die IDE fest. Dies bietet dem Nutzer die Schnittstelle für das Erfassen der fehlenden Bewegungen im Tanzszenario. Zudem nutzt

Choregraphe als Schnittstelle von der Broker-Komponente, um spezifische Methoden aufzurufen sowie das Ablegen der Behaviors im NAO.

4.7.2. Interaktionsdiagramm

Das Sequenzdiagramm aus der unten stehenden Abbildung 4.13 modelliert einen typischen Ablauf der Interaktion. Zudem hält es den Zeitpunkt der Interaktivität des Nutzers fest.

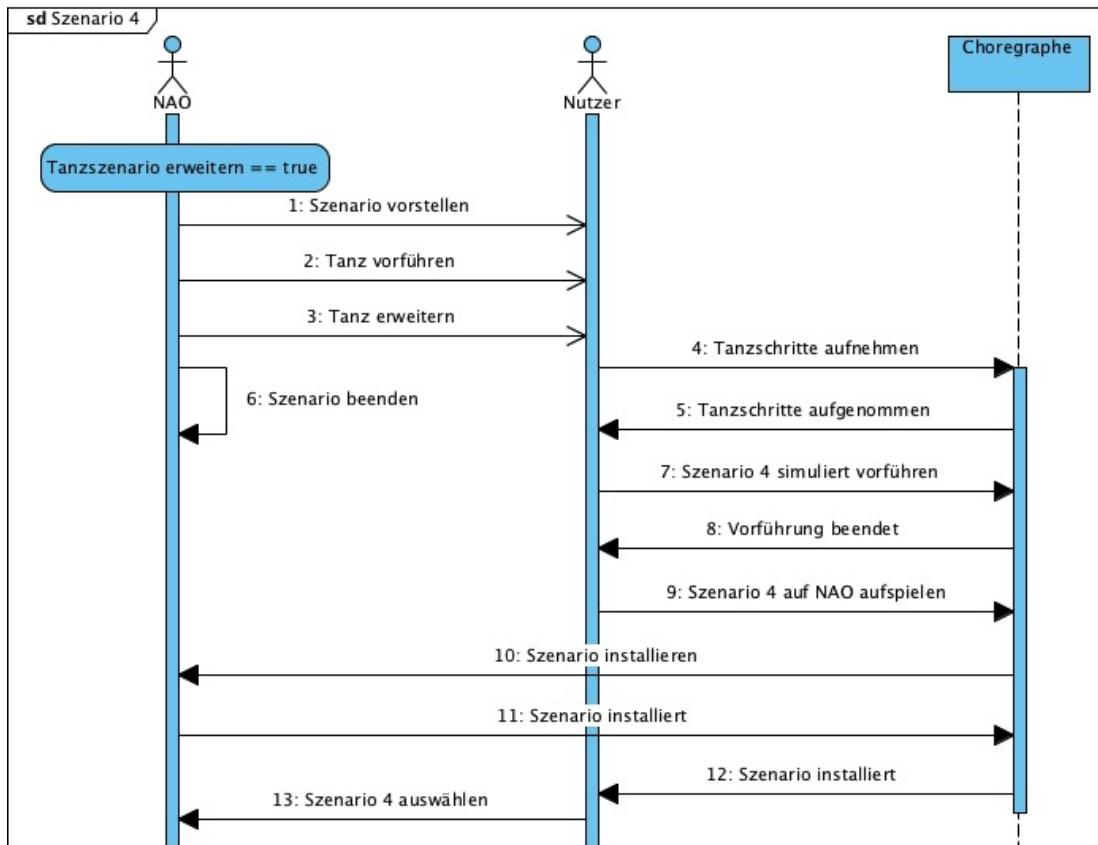


Abbildung 4.13.: Sequenzdiagramm: Szenario 4 "Tanz erweitern".

An der Interaktion sind zwei Lebenslinien von Teilnehmern beteiligt, die an der Interaktion teilnehmen. Eine weitere Lebenslinie wird vom System benötigt, um die Interaktivität des Nutzers zeitlich zu visualisieren.

- Tänzer: Der NAO nimmt die Rolle des Tänzers ein, um den Tanz vorzuführen.
- Zuschauer: Der Nutzer nimmt die Rolle des Zuschauers ein. Er beobachtet die Tänze und bringt sich interaktiv in das Szenario ein.

4. Konzept

- IDE: Die IDE dient dem Nutzer für die Aufnahme fehlender Tanzschritte in das Szenario.

Damit der Tanz beginnen kann, muss der Nutzer das Szenario in der Hauptauswahl verbal aussuchen. Bevor die vordefinierte Tanzeinlage startet, führt der NAO dem Nutzer das spezielle Szenario mit den darin enthaltenen Lücken vor (1). Autonom beginnt der NAO mit seiner einstudierten Tanzeinlage (2). Ist die Tanzeinlage vollendet, wird vom Nutzer verbal um eine Erweiterung des Szenarios gebeten und der NAO bedankt sich bei seinem Publikum (3). Der NAO begibt sich im Anschluss daran in den Hauptzustand zurück, wo der Nutzer verschiedene Szenarien wählen kann (6). Der Nutzer entscheidet sich dafür, die Tanzeinlage durch fehlende Bewegungen über die IDE (4, 5) zu erweitern. Sofern aus Nutzersicht die Erweiterungen vorgenommen wurden, kann der Tanz auf einem simulierten NAO getestet werden (7, 8). Ist der Nutzer mit dem Tanz zufrieden, wird die Überarbeitung des Szenarios auf dem NAO übertragen (9 – 12). Kommt es zur erfolgreichen Installation der Erweiterungen, kann sich der Nutzer das Szenario verbal aus dem Hauptzustand vom NAO aussuchen und seine Version der Tanzeinlage betrachten (13).

In diesem Kapitel wurde auf der Grundlage aus Kapitel 3 aus der Analyse die Konzepte für die vier Szenarien entworfen.

5. Realisierung

Im folgenden Kapitel wird die Realisierung des Konzeptes dargestellt, wobei auf wichtige Aspekte der zu implementierenden vier Szenarien (vgl. Kapitel 5.1 bis 5.4) eingegangen wird. Ersichtlich wird, welche Ziele des Konzeptes realisiert wurden.

Aus der Evaluierung des Abschnitts 4.1 wird die Realisierung mit der IDE genutzt. Alle anfallenden Bewegungen werden mit der Timeline in der IDE implementiert. Durch die Nutzung der Timeline lassen sich Bewegungsabläufe effektiver entwickeln. Die bestimmten Logiken und Strategien der Spielszenarien wurden mit Python umgesetzt. Die Projektstruktur der vier Szenarien ist jeweils nach demselben Prinzip realisiert worden. Jedes Szenario besitzt seine eigenen Hauptebenen, die mit weiteren Flussdiagramm-Boxen verschachtelt sind. Anhand der Flussdiagramm-Boxen werden die Logiken und Eigenschaften des jeweiligen Szenarios voneinander getrennt, um eine lose Kopplung der Komponenten zu erreichen.

5.1. Umsetzung - Szenario 1

Aus dem Abschnitt 4.4 wird das Konzept für das Szenario 1 ersichtlich, das vollständig realisiert wurde.

Erstellung von Bewegungen

Im Folgenden wird auf einige Besonderheiten der Realisierung eingegangen. Als Beispiel für einen Tanz dient das Musikstück „Macarena“.

Choregraphie erlaubt es, die aktuelle Position des NAO anhand eines Keyframes in der Timeline festzuhalten. Mit der Timeline können unendlich lange Bewegungen aufgenommen werden, wie es die physikalische Ebene des NAO erlaubt.

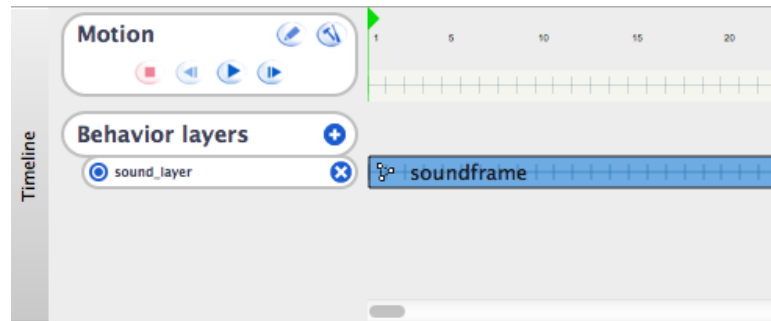


Abbildung 5.1.: Timeline zur Aufnahme von Bewegungen.

Die rechteckigen Boxen repräsentieren ein Keyframe in der Timeline, welches dem NAO erlaubt, diejenige Position einzunehmen, welche er gespeichert hat. Die Nummerierung der Timeline bezieht sich auf die Zeit in Millisekunden und hält die Sequenzen der Frames fest.

Jede folgende Nummerierung in der Timeline stellt eine Zeitspanne von 10 Millisekunden dar. Das Erreichen von Frame 10 dauert 100 ms, 200 ms sind es bis zum Erreichen von Frame 20. Die Körperhaltung kann zu einer beliebigen Zeit in der Timeline mittels *Store joints in keyframe -> Whole body (F8)* aufgenommen werden. Die IDE bietet verschiedene Optionen, um die Einstellung der Winkel zu treffen. Eine dieser Optionen gelingt anhand des Animationsmodus. Beim Aktivieren des Animationsmodus ist der Roboter in einem Warte zustand. Damit ein Körperteil in die entsprechende Position gestellt wird, muss dafür zwischenzeitlich die Steifheit des NAO abgestellt werden. Dieses kann man durch das Berühren bestimmter Sensoren am NAO oder die 3D-Ansicht des simulierten NAO in der IDE abstellen. Die Tabelle 5.1 dient der Übersicht über die Sensoren, mit denen die Steifheit der Gelenke des NAO abgestellt werden kann. Der Animationsmodus erlaubt es, den NAO wie eine Marionette in die jeweilige Wunschposition zu bewegen, um die Winkel der Gelenke in einem Keyframe festzuhalten.

Körperbereich	Taktiler Sensor
Bein	Jedes Bein besitzt einen Bumper mit dem die Steifheit gelöst oder gesperrt wird.
Arm	Jeder Arm besitzt berührungs-Sensoren. Werden diese umschlossen löst oder sperrt sich die Steifheit des Arms.
Kopf	Durch betätigen des Mittleren Kopfsensors kann die Steifheit des Kopfes gelöst oder gesperrt werden.

Tabelle 5.1.: Sensoren um NAOs Steifheit der Gelenke zu lösen oder sperren.

5. Realisierung

Der Timeline können weitere Ebenen hinzugefügt werden, um ein paralleles Abspielen von Modulen aufzubauen. Die Root-Ebene enthält immer die Bewegungsabläufe mit den entsprechenden Keyframes. Eine weitere Ebene ist für das Szenario erforderlich. Auf dieser Ebene wird das Soundsystem des NAO in Anspruch genommen, welches parallel zu den Bewegungsabläufen die passenden Audiodateien abspielt. Das vorgefertigte Behavior Play Sound File wurde für das Abspielen von Audiodateien genutzt, da es das Modul ALAudioPlayer aus dem Abschnitt 4.4.2 realisiert.

Damit die Bewegungsabläufe für das Musikstück „Macarena“ aufgenommen werden, müssen im Vorfeld die einzelnen Tanzschritte identifiziert werden. Die Abbildung 8 zeigt die Hauptschritte der einzelnen Tanzschritte. Die Tanzschritte bestehen aus insgesamt 13 Hauptschritten, welche willkürlich in Keyframes aufgenommen wurden. Zu jedem Keyframe erfolgt eine Interpolation, welche jeweils eine sanfte Bewegung umsetzen soll. Trotz der Interpolation zwischen allen Keyframes kommt es meistens zu ruckartigen und sehr klumpigen, sozusagen roboterartigen Bewegungen. Um die Bewegungen menschlicher darzustellen, mussten zwischen allen Keyframes weitere Nebenschritte aufgezeichnet werden. Die Timeline besitzt zudem einen Erweiterungsektor, der es ermöglicht, präzisere Einstellungen bezüglich des aufgenommenen Keyframes vorzunehmen. Somit kann jeder Keyframe weiter in einzelne Gelenkbereiche aufgeteilt werden, um neue Keyframes zu konstruieren (siehe Abbildung 5.2). Dies hat den Vorteil, schönere Bewegungsbögen für den Bewegungsablauf festlegen zu können. Die Aufteilung der Keyframes steuert eine asynchrone Bewegung herbei und sollte bei jedem Tanz wohlüberlegt eingesetzt werden. Es stellt sich heraus, dass die Aufteilung der Keyframes bei dem Musikstück „Ententanz“ keine gute Idee ist, da das Klatschen hierbei ein paralleles Bewegungsmuster ist.

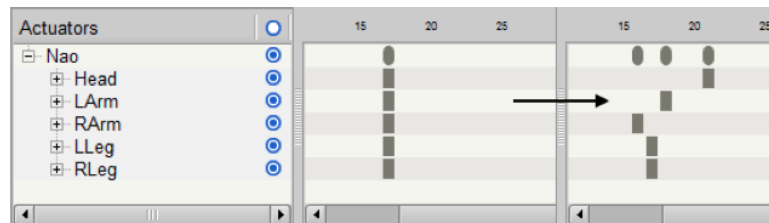


Abbildung 5.2.: Erweiterungsektor der Timeline.

Der Tanz „Macarena“ besteht aus insgesamt 60 einzelnen Keyframes, was im Zusammenspiel mit der Interpolation einen schönen Tanzverlauf darstellt.

Keyframe Anordnung

Die für das Szenario ausgewählten Musikstücke besitzen ein unterschiedliches Tempo und dieses wird in der Einheit bpm (beats per minute) angegeben. Dies muss für ein korrektes Platzieren der einzelnen Keyframes in der Timeline beachtet werden, damit die 13 Hauptschritte an der richtigen Zeitposition gesetzt werden.

Um das Tempo des Musikstücks zu analysieren, werden der Song und eine Stoppuhr benötigt. Die meisten Musikstücke bestehen aus einem 4/4-Takt, der sich aus 32 Beatschlägen ergibt. Jeder Vierteltakt umfasst acht Beatschläge. Ausschließlich bei den Musikstücken „Hound Dog“ und „I Feel Good“ besteht der 4/4-Takt aus 16 Beatschlägen, was zur Folge hat, dass jeder Takt mit vier Beatschlägen gezählt wird. Anhand einer Stoppuhr wird die Zeit bis zum Erreichen des 4/4-Taktes gemessen. Mithilfe der unten stehenden Formeln und des ermittelten Zeitintervalls können die bpm jedes Musikstückes berechnet werden.

$$bpm(32\text{Beatschlaegen}) = \frac{32bpm}{ts} \cdot 60s \quad (5.1)$$

$$bpm(16\text{Beatschlaegen}) = \frac{16bpm}{t} \cdot 60s \quad (5.2)$$

Die Tabelle 5.2 fasst die ermittelten Zeiten und die dazu berechneten bpm der jeweiligen Musikstücke zusammen. Durch die berechneten bpm können die 13 Hauptschritte des Tanzes

Musikstück	T	bpm
Macarena	18,58	103
Ententanz	19,43	99
Kung Fu	18,93	101
Hound Dog	11,25	85
I Feel Good	11,53	83
Uptown Funk	16,58	115
Loose Yourself	22,4	85
Gangnam Style	14,8	130

Tabelle 5.2.: Die berechneten Beats pro Minute.

von „Macarena“ korrekt an der Zeitachse in der Timeline platziert werden. Die 13 Hauptschritte decken den 2/4-Takt des Musikstücks „Macarena“ ab und müssen somit einmal wiederholt werden, damit der 4/4-Takt abgedeckt ist. Dadurch, dass in 60 Sekunden 103 Beatschläge untergebracht sind, wird die Zeit gesucht, in der die 26 Hauptschritte platziert werden. Hierfür wird ein Dreisatz aufgestellt und nach der Zeit der 26 Hauptschritte gesucht. Das Endresultat

ist, dass alle 0,6 Sekunden ein Hauptframe in der Timeline platziert wird, damit der NAO im Takt einen Beat mit der passenden Bewegung trifft. Die weiteren Tänze für die Choreographie wurden nach derselben Vorgehensweise analysiert, der Hauptframe wurde aufgezeichnet und ist anhand des Taktes korrekt zum Musikstück platziert worden.

5.2. Umsetzung - Szenario 2

Aus dem Abschnitt 4.5 wurde das Konzept für das zweite Szenario ersichtlich und vollständig realisiert. Im folgenden Abschnitt wird auf die Besonderheit des einzelnen Quellcodes bei der Implementierung eingegangen.

Für die Realisierung des Szenarios wurden zuerst Bewegungsabläufe und die Symbole „Sche-re“, „Stein“ und „Papier“ mit dem Animationsmodus in der IDE erstellt, wie in Abschnitt 5.1 beschrieben worden ist.

Spielablauf

Die Abbildung 5.3 zeigt den gesamten Spielverlauf mit dem Kernverhalten des Spieles. Sie besteht aus weiteren Flussdiagramm-Boxen.

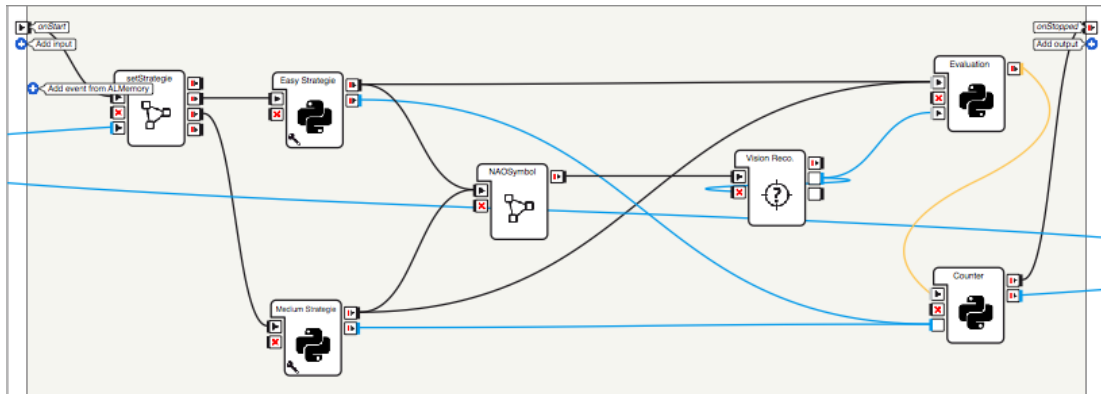


Abbildung 5.3.: Übersicht der Behavior des Szenario 2. Erstellt mit der IDE.

setStrategie-Box

Die *setStrategie*-Box dient zum Festlegen einer Strategie im gesamten Spielverlauf für den NAO. Der NAO befindet sich derzeit in einem Wartezustand und wartet mithilfe des Moduls Speech Recognizing auf die Antwort des Nutzers, was im Abschnitt 4.3 näher erörtert

5. Realisierung

wurde. Der NAO erkennt aus einer vordefinierten Liste die Wörter *easy*, *medium* sowie *hard*. Ist ein Element aus der Liste identifiziert worden, wird über einen Switch-Case die Strategie für den gesamten Spielverlauf festgelegt.

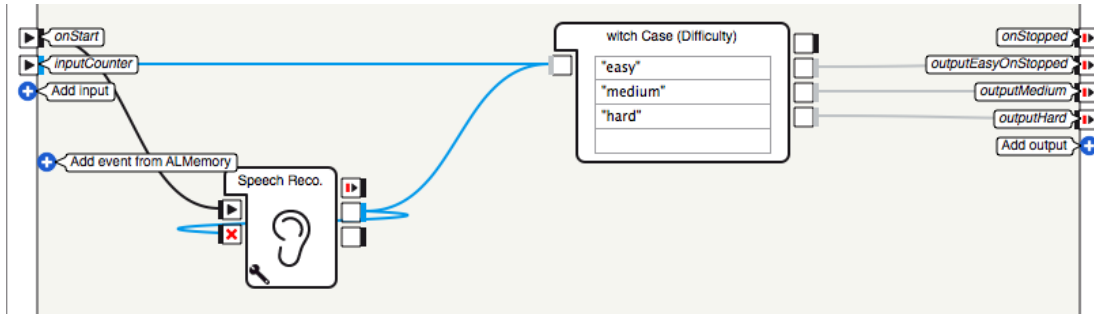


Abbildung 5.4.: Initialisierung.

Easy Strategie-Box

Die Easy-Strategie ist in Python implementiert und hält die Déjà-vu-Strategie mit dem zugehörigen Verhalten aus dem Abschnitt 4.5.2 fest. Das Verhalten bei der Easy-Strategie besagt, dass in jeder Spielrunde dasselbe Symbol gewählt wird.

```
1 import random
2
3 class MyClass(GeneratedClass):
4     _instance = None
5
6     def __init__(self):
7         if not self._instance:
8             self._instance = GeneratedClass.__init__(self)
9             self.grn = random.randint(1, 3)
10        return self._instance
11
12    def onInput_onStart(self):
13        #output with the generatet random number
14        self.onStopped(self.grn)
15        #output with a string
16        self.outputEasy("easy")
17    pass
```

Listing 5.1: Easy Strategie

Der oben angegebene Quellcode hält die Initialisierungsmethode fest. Der wesentliche Kern bei dieser Methode gilt der Zahl der Objekte, die erzeugt werden. Es soll maximal ein Objekt mit seinen Eigenschaften existieren. Beim Aufruf der Methode wird primär die Abfrage nach einer erzeugten Instanz gestellt (Zeile 7). Gibt es ein positives Ergebnis, ist die Schlussfolgerung, dass derzeit kein Objekt instanziiert worden ist. Somit wird das Objekt (Zeile 8) mit den dazugehörigen Eigenschaften (Zeile 9) angelegt. Während der Instanzierung werden mithilfe der Random-Klasse Zufallszahlen generiert. Mit den entsprechenden Parameterübergaben 1 bis 3 können die Zahlen Eins, Zwei und Drei generiert werden. Über die generierte Zahl wurde das zu spielende Symbol getroffen. Die generierte Zahl dient im weiteren Verlauf als Input für die Box *NAOSymbol* sowie zur *Evaluation* der Spielrunde. Ist jedoch die Abfrage nach einer nicht existierenden Instanz negativ, so lautet die Schlussfolgerung, dass ein Objekt existiert. Dann wird kein neues Objekt erzeugt, sondern dieses mit seinen Eigenschaften zurückgeliefert. Somit wird die aktuelle Spielrunde mit dem bereits erzeugten Symbol für die nächste Spielrunde gespielt.

Medium Strategie-Box

Die Medium-Strategie ist in Python geschrieben und hält die Random-Strategie mit dem entsprechenden Verhalten aus Abschnitt 4.5.2 fest. Das Verhalten bei der Random-Strategie besagt, dass in jeder Spielrunde ein neues zufälliges Symbol gewählt wird.

```
1 def onInputOnStart(self):
2     grn = random.randint(1, 3)
3     #output with the generatet random number
4     self.onStopped(grn)
5     #output with a string
6     self.outputMedium("medium")
7     pass
```

Listing 5.2: Medium Strategie

Im oben angegebenen Quellcode ist die Methode `onInputOnStart(self)` aufgelistet. Durch Nutzung der Random-Klasse wird auch bei dieser Strategie die Auswahl eines Symbols zufällig getroffen, das für die aktuelle Spielrunde gespielt wird. Durch eine erfolgreiche Spielrunde wird ein weiteres Inputsignal in die Klasse gesendet. Somit erlangt der Zufallszahlengenerator den Anstoß, ein neues, zufälliges generiertes Symbol zu erzeugen, um damit in der neuen Spielrunde anzutreten.

NAOSymbol-Box

Die Box *NAOSymbol* hält intern die drei Armbewegungen mit den dazugehörigen Symbolen *SStein*, *Schere*, *Papier* fest (siehe Abbildung 5.5). Über einen Switch-Case wird das Eingangssignal, das bei der zuvor gespielten Strategiewahl zufällig vom Zufallszahlgenerator generiert wurde, ausgewertet. Die dazu passende Gestik wird dem Nutzer gezeigt, das zum richtigen Zeitpunkt.

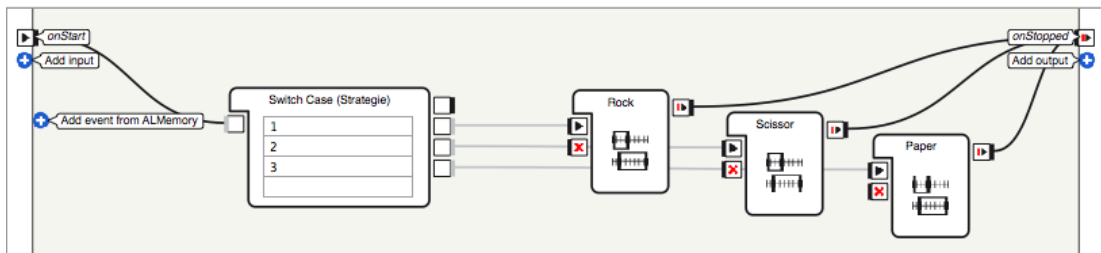


Abbildung 5.5.: Auswertung des zu spielende Symbol von NAO.

Vision Recognition-Box

Sobald der Zeitpunkt zum Aufdecken der zu spielenden Symbole eintrifft, startet der NAO mit dem Modul *ALVisionRecognition*, um die Strategie des Nutzers zu identifizieren. Eine genauere Beschreibung des Moduls ist im Abschnitt 4.5.3 erfolgt. Nach dem erfolgreichen Wiedererkennen des zu spielenden Symbols des Nutzers wird ein String-Signal in die *Evaluation* gesendet. Der String enthält den Namen des identifizierten Symbols des Nutzers.

Evaluation-Box

Die Evaluations-Box ist in Python geschrieben und dient zur Auswertung des Gewinners. Sie benötigt zwei Inputsignale, um anhand der Auswertungsmatrix (siehe Abschnitt 3.4) eine korrekte Auswertung durchzuführen. Das erste Inputsignal enthält die zu spielende Strategie des NAO selbst. Sie wird zufällig oder bewusst durch Generierung aus der *Strategie-Boxen* festgelegt. Das Eintreffen des Signales dient zur Anregung der Methode `onInputOnStart(self, p)`. Die Methode konvertiert den Parameter in den entsprechenden Typen um. Dieser ist eine ganzzahlige Zahl von Eins bis Drei und wird bei der späteren Auswertung wiederverwendet.

```

1 def onInput_onStart(self, p):
2     #Aufruf der Konvertierungsmethode
3     p = self.typeConversion(p)

```

```
4     #globale Variable fuer abspeicherung des Kovertiertentyps
5     global cmp
6     cmp = p
7     pass
```

Listing 5.3: Quellcodeabschnitt Evaluation-Box: Inputsignal des NAO

Sobald der NAO die zu spielende Strategie des Besuchers identifiziert hat, dient das Eintreffen des Ausgangssignals der *VisualRecognition-Box* zum Starten der Methode `onInputInput-Player(self, p)`. Die Methode erwartet als Parameter das identifizierte Symbol des Nutzers, welches als String übergeben wird. Der Parameter wird typgerecht in eine ganzzahlige Zahl konvertiert. Nach dem erfolgreichen Setzen der Symbole sind von beiden Spielern die Strategien bekannt und die Spielrunde kann ausgewertet werden.

```
1 def investigatingWinner(self, computer, player):
2     computer = computer
3     player = player
4
5     #symbole = { 1 : "Stein", 2 : "Schere", 3 : "Papier" }
6     rules = { 1 : 2, 2 : 3, 3 : 1 }
7
8     if computer == player:
9         self.tts.say("No_one_Win")
10        self.win = 2
11    elif rules[computer] == player:
12        self.tts.say("I_Win")
13        self.win = 1
14    else:
15        self.tts.say("OH_I_Loose")
16        self.win = 0
17    pass
```

Listing 5.4: Quellcodeabschnitt Evaluation-Box: Spielrunde Auswertung

Im oben angegebenen Quellcode wird die Methode zur Auswertung der Spielrunde abgebildet. Die Signatur erwartet zwei Parameterwerte (Zeile 1), dies sind die zu spielenden Strategien des Nutzers und des NAO. Die ganzzahligen Nummern repräsentieren die Symbole des Spieles. Somit entsteht folgende Regelung: 1 schlägt 2, 2 schlägt 3 und 3 schlägt 1. Bei der Auswertung (Zeile 8) wird als Erstes überprüft, ob beide Spieler dieselbe Strategie gewählt haben `If computer == player`. Ist dies der Fall, so wird die Spielrunde als Unentschieden gewertet.

Bei der folgenden Bedingung (Zeile 11) wird geprüft, ob der NAO der Gewinner der Spielrunde ist

`elif rules[computer] == player`. Dies verhält sich wie folgt: Anhand der Strategiezahl des NAO wird die Auswertungsmatrix `rules = (1 : 2, 2 : 3, 3 : 1)` die Zahl zurückliefern, mit welcher der NAO überlegen ist. Hat der Nutzer genau die Strategie gewählt, welche die Auswertungsmatrix zurückliefert, heißt dies, das die Strategie vom NAO der des Mitspielers überlegen ist. Somit geht der NAO als Gewinner aus der Spielrunde hervor. Schlussfolgernd gilt der Nutzer als Gewinner, wenn es kein Unentschieden gibt und der NAO nicht gewonnen hat (Zeile 14).

5.3. Umsetzung - Szenario 3

Aus dem Abschnitt 4.6 wurde das Konzept für das dritte Szenario ersichtlich und vollständig realisiert. Im folgenden Abschnitt wird auf die Besonderheit der einzelnen Quellcodes bei der Implementierung eingegangen. Als Beispiel dient die Realisierung einer Frage.

Dadurch, dass der NAO bei diesem Szenario nur moderiert, findet die Realisierung ohne Bewegungsabläufe statt. Der NAO wird im Vorfeld über das Behavior *SitDown* in eine Sitzposition gebracht. Das Behavior sorgt dafür, dass der NAO sanft in die Sitzposition gesetzt wird, egal, aus welcher Körperhaltung sein aktueller Zustand besteht.

Spielablauf

Die Main-Box (siehe Abbildung 5.6) hält den gesamten Spielverlauf mit dem Kernverhalten des Spieles „Quiz-Show“ fest. Es besteht aus mehreren Flussdiagramm-Boxen, welche verschachtelt und aneinandergereiht sind. Für die Implementierung sind die vorgefertigten Module aus NAOqi genutzt worden (siehe Tabelle 4.3).

Die Main-Box besteht aus einer weiteren Flussdiagramm-Box (Questions-Box) und drei Sprach-Boxen. Die *Questions-Box* besitzt drei Ausgangssignale und zeigt damit verschiedene Zustände des Spieles auf. Je nach interner Auswertung wird ein Ausgangssignal gesendet. Die *Sprach-Boxen* dienen zur Mitteilung des aktuellen Zustands an den Nutzer. Die Zustände sind „Falsch“, „Spiel gewonnen“ und „Zeit überzogen“. Jeder Aufruf der Sprach-Boxen leitet nach dem Ablauf die sofortige Beendigung des Spieles ein.

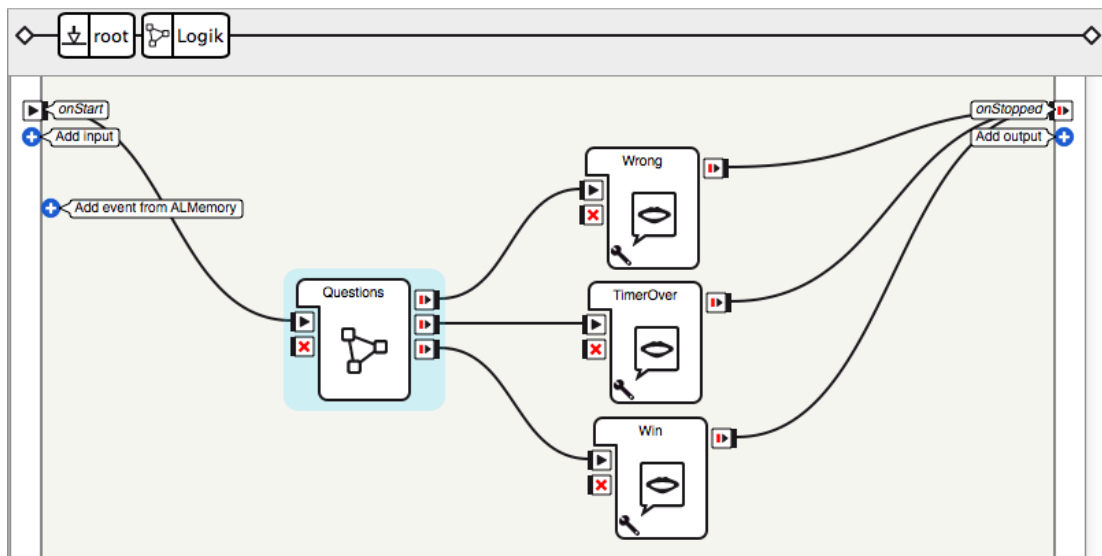


Abbildung 5.6.: Hauptebene des Spiel "Quiz-Show".

ALTextToSpeech

Die *Sprach-Boxen* werden ständig vom Modul `ALTextToSpeech` genutzt, das im Abschnitt 4.3 für die Nutzung des Szenarios identifiziert wurde. Das Modul konvertiert den geschriebenen Text in gesprochene, d. h. hörbare Audioreden. Somit kann der NAO alles verbalisieren, was vorab in Textform erfasst worden ist.

Durch das Nutzen des Attributes `post` können Methodenaufrufe in einem parallelen Task ausgeführt werden. Das Attribut muss beim Aufrufen der genutzten Methode gesetzt sein.

```
1 tts.post.say("Congratulation_\Pau=200\._You_win_the_game.")
```

Listing 5.5: Quellcodeabschnitt parallele Task beim Methodenaufrufe setzten

Der obige Quellcodeausschnitt zeigt, dass zwischen dem `ALTextToSpeech-Objekt` und dem Methodenaufruf `say()` das Attribut `post` genutzt wird. Die Sprachanweisung wird im Hintergrund neben anderen Modulen ausgeführt. Das Modul erlaubt, Einstellungen bei der Stimmlage und der Redegeschwindigkeit vorzunehmen. Es trägt durch Verringern der Redegeschwindigkeit zum Verständnis des Texts bei. Hierfür können Makros eingebunden werden. Mit dem Makro `\RSDP=X\` wird die Redegeschwindigkeit verändert. Für `X` wird ein Prozentwert von 0 bis 100 Prozent gesetzt. Aber auch hier gibt es Textpassagen, die der NAO immer noch zu schnell ausführt. Durch das Setzen von weiteren Makros in bestimmten Textpassagen können

zusätzlich Pausen erzwungen werden, um die jeweilige Passage deutlicher auszusprechen. Dies wird mithilfe von `\Pau=X\` vorgenommen. Für X wird eine Zeit in Millisekunden festgelegt.

Question (Ebene 2)

Die Question-Box (Ebene 2) ist die tiefste Ebene im Flussdiagramm des Spieles. Der Aufbau einer Question-Box umfasst fünf Sprach-Boxen und vier Wait-Module. Die erste Sprach-Box enthält die aktuell dem Nutzer zu stellende Frage. Die weiteren vier Sprach-Boxen halten die dazugehörigen Antwortmöglichkeiten bezüglich der Frage fest. Es ist sinnvoll, dass der NAO eine kleine Pause vornimmt, bevor die nächste Sprach-Box angeregt wird. Dies erleichtert dem Nutzer die akustische Aufnahme der Fragen sowie der Antwortmöglichkeiten.

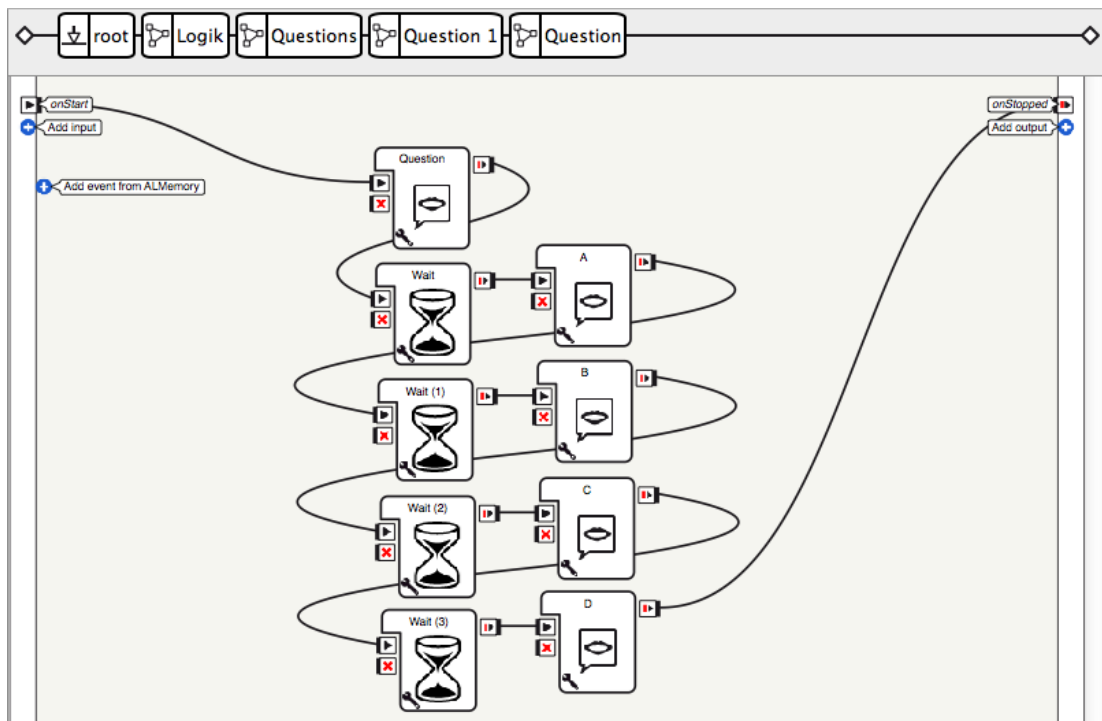


Abbildung 5.7.: Aufbau der Fragen und der Antwortmöglichkeiten”.

Question (Ebene 1)

Die Question-Box (Ebene 1) dient zur Aufforderung und Auswertung der Antwort des Nutzers. Damit der NAO die Antwort des Nutzers aufnimmt, wird das Speech-Recognition-Modul eingesetzt. Dieses Modul ermöglicht dem NAO, Spracheingaben des Nutzers aus einer vordefinierten

5. Realisierung

Liste von Wörtern wiederzuerkennen. Die Liste enthält die Buchstaben „A“, „B“, „C“ und „D“. Wurde ein Element vom NAO gehört, das gut zu einem Element aus der Liste passt, wird das Ausgangssignal mit dem Element gesendet. Parallel startet das System eine Zeitintervall von 30 Sekunden mit dem vorgegebenen Timer-Modul. Der Timer dient für das System als zeitliche Begrenzung zur Abgabe der Antwort. Das Zeitintervall wird unterbrochen, sobald die Antwort des Nutzers ein Element des Spracherkennungsmoduls trifft. Andernfalls wird, sobald der Timer abgelaufen ist, ein Ausgangssignal an die Haupt-Box gesendet. Dieses Signal regt die entsprechende Sprach-Box an, dem Nutzer den Endzustand des Spieles mitzuteilen.

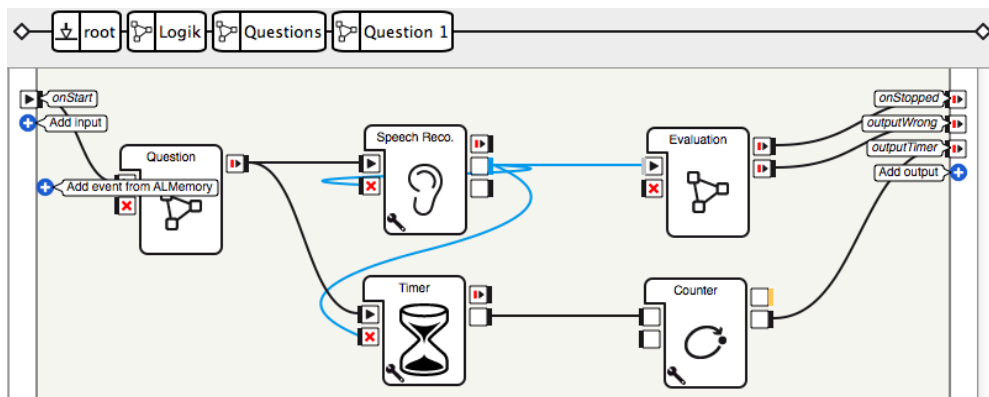


Abbildung 5.8.: Überblick des Ablaufes einer gestellten Frage.

Evaluation

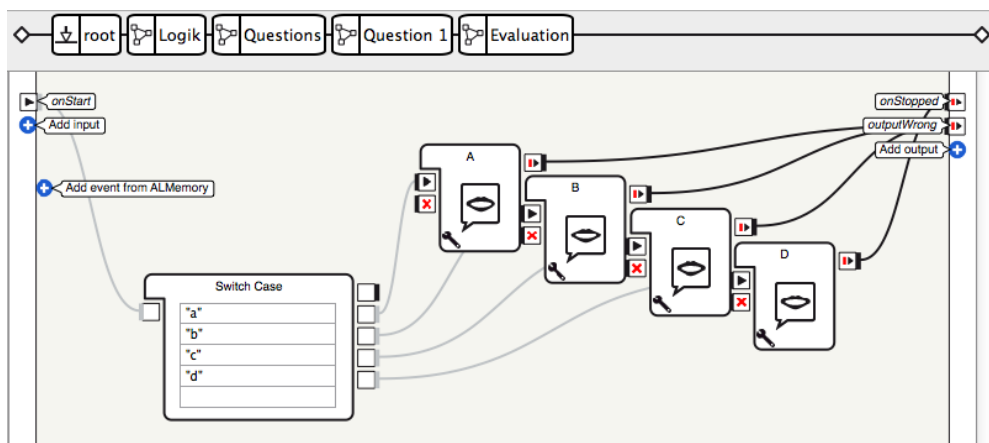


Abbildung 5.9.: Überblick des Switch-Case.

Die Evaluations-Box dient zur Auswertung der Antwort des Nutzers. Da das Eingangssignal die Antwort als Parameter bereitstellt, kann anhand eines Switch-Case leicht auf Korrektheit überprüft werden. Die Evaluations-Box hält zwei weitere Ausgänge für unterschiedliche Zustände fest. Sie dienen der Festlegung dessen, ob die Antwort korrekt oder falsch ist. Wurde die betreffende Frage falsch beantwortet, führt das Ausgangssignal bis auf die oberste Ebene und beendet das Spiel. Andernfalls dient das Ausgangssignal zur Auslösung der nächsten Frage.

5.4. Umsetzung - Szenario 4

Der kommende Abschnitt verdeutlicht die Aufnahme von Bewegungsabläufen anhand der **IDE** und nicht die konkrete Realisierung des unvollständigen Tanzes. Die Umsetzung des unvollständigen Tanzes ist dem vorherigen Abschnitt 5.1 zu entnehmen, da der Tanz nach den gleichen Prinzipien entwickelt worden ist wie die vorherigen Tänze aus dem Szenario 1. Es behandelt den Erweiterungspunkt, welchen die **IDE** für das Szenario darstellt.

Bewegungsablauf

Bewegungsabläufe werden in Choregraphie anhand von Keyframes festgehalten. Die simple Zeitachse ist nicht konform, um Bewegungen in Keyframes zu speichern, daher kann der Nutzer auf dem Stift-Button die erweiterte Zeitachse öffnen (siehe Abbildung 5.10). Die erweiterte Zeitachse erlaubt das Ansprechen und Aufzeichnen einzelner Körperregionen des NAO im Keyframe.

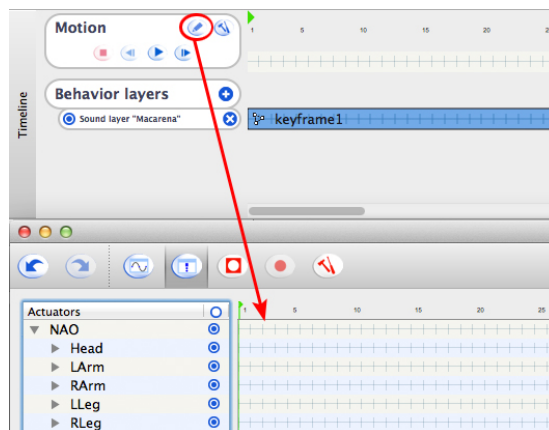


Abbildung 5.10.: Erweiterte Timeline öffnen.

Simulation des NAO

Mithilfe des Simulators in der IDE können alle Körperregionen einzeln bewegt werden. Durch Auswählen einer Körperregion im Simulator können entsprechend die Winkel geändert werden. Hierfür muss vorab die Steifheit der Körperregion aufgehoben werden. Sobald sich die Farbe bezüglich der Steifheit von Rot zu Grün ändert, können die Gelenke bewegt werden. Diese Methode erweist sich in der Praxis als knifflig, da die Konzentration der einen Hand beim NAO und die der anderen Hand beim Klicken verharret, um die Steifheit zu deaktivieren.

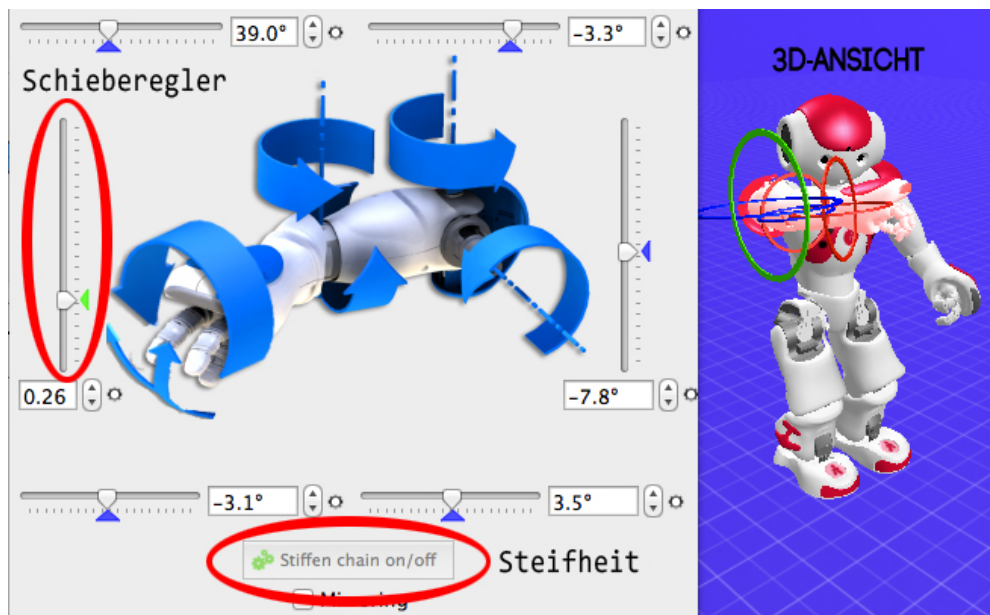


Abbildung 5.11.: Bewegungsfreigabe einzelner Körperregionen über den NAO-Simulator.

Als weitere Variante, um Bewegungen aufzunehmen, gibt es den Animationsmodus. Diese Methode wurde zur Realisierung aller anfallenden Bewegungsabläufe in den Szenarien eingesetzt. Bei der Realisierung des Szenarios 1 wird diese Methode im Abschnitt 5.1 erörtert.

Das Kapitel 5 hält die Entwicklung der Konzepte aus dem Kapitel 4 fest. Es wurden die Anforderungen der Anwendungsarchitektur implementiert.

6. Systemvalidierung

Um den Erfolg der Realisierung zu prüfen, wird nachfolgend die Validierung des Systems erläutert. Für jedes der vier zuvor beschriebenen Szenarien (vgl. Kapitel 5.1 bis 5.4) gibt es zahlreiche Vorbedingungen und Kriterien, die für eine erfolgreiche Bewertung der Realisierung erfüllt werden müssen. Die im Rahmen der Analyse an alle Szenarien gestellten Anforderungen aller Szenarien aus der Analyse 3 werden im Hinblick auf die Richtigkeit des Konzeptes 4 und der Implementierung 5 abgeglichen. Für die Validierung wurde die implementierte Präsentation auf den NAO geladen und ausgeführt, um das System unter realen Bedingungen zu testen. Die Abläufe fanden in der Zeit vom 10. July 2016 bis zum 30. July 2016 stets an den Bachelor-/Master-Arbeitsplätzen an der Hochschule für Angewandte Wissenschaften Hamburg im Raum 11.87 statt.

6.1. Validierung - Szenario 1

Das erste Szenario (vgl. Kapitel 3.4) setzte den Schwerpunkt auf den kinematischen Körper, wobei es als Tanzszenario realisiert wurde. Mithilfe von verschiedenen Tänzen wurde eine Choreographie für den NAO erstellt. Sobald die Hauptchoreographie abgelaufen ist, nimmt der NAO eine Interaktion mit dem Nutzer auf, wobei Letzterer den Schlusstanz auswählen kann.

Vorbedingungen

Der Validierungsablauf wurde auf verschiedenen Böden (Teppichboden, Linoleum, Holz (Tisch)) ausgeführt, um die Stabilität des NAO beim Ablauf zu testen. Dabei müssen gemäß der zehn Durchläufe für jede Bodenbegebenheit stattfinden. Zudem muss der Akku vollständig geladen sein, um den Durchlauf jeweils ohne das Ladekabel durchzuführen.

Validierungskriterien

Folgende Kriterien muss der NAO beim Ablauf erfüllen, damit das Szenario als erfolgreich betrachtet werden kann:

- Durchlauf der Choreographie, ohne zu fallen
- Durchlauf der Choreographie, bevor der Akku entladen ist
- Den Nutzer zur Wahl des Tanzes auffordern
- Aufforderung des Nutzers entgegennehmen und korrekt auswerten

Validierungsablauf

Es folgt eine Ablaufbeschreibung für ein Erfolgsszenario. Die weiteren Durchläufe werden im Validierungsergebnis erörtert.

Der NAO ist im Wartezustand und erwartet die Eingabe zur Szenarienwahl des Nutzers. Mit der Identifikationsnummer „1“ wurde das Szenario 1 ausgewählt. Als Erstes folgt eine Initialisierung, indem der NAO in eine aufrechte Position gebracht wird. Hat der NAO die Position erfolgreich erreicht, startet gemäß Implementierung autonom die Vorführung der Hauptchoreographie. Erreicht der NAO die Endposition der Hauptchoreographie, begibt er sich in eine aufrechte Position zurück. Der NAO startet parallel eine Interaktion mit dem Nutzer. Die entsprechende Aufforderung betrifft die Auswahl zur Schlusschoreographie. Es ist das Modul *TextToSpeech* in Kombination mit *SpeechRecognition*, das diese Interaktion ermöglicht. Das System enthält eine Liste von Wörtern, auf die der NAO entsprechend reagiert. Der Nutzer kann zwei unterschiedliche End-Tänze wählen. Anhand des Identifikationsbuchstabens „A“ ist die Schlusschoreographie vom Nutzer gewählt worden. Nach erfolgreicher Auswertung beginnt der NAO mit der gewünschten Schlusschoreographie. Durch das Abarbeiten dieser wird das Ziel des Szenarios 1 erreicht, somit wird das Szenario 1 beendet und der NAO begibt sich in den Idle-Zustand zurück.

Validierungsergebnis

Die aus den Validierungsabläufen resultierenden Ergebnisse werden anhand der Validierungskriterien analysiert. Die Analyse zeigt, ob die Validierungskriterien erfüllt worden sind.

Teppichboden: Bei allen zehn Abläufen auf dem Bodenbelag, welcher starke Unebenheiten in der Struktur aufweist, gerät der NAO ständig aus dem Gleichgewicht und fällt um. Sobald der NAO die Position ändert oder in starkem Maße die Beingelenke in Anspruch nimmt, sind auffällige Gleichgewichtsstörungen wahrzunehmen.

Linoleum: Auf dem Boden des Raums 11.87 ist die Analyse der zehn Abläufe sehr unterschiedlich ausgefallen. Es wurden drei unterschiedliche Startpositionen im Raum gewählt, auf welche der NAO gestellt und woraufhin dann das System gestartet wurde. Bei dem ersten Startpunkt mit drei Durchläufen war das Ergebnis positiv. Der NAO kam an diesem Punkt nicht aus dem Gleichgewicht und somit auch nicht zu Fall. Bei den weiteren beiden Startpunkten mit insgesamt sieben Durchläufen ist das Ergebnis negativ ausgefallen. Der NAO konnte an diesen Punkten sein Gleichgewicht nicht halten und ist beim Ablauf ständig hingefallen. Beim letzten Durchlauf wurde der Torso des NAO vom Nutzer stabilisiert, was zu einem erfolgreichen Ablauf führte. Der Erfolg kam dadurch zustande, dass der Massenmittelpunkt des NAO in der Ebene stabilisiert wurde.

Holz (Tisch): Der zuvor geschilderte Validierungsablauf wurde auf dem Tisch durchgeführt, was dessen Erfolg gewährleistet hat. Bei allen zehn Durchläufen hat der NAO die implementierte Choreographie komplett durchlaufen, ohne zu fallen. Alle Interaktionen wurden korrekt gestartet und die Anweisungen korrekt entgegengenommen.

Schlussfolgernd benötigt das Szenario 1 eine Oberfläche, die eben ist, damit der NAO die Choreographie erfolgreich ausführt. Jeder einzelne Validierungsablauf konnte ohne ein erneutes Laden des Akkus durchgeführt werden. Zudem konnte der NAO eine Aufforderung an den Nutzer stellen und dessen Antwort korrekt erfassen sowie auswerten.

Der Validierungsablauf auf dem Tisch konnte ohne Probleme durchgeführt werden, die Validierungskriterien sind erfolgreich eingehalten worden und können somit als erfolgreich angesehen werden.

6.2. Validierung - Szenario 2

Das zweite Szenario ist ein gestenbasiertes Spiel („Schere, Stein, Papier“). Hier duelliert sich der Nutzer mit dem NAO. Sobald einer der Kontrahenten drei Spielrunden gewonnen hat, ist das Spiel beendet.

Vorbedingungen

Für das Szenario 2 müssen folgende Vorbedingungen für den Validierungsablauf herrschen: Der Raum wird mit unterschiedlichem Licht (Tageslicht, Deckenbeleuchtung) beleuchtet. Somit kann die Kamera auf deren Sensibilität hin validiert werden. Für den Nutzer werden drei Symbolbilder angefertigt, welche die Formen Schere, Stein und Papier festhalten. Der Nutzer

ist dazu verpflichtet, anstelle seiner Hand die Symbolbilder zu nutzen. Es finden jeweils fünf Abläufe für alle drei Strategien statt.

Validierungskriterien

Auf folgende Kriterien wird beim Validierungsablauf getestet:

- Aufnahme der gespielten Symbolbilder in einer Spielrunde
- Korrekte Auswertung einer Spielrunde
- Validierung der drei Strategien auf Richtigkeit der Logik
- Validierung auf Richtigkeit des Spielablaufs und der Beendigung

Validierungsablauf

Der folgende Validierungsablauf wird anhand der Strategie „Easy“ (vgl. Kapitel 4.5.2) beschrieben. Der Raum wird von der Deckenbeleuchtung künstlich beleuchtet. Alle weiteren Strategien werden im Validierungsergebnis erörtert.

Bei der korrekten Auswertung der Identifikationsnummer „2“ startet das Szenario 2. Als Erstes wird der NAO durch die Initialisierung in eine schonende Sitzposition gebracht. Daraufhin folgt eine Interaktion des NAO mit dem Nutzer, der die Schwierigkeitseinstellung auswählt. Bei diesem Validierungsablauf werden die Spielrunden mit der Strategie „Easy“ gespielt. Der Nutzer muss dafür den Identifikationsbuchstaben „A“ laut aussprechen. Der NAO hält daraufhin die Strategie für den Spielverlauf fest und startet das Spiel.

Bevor die Synchronisationsbewegung startet, wählt der NAO seine Strategie, mit der er in der aktuellen Spielrunde gegen seinen Kontrahenten antritt. Nach deren erfolgreicher Wahl startet der NAO mit der Synchronisationsbewegung und macht schließlich seine Strategie für den Kontrahenten sichtbar, indem die jeweilige Geste mit der Hand imitiert wird. Der NAO wählt die Schere. Der Nutzer wählt parallel zur Synchronisationsbewegung eines der regulären Symbolbilder aus und zeigt dieses dem NAO. Der Nutzer spielt Stein. Ist die Strategie erfasst worden, kann der NAO anhand der Auswertungsmatrix den Gewinner der ersten Spielrunde korrekt auswerten und benennen. Sobald die Auswertung stattgefunden hat, wird dieses dem Nutzer mithilfe des TextToSpeech-Moduls verbal über das Soundsystem des NAO mitgeteilt. Die Auswertung der Spielrunde ist korrekt ermittelt und mitgeteilt worden.

Der NAO startet eine neue Spielrunde, da zu diesem Zeitpunkt noch keiner der Mitspieler drei Spielrunden gewonnen hat. Der NAO wählt auch in dieser Runde das Symbol „Schere“ aus. Der Nutzer wählt „Papier“. Auch bei dieser Runde ist die Auswertung korrekt erfolgt. Der Punktestand beträgt somit 1 : 1.

Bei den nächsten beiden Spielrunden hat der NAO – wie bei der Implementierung immer dasselbe Symbol gewählt. Der Nutzer nutzt sein Wissen um dieses Erscheinungsbild und kann daraufhin beide Runden für sich entscheiden. Somit beträgt der Punktestand 1 : 3. Der NAO bedankt sich beim Nutzer für den Spielverlauf und hat korrekt erkannt, dass das Spiel beendet werden muss.

Validierungsergebnis

Das Validierungsergebnis zeigt, ob die Validierungskriterien erfüllt worden sind.

Tageslicht: Bei allen Validierungsabläufen, bei denen der Raum mit starkem Tageslicht beleuchtet ist, hat der NAO große Probleme, die Symbolbilder zu erkennen. Oft wurden die Symbolbilder von ihm falsch erfasst. Nachdem durch Zuziehen der Gardinen die Lichtverhältnisse optimiert worden sind, ist dem NAO immer eine korrekte Auffassung der Symbolbilder gelungen. Jede implementierte Strategie trägt in ihrer eigenen Logik korrekt zur Auswertung der Spielrunde bei. Das System erkennt bei jeder Strategie die Beendigung eines Spieles.

Deckenbeleuchtung: Bei den Validierungsabläufen mit der Deckenbeleuchtung hat der NAO alle Symbolbilder erfolgreich erkannt. Die Auswertung der einzelnen Strategien bezüglich der Symbolbilder war auch immer korrekt.

Das Szenario 2 erfüllt somit bei allen 15 Validierungsabläufen alle Validierungskriterien und wird somit als erfolgreich angesehen. Zu starke Lichteinstrahlung im Raum manipuliert das Kamerasystem bei der Erfassung der Symbolbilder und sollte daher reduziert werden.

6.3. Validierung - Szenario 3

Das dritte Szenario ist ein kommunikatives Spiel, welches als Quiz-Show realisiert worden ist. Der Nutzer nimmt hierbei die Rolle des Kandidaten ein und der NAO die des Moderators. Der Moderator stellt dem Kandidaten bis zu acht Fragen. Das Spiel wird beendet, sobald der Kandidat eine Frage falsch oder alle acht Fragen korrekt beantwortet hat.

Vorbedingungen

Die Vorbedingungen für das kommunikative Szenario beziehen sich auf die Interaktionen. Für die Aufnahme muss der Geräuschpegel im Raum so niedrig wie möglich gehalten werden. Es werden fünf Validierungsabläufe durchgeführt.

Validierungskriterien

Das Szenario muss folgende Validierungskriterien einhalten:

- Verständliche Fragestellung an den Nutzer
- Zu jeder Frage vier Antwortmöglichkeiten zur Auswahl stellen
- Korrekte Aufnahme der Antworten des Nutzers
- Maximal drei Joker-Nutzungen im gesamten Spielverlauf
- Zeitliche Begrenzung von 30 Sekunden, um auf eine Frage zu antworten

Validierungsablauf

Wurde das Szenario vom System anhand der Identifikationsnummer „3“ gestartet, begibt sich der NAO in die Initialisierungsstellung. Hat er diese Position eingenommen, stellt der NAO dem Nutzer die erste von insgesamt acht Fragen. Zu jeder Frage werden vier Antwortmöglichkeiten vorgestellt. Der Nutzer hat dem NAO beim ersten Durchlauf bei der ersten Frage innerhalb der Zeitvorgabe keine Antwort mitgeteilt. Der NAO bedankt sich daraufhin für den Spielverlauf und beendet das Spiel wie bei der Implementierung. Beim zweiten Durchlauf wurde die erste Frage vom Nutzer falsch beantwortet. Auch hier reagiert das Szenario ordnungsgemäß und der NAO beendet das Spiel. Beim nächsten Durchlauf wurde die Nutzung eines Jokers zum Erreichen der nächsten Frage genutzt. Die Nutzung des Jokers sorgt dafür, dass von vier Antwortmöglichkeiten eliminiert werden. Der Nutzer hat daraufhin die korrekte Antwort ausgewählt und ist dadurch zur nächsten Frage gelangt. Die kommenden sieben Fragen wurden ebenfalls korrekt beantwortet. Der NAO gratuliert dem Nutzer abschließend zum Sieg und beendet daraufhin das Spiel.

Validierungsergebnis

Der Validierungsablauf wurde bei unterschiedlichem Lärmpegel aufgenommen, um die Interaktionen zu beobachten.

Flüstern: Bei einem Durchlauf, bei dem der NAO und der Nutzer allein im Raum sind (30 dB), kam es zu keinem Missverständnis – weder von der Aufnahme der Fragen noch von der Aufnahme der Antworten her.

Normales Gespräch: Auch die Durchläufe bei einem niedrig gehaltenen Lärmpegel (60 dB) wertet das System bei allen Antwortmöglichkeiten des Nutzers korrekt aus. Der Nutzer musste die Antworten bei einigen Fragen jedoch wiederholt eingeben, da sie der NAO als Hintergrundgeräusch wahr genommen hat. Es wurde keine Antwort aus den Hintergrundgeräuschen herausgefiltert, sondern nur die vom Nutzer akzeptiert.

Gruppengespräch: Bei den Durchläufen, bei denen der Lärmpegel stark erhöht war (75 dB), zeigt das System Instabilität. Es kommt zu Zeitüberschreitungen trotz kontinuierlicher Antworten. Zudem kommt es zu einer Fehlaufnahme der Antworten. Des Weiteren erkennt der NAO Antworten aus den Gesprächen der Gruppen.

Das Szenario wird somit (nur) in Räumen als erfolgreich angesehen, in denen der Lärmpegel niedrig gehalten wird. Nur dann können alle Validierungskriterien eingehalten werden.

6.4. Validierung - Szenario 4

Das Szenario vier ist das interaktive Szenario. Dem Nutzer wird ein unvollständiger Tanz vorgeführt, welchen er vollenden soll, indem die fehlenden Tanzschritte implementiert werden. Für das interaktive Szenario empfiehlt es sich eine Evaluation mit Probanden durchzuführen, um ein akzeptables Test als erfolgreich anzunehmen.

Vorbedingungen

Die Vorbedingungen für das interaktive Szenario beziehen sich auf die Gegebenheiten, um das Szenario interaktiv zu verfassen. Zudem muss der Nutzer die fehlenden Tanzschritte zuerst identifizieren. Alle Hauptbewegungen werden dem Nutzer mittels Bilder mitgegeben, was eine Identifizierung erleichtert. Ein Computersystem muss gestellt werden, auf dem die Software *Choregraphe* installiert ist

Validierungskriterien

Die Validierungskriterien sind aus dem Szenario 1 zu entnehmen und Validierung auf die Systemvision.

Validierungsablauf

Der folgende Validierungsablauf wurde mit den Kommilitonen Davut Kuru am 12. Juli.2016 vorgenommen. Es wurde schnell erkannt, welche Bewegungen im Tanz fehlten. Durch die gute Bedienbarkeit von Choregraphe war es für Herr Kuru ein leichtes die fehlenden Bewegungen schnell aufzunehmen und zu Simulieren.

Validierungsergebnis

Für ein besseres Ergebnis muss für das Szenario 4 eine separate Evaluation stattfinden indem Probanden eingeladen werden und sich trauen interaktiv den Tanz neu zu Gestalten. Dadurch dass Herr Kuru technisches Know-how hat, war dies keine Hürde um zum Erfolg zu kommen.

6.5. Anforderungsabgleich

Nachfolgend wird tabellarisch aufgeführt, welche funktionalen, technischen und nicht-funktionalen Anforderungen im Konzept und bei der Realisierung erfüllt wurden.

Anforderung	Konzept	Realisierung
S1/S2/S3/S4-FA-1 (Nutzer registrieren, Auswählen der Szenarien)	✓	✓
S1-FA-2 (Auswählen der Schlusschoreographie)	✓	✓
S2-FA-2 (Spieleinstellung hinzufügen)	✓	✓
S2-FA-3 (Symbolbilder wählen und zeigen)	✓	✓
S2-FA-4 (Symbolbilder erfassen, auswerten und speichern)	✓	✓
S2/S3-FA-5 (Beendigung des Szenario)	✓	✓
S3-FA-2 (Antwortmöglichkeit wählen)	✓	✓
S3-FA-3 (Frage stellen und auswerten)	✓	✓
S3-FA-4 (Joker zu Verfügung stellen)	✓	✓
S4-FA-2 (Aufnahme der fehlenden Tanzschritte mit Choregraphie)	✓	✓

Tabelle 6.1.: Erfüllung der funktionalen Anforderung in der Konzeptrealisierung.

Die Tabelle 6.1 zeigt, dass alle funktionalen Anforderungen von den Szenarien der interaktiven Präsentation beim Konzept berücksichtigt und bei der Realisierung implementiert wurden. Die Anforderungen S4-TA-1, S4-TA-2 und S4-TA-3 sind bei der Nutzung der Präsentation nötig, um das Szenario 4 erfolgreich abzuschließen. Sie können beim Validierungsablauf nicht überprüft werden, da das Computersystem nicht bereitgestellt worden ist, mit dem die spätere

6. Systemvalidierung

Anforderung	Konzept	Realisierung
S1-TA-1 (Audiodateien wiedergeben)	✓	✓
S1-TA-2, S2-TA-2, S3-TA-1 (Anforderungen aufnehmen)	✓	✓
S1-TA-3 (Nutzung aller DoF)	(✓)	✓
S2-TA-1 (Visuelles wahrnehmen der Symbolbilder)	✓	✓
S4-TA-1 (Ausführung auf Computersystemen mit der Entwicklungsumgebung "Choregraphie")	-	-
S4-TA-2 (Choregraphie Versionierung)	-	✓
S4-TA-3 (Betriebssystem des Computersystems)	-	-

Tabelle 6.2.: Erfüllung der technischen Anforderung in der Konzeptrealisierung.

Präsentation ausgeführt wird.

S4-TA-2 ist in der Realisierung als erfolgreich angenommen worden, da die Implementierung der Präsentation mit dieser Choregraphie-Version umgesetzt wurde.

S1-TA-3 beschreibt, dass bei der Implementierung alle **DoF** in der Choreographie einmal genutzt werden müssen. Bei der Konzeptionierung wurde nur beschrieben, wie **DoF** angesprochen und implementiert werden.

Anforderung	Konzept	Realisierung
S1-NFA-1 (Reaktionszeit)	✓	✓
S1-NFA-2, S2-NFA-1, S3-NFA-1 (Zeitlichtbegrenzung nach Anforderungen)	✓	✓

Tabelle 6.3.: Erfüllung der nicht-funktionalen Anforderung in der Konzeptrealisierung.

Die Tabelle 6.3 zeigt, dass alle nicht-funktionalen Anforderungen von den Szenarien der interaktiven Präsentation beim Konzept berücksichtigt und bei der Realisierung implementiert wurden.

6.6. Hardwarebeschränkung des NAO

Während der Validierungsabläufe sind Versuche an bestimmten Hardwarekomponenten vorgenommen worden, woraus sich die nachfolgenden Erfahrungen mit dem NAO ergeben haben.

Kamera

Bei der Nutzung der Kamerakomponente sind folgende spezifische Probleme bei der Versuchsdokumentation wahrgenommen worden:

Bei der Nutzung des Kamerasystems traten kurze Bildverluste auf, welche bei der Verwendung zur Nicht-Erkennung der Objekte (z. B. der Symbolbilder) führten. Dies machte sich bei der Versuchsdokumentation des zweiten Szenarios bemerkbar, da dem NAO hierbei eine Zeitvorgabe zur Erkennung eines Objektes gesetzt worden ist. Eine weitere Herausforderung, um Objekte wiederzuerkennen, stellt ein sonnendurchfluteter Raum dar. Das Kamerasystem ist gegenüber starken Lichtverhältnissen hoch empfindlich. Durch das Zuziehen der Gardinen ist eine starke Verbesserung bei der Objekterkennung im sonnendurchfluteten Raum entstanden.

Akkulaufzeit

Bei einer kontinuierlichen und langen Nutzung der Präsentation ist die Akkuleistung nicht ausreichend. Die Inbetriebnahme des NAO sollte bevorzugt ohne Ladekabel in Anspruch genommen werden, da das Kabel einige Bewegungsabläufe verhindert und der NAO in Gefahr ist, umzufallen. Es empfiehlt sich, das Laden des Akkus während des Idle-Zustands des NAO vorzunehmen.

Gelenküberhitzung

Ein starkes Manko betrifft die Motorik der einzelnen DoF. Die schnelle Überhitzung der Motorik des NAO schränkt die Präsentation der realisierten Bewegungsabläufe ein. Versuchsdurchführungen, bei denen Bewegungsabläufe stattfanden, konnten meist (nur) 20 Minuten lang getestet werden, bevor eine Überhitzung stattfand. Oft fand eine Überhitzung schon nach 15 Minuten statt. Der NAO muss bei einer Überhitzung der Gelenke umgehend ausgeschaltet werden, um diese abzukühlen, damit an der Motorik keine Beschädigung entsteht.

6.7. Umfeld

Im kommenden Abschnitt wird ein Setup für eine hohe Performance-Gewinnung vorgestellt, welche aus den Validierungsabläufen aller Szenarien abgeleitet wurde.

Für einen idealen Ablauf befinden sich der Nutzer, der Betreuer und der NAO in einem neutral gehaltenen Raum. Als neutral wird definiert, dass die Wände idealerweise in Weiß gestrichen sind, der Raum mit künstlichem Licht beleuchtet und der Lärmpegel des Raumes niedrig gehalten wird. Der Raum benötigt eine Stromversorgung, damit der Akku des NAO im Idle-Zustand geladen werden kann. Zudem sollte sich ein Stuhl in dem Raum befinden, wo der Nutzer in einem kurz gehaltenen Sicherheitsabstand von mindestens 60 cm zum NAO Platz nimmt. Der NAO wird auf einem Tisch (min. Arbeitsfläche 60x60 cm) gegenüber dem Nutzer aufgestellt. Der Betreuer hält sich möglichst nah beim NAO und beim Nutzer auf, um Schäden vorbeugen zu können, falls der NAO stürzt oder um dem Nutzer bei den Interaktionen zu helfen. Die Abbildung 6.1 hält den Aufbau für einen idealen Ablauf der Szenarien fest.

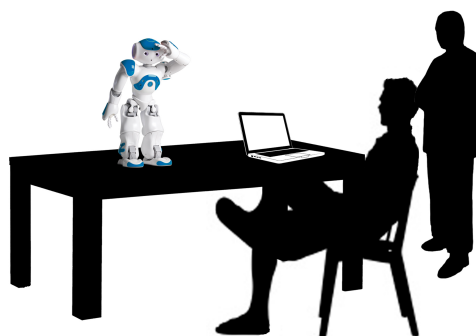


Abbildung 6.1.: Versuchsaufbau für eine hohe Performance-Gewinnung.

6. Systemvalidierung

In diesem Kapitel wurden die Anforderungen der Szenarien mit dem Konzept und der Realisierung abgeglichen und unter realen Bedingungen validiert.

7. Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, durch vier Szenarien die Mensch-Maschine-Interaktion auf Basis der NAO-Plattform zu realisieren sowie einen technischen und fachlichen Durchbruch in der Entwicklung des NAO zu schaffen. Die vier Szenarien waren folgende: „Tanzszenario“, „Schere, Stein, Papier“, „Quiz-Spiel“, „Tanzszenario erweitern“.

In den Grundlagen erfolgte zuerst eine Einführung in Bezug auf den humanoiden Roboter NAO. Hierfür wurden die NAO-Roboterfamilie und deren unterschiedliche kinematische Körper vorgestellt. Weiterhin wurde der NAO-H25 im Zusammenhang mit seinen für die Thesis relevanten technischen Daten erörtert. Zudem erfolgte die Betrachtung der Entwicklungssoftware des NAO, damit eigenständige Applikationen entwickelt werden können. Hierfür wurden die Softwarearchitektur und die Hauptsoftware „NAOqi“ näher betrachtet. Verdeutlicht wurden hierbei das Schlüsselkonzept zu den Methodenaufrufen und eine Übersicht über alle Module, die genutzt werden können.

Vor der Analyse des Systems wurde eine Systemvision vorgestellt. Sie beschreibt die Szenarien-Auswahl. Anschließend wurde anhand eines UML Anwendungsfalldiagramms die Funktionalität des Systems erläutert. Szenarienübergreifend und -spezifisch wurden in den Unterkapiteln tabellarisch Anwendungsfallbeschreibungen und konkrete Anforderungen an das System gestellt, um das System abzugrenzen. Szenarienspezifisch wurde jeweils ein Aktivitätsdiagramm konstruiert, um den Verlauf des Prozesses sowie Alternativabläufe der Szenarien darzustellen.

Im nächsten Schritt wurde in Kapitel 4 das System entworfen. Zuerst fand eine Evaluierung zur Modulprogrammierung statt, um die gewünschten Module konsequent und effektiv zu implementieren. Darauffolgend wurden die Hardwarekomponenten und die dazugehörigen Module identifiziert und näher erörtert. Szenarienspezifisch wurde in den Unterkapiteln ein fachliches Datenmodell vorgestellt. Aus dem fachlichen Datenmodell konnte ein Komponentenschnitt vorgestellt und das resultierende Komponentendiagramm abgeleitet werden. Letztlich modelliert das Sequenzdiagramm die Interaktionen zwischen dem Nutzer und dem NAO wäh-

rend des Ablaufs der vier Szenarien.

Die im Rahmen der vorliegenden Arbeit entwickelten Konzepte stellen eine akzeptable Grundlage für die Realisierung der Szenarien dar. Die Nutzung vieler vorgefertigter Modul-Methoden in Kombination mit Python-Skripten vereinfachte die Realisierung. Diese Vereinfachung wurde genutzt, um die vier Szenarien mit möglichst wenig Aufwand zu erstellen und im zeitlichen Rahmen einer Bachelor-Arbeit eine funktionsfähige Präsentation für die Außendarstellung abzuliefern.

Anhand von Validierungsabläufen wurde sichergestellt, dass die Tragfähigkeit der Anforderungen im Konzept konzeptioniert und in der Realisierung implementiert worden sind. Ebenso wird deutlich, dass die Szenarien erfolgreich implementiert worden sind. Für die Abläufe wurden im Vorfeld jeweils relevante Vorbedingungen und Kriterien aufgestellt und überprüft.

Die entwickelten Konzepte mit ihren vier Szenarien stellen eine akzeptable Grundlage für die Realisierung der Szenarien dar. Die Szenarien halten fest, dass Mensch-Maschine-Interaktionen erfolgreich auf den Roboter NAO der Firma Aldebaran Robotics übertragbar sind. Jedoch sind diesbezüglich bereits jetzt Verbesserungen möglich: Mithilfe der Bildverarbeitungsbibliothek OpenCV besteht die Möglichkeit einer besseren Bilderkennung und von deren Auswertung. Durch die Einbindung dieser Bildverarbeitungsbibliothek kann die Weiterentwicklung des Szenarios 2 „Schere, Stein, Papier“ sinnvoll sein: Zum einen kann der Besucher dann auf die vorliegend zur Verfügung gestellten Symbolbilder verzichten und seine eigene Hand zur Strategieauswahl nutzen, zum anderen wäre (dadurch) der Spaßfaktor beim Spielen größer.

Um die verfasste Systemvision zu testen, empfiehlt es sich, eine Nutzerstudie durchzuführen, indem insgesamt 10-15 junge und ältere Probanden eingeladen werden, um die Szenarien durchzuspielen. Hierbei kann mittels einer Evaluierung festgestellt werden, ob die einzelnen Szenarien der Systemvision gerecht werden oder gegebenenfalls im Hinblick auf die Systemvision anzupassen sind. Ein weiterer Evaluationspunkt ist es, ob die Interaktionsmöglichkeiten von NAO, die sprach- sowie gestenabhängig sind, bei den Nutzern gut ankommen oder ob sie zu künstlich übermittelt werden. Dies sind Faktoren, wodurch die Systemvision an Freude einbüßen kann und schlimmstenfalls Langeweile vermittelt. Das interaktive Szenario 4 („Tanzszenario erweitern“), welches in der Systemvision eine entscheidende Rolle hat, indem es (im Idealfall) zur Stärkung des Interesses von Schülerinnen an der Informatik dient, könnte mit der Evaluierung ebenfalls ersichtlich werden.

7. Zusammenfassung und Ausblick

Trotz der vereinfachten Akzeptanz konnte am Beispiel des NAO in dieser Arbeit gezeigt werden, dass eine Mensch-Maschine-Interaktion erfolgreich zu realisieren ist. Mit dem Ergebnis dieser Thesis kann die im Rahmen dieser entwickelte Präsentation zur Außendarstellung der Hochschule für Angewandte Wissenschaften Hamburg genutzt werden.

Literatur

- [1] Aldebaran Robotics. *NAO T2*. URL: http://doc.aldebaran.com/2-1/family/nao_t2/index_t2.html (besucht am 07.09.2016).
- [2] Aldebaran Robotics. *NAO H25*. URL: http://doc.aldebaran.com/2-1/family/nao_h25/index_h25.html (besucht am 07.09.2016).
- [3] The Robocup Federation. *Standard Platform* « *RoboCup*. 1998 - 2016. URL: <http://robocup.org/robocup-soccer/standard-platform> (besucht am 16.07.2016).
- [4] Aldebaran Robotics. *NAO: Humanoid Robot Platform*. 2012. URL: https://www.aldebaran.com/sites/aldebaran/files/datasheet_ao_next_gen_en.pdf (besucht am 16.07.2016).
- [5] David Gouaillier u. a. "Mechatronic design of NAO humanoid". In: *2009 IEEE International Conference on Robotics and Automation*. Institute of Electrical & Electronics Engineers (IEEE), 2009. DOI: [10.1109/robot.2009.5152516](https://doi.org/10.1109/robot.2009.5152516). URL: <http://dx.doi.org/10.1109/ROBOT.2009.5152516>.
- [6] Aldebaran Robotics. *Aldebaran documentation v2.1.4.13 - Motors*. 2015-10-06. URL: http://doc.aldebaran.com/2-1/family/robots/motors_robot.html (besucht am 18.07.2016).
- [7] Aldebaran Robotics. *Datasheet NAO H21/H25 Model*. November 2016. URL: https://www.aldebaran.com/sites/aldebaran/files/datasheet_ao_next_gen_en.pdf (besucht am 07.09.2016).
- [8] Aldebaran Robotics. *Programming*. URL: http://doc.aldebaran.com/2-1/dev/programming_index.html#dev-index (besucht am 07.09.2016).
- [9] Aldebaran Robotics. *Aldebaran documentation v2.1.4.13 - NAOqi*. 2015-10-06. URL: <http://doc.aldebaran.com/2-1/dev/naoqi/index.html> (besucht am 06.08.2016).

- [10] Markus Wessler. *Entscheidungstheorie*. Springer Science mathplus Business Media, 2012. DOI: [10.1007/978-3-8349-3734-6](https://doi.org/10.1007/978-3-8349-3734-6). URL: <http://dx.doi.org/10.1007/978-3-8349-3734-6>.
- [11] C. Kecher und A. Salvanos. *UML 2.5: Das umfassende Handbuch*. Galileo Computing. Rheinwerk Verlag GmbH, 2015. ISBN: 9783836229777.
- [12] Johannes Siedersleben, Hrsg. *Moderne Software-Architektur: Umsichtig planen, robust bauen mit Quasar*. Heidelberg: dpunkt, 2004. ISBN: 978-3-89864-292-7.

Anhang

A. Aktivitätsdiagramm

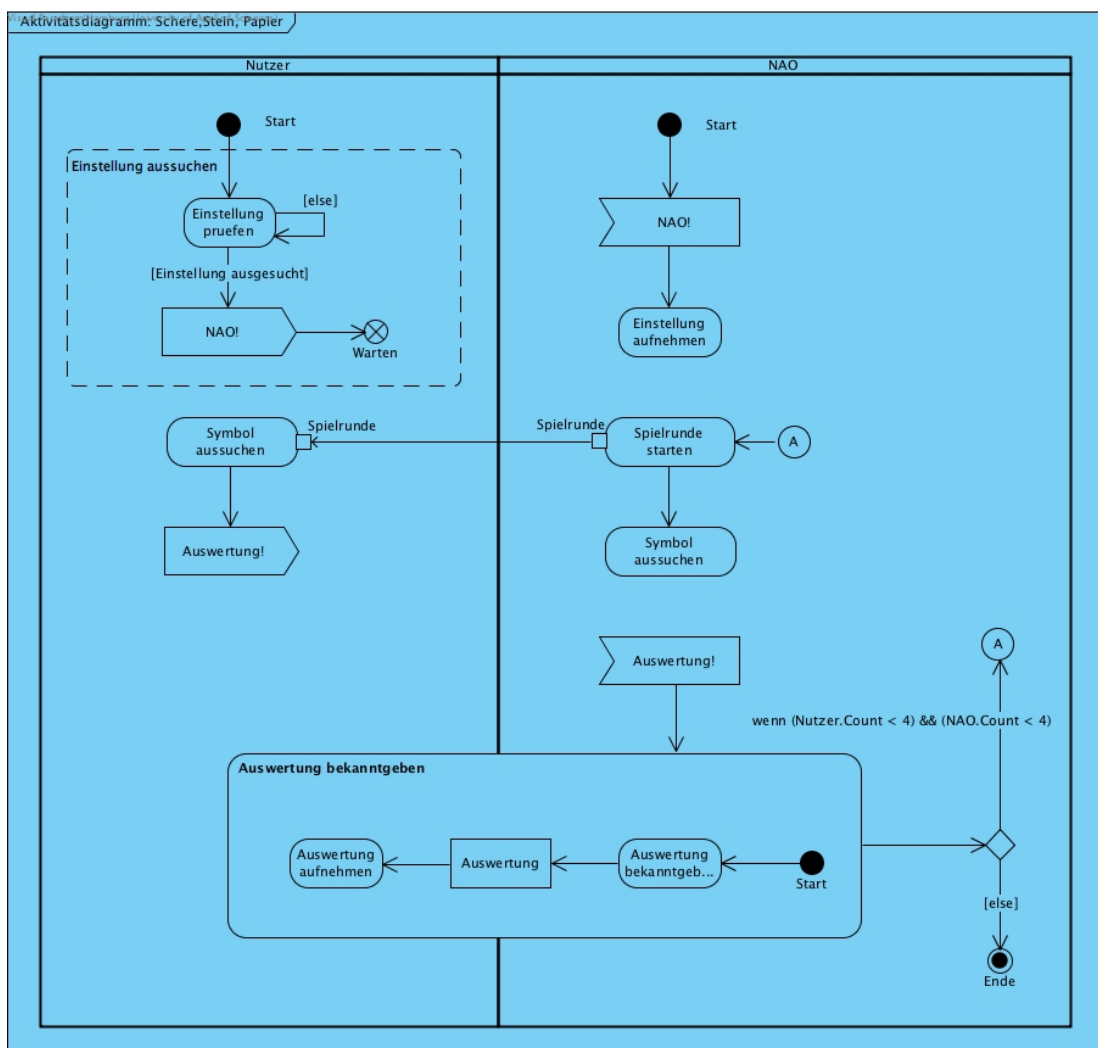


Abbildung 1.: Aktivitätsdiagramm: Schere, Stein, Papier

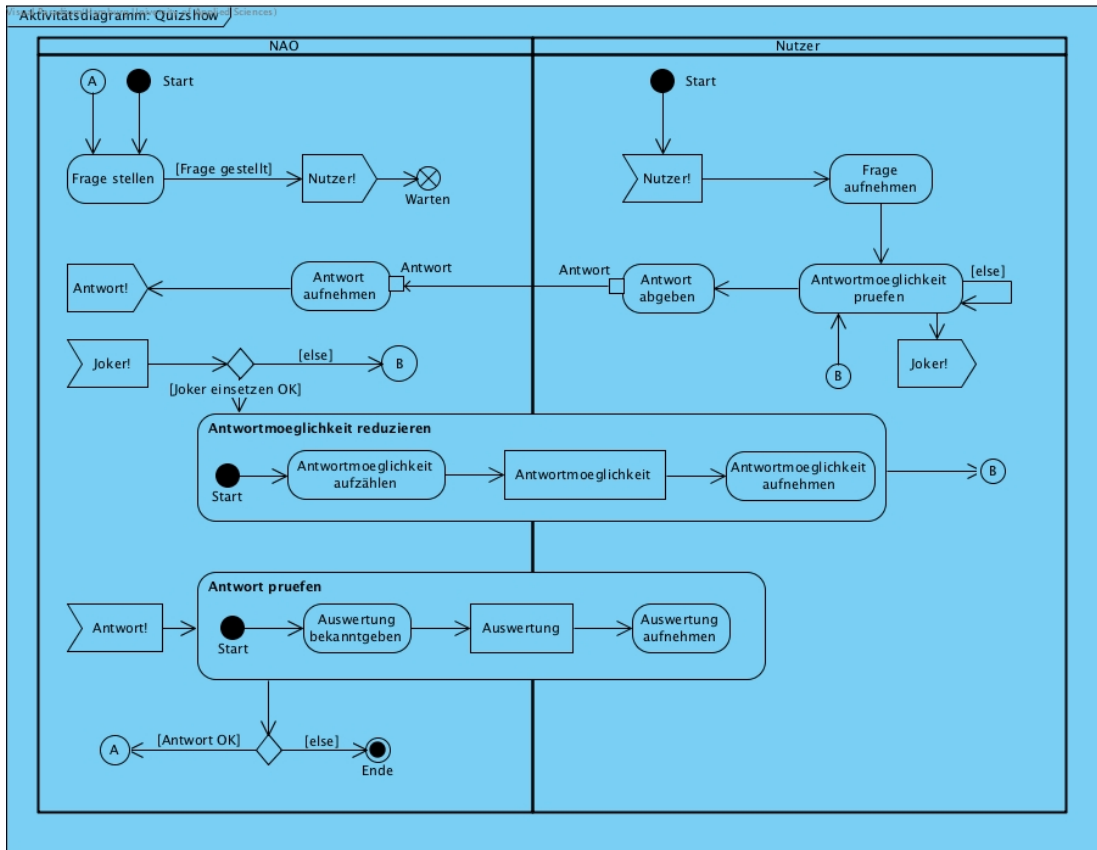


Abbildung 2.: Aktivitätsdiagramm: Quizshow

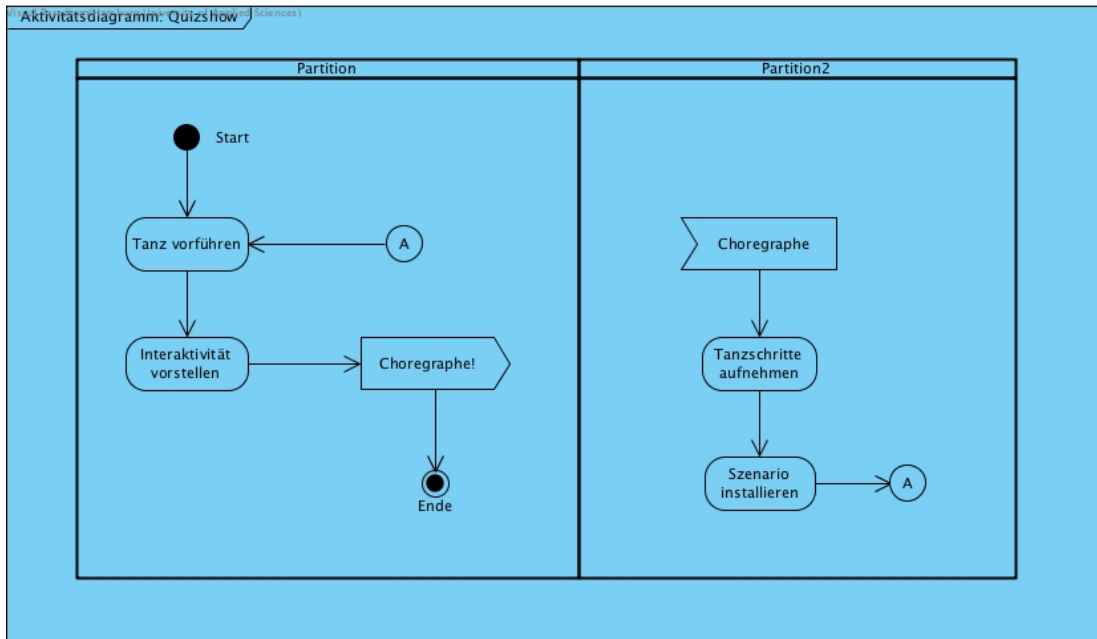


Abbildung 3.: Aktivitätsdiagramm: Tanzszenario erweitern

B. Anwendungsfälle

Strategie wählen

Name	Strategie wählen
Kurzbeschreibung	Der Nutzer setzt den Schwierigkeitsgrad für den gesamten Spielverlauf beim NAO ein.
Verwendete Anwendungsfälle	-
Erweiterungspunkt	-
Akteur	Nutzer
Vorbedingung	NAO hat das Spiel Schere, Stein, Papier gestartet.
Ergebnis	NAO setzt die Strategie für das gesamte Spielverlauf.
Nachbedingung	NAO startet den Spielverlauf.
Standard Ablaufschritte	<ol style="list-style-type: none"> 1. Der Nutzer wählt die Einstellung Easy anhand der Identifikationsbuchstaben A. 2. NAO setzt entsprechend der Identifikationsbuchstaben den Schwierigkeitsgrad ein. 3. Nach erfolgreichem setzen des Schwierigkeitsgrades startet NAO die Spielrunde.
Alternative Ablaufschritte	<ol style="list-style-type: none"> 1.1. Der Nutzer wählt die Einstellung Medium anhand des Identifikationsbuchstaben "B". 1.2. Der Nutzer wählt die Einstellung Hard anhand des Identifikationsbuchstaben "C".

Tabelle 1.: Anwendungsfallbeschreibung: "Strategie wählen".

Joker einsetzen

Name	Joker einsetzen
Kurzbeschreibung	Der Nutzer wählt für die aktuelle Frage einen Joker aus.
Verwendete Anwendungsfälle	-
Erweiterungspunkt	Spielszenario auswählen Quizshow.
Akteur	Nutzer
Vorbedingung	NAO hat eine Frage mit den Antwortmöglichkeiten gestellt.
Ergebnis	NAO reduziert die Antwortmöglichkeiten.
Nachbedingung	-

Standard Ablaufschritte	1. Der Nutzer wählt die Option Joker. 2. NAO reduziert die Antwortmöglichkeit und teilt sie dem Nutzer mit.
Alternative Ablaufschritte	2.1 NAO lehnt das nutzen eines Joker ab, da der Nutzer alle zu Verfügung gestellten Joker bereits eingesetzt hat.

Tabelle 2.: Anwendungsfallbeschreibung: "Joker einsetzen".

Interaktivesszenario erweitern

Name	Interaktivesszenario erweitern
Kurzbeschreibung	Dem Nutzer steht die Möglichkeit den Tanz zu Erweitern.
Verwendete Anwendungsfälle	-
Erweiterungspunkt	Interaktivesszenario erweitern.
Akteur	Nutzer
Vorbedingung	NAO hat das Szenario erfolgreich vor getanzt.
Ergebnis	Der Nutzer hat die fehlenden Tanzschritte aufgezeichnet.
Nachbedingung	Der Nutzer wählt erneut das Szenario beim NAO.
Standard Ablaufschritte	1. Der Nutzer zeichnet die fehlenden Bewegung mit der IDE im Animationsmodus auf. 2. Der Nutzer überprüft zunächst im Simulator, ob die gewünschten Bewegungen akzeptable sind. 3. Der Nutzer installiert die interaktive Präsentation auf dem NAO.
Alternative Ablaufschritte	-

Tabelle 3.: Anwendungsfallbeschreibung: "Interaktivesszenario erweitern".

C. Evaluation

Programmierung anhand Choregraphie

Der Grünmarkierter Bereich heißt flow diagram panel (deutsch Flussdiagramm). Dieser Bereich dient dem Entwickler zum erstellen von behaviours. Ein flow diagram besteht aus einer Menge von Boxen die graphisch mit nur einem Eingang verbunden sind. Als erste Ebene wird die Root-Ebene bezeichnet in der das behaviour als Boxen festgehalten wird. Jede Box stellt eine unterliegende Ebene dar, dass selbst ein eigenes flow diagram besitzt in der ein Unikats Verhalten erstellt werden kann. Das obige Beispiel Programm in der Abbildung XX soll verbal über seinem Soundsystem und durch Winken des Armes eine Begrüßung Szenario wiedergeben. Bei der Ausführung des Programmes wird ein Start-Signal in die erste Box gesendet. In diesem ist das Ziel den NAO ohne Ihn aus dem Gleichgewicht zu bringen oder zu beschädigen, in die Initialstellung zu bewegen, unabhängig in welcher Position sein letzterer Zustand war. Wurde diese Animation erfolgreich oder erfolglos abgeschlossen, so wird ein Signal ausgehend in die nächste Box geführt und dient für das Starten. Der Hauptbehavior, welche die Begrüßung ist, soll nach Beendigung, das letzte Modul anregen den NAO in einer schonenden Sitzposition zu bringen. Damit ein Start sowohl eine Beendigung eines Programmes stattfindet muss das Signal dabei ein vordefinierter Startpunkt und Endpunkt durchlaufen.

Modulprogrammierung (Proxy)

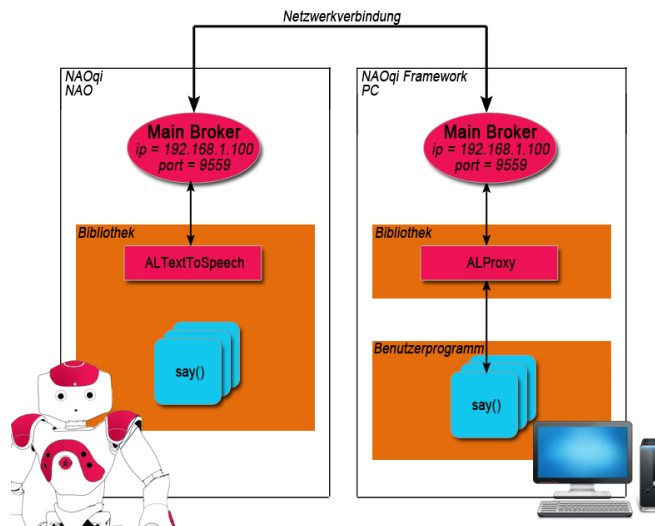


Abbildung 4.: Modul Programmierung: Broker zu Proxy

D. Interaktionsdiagramm

Im folgendem Abschnitt werden die Sequenzdiagramme des Szenario 2 in der Spielstrategie dargestellt. Es sind rein Ausschnitte dargestellt, welche den unterschied der Strategie Zeitlich verdeutlichen soll.

Easy-Strategie

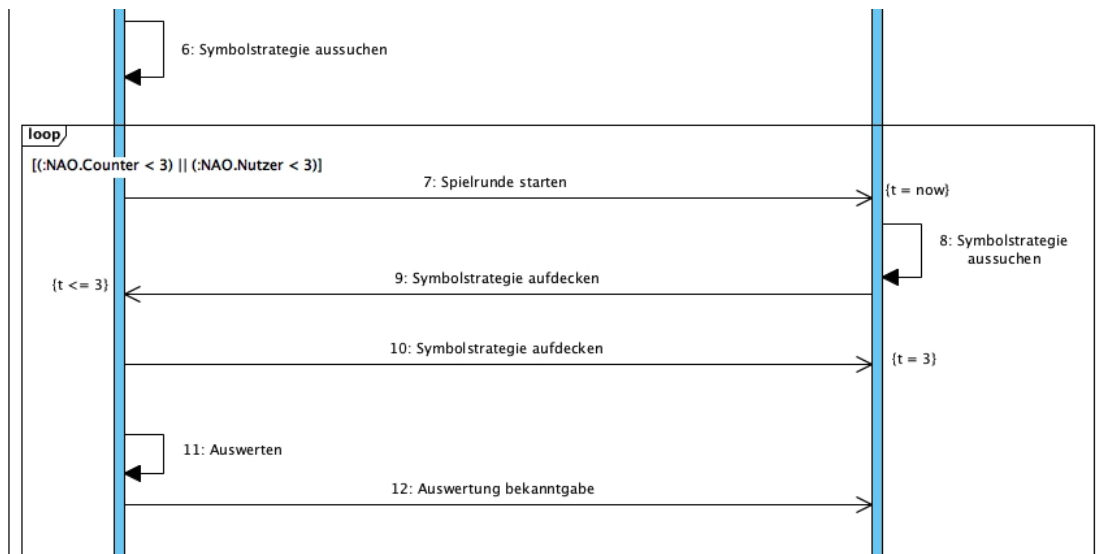


Abbildung 5.: Sequenzdiagramm: Ausschnitt "Easy-Strategie".

Aus der Abbildung 5 wird ersichtlich, dass die Wahl des Symbol vor starten einer Spielrunde gesetzt wird (6). Hat NAO eine Entscheidung getroffen fordert er den Nutzer auf ein Symbol für die Spielrunde zu wählen (7). Dem Nutzer wird eine Zeitspanne vorgegeben indem das Symbol gewählt sowie an NAO präsentiert wird (8, 9). NAO präsentiert sein gewähltes Symbol quasi Zeitgleich zum Nutzer (10). Der NAO erfasst das Symbol des Nutzers, wertet es aus und teilt dies dem Nutzer mit (11, 12). Ist noch kein klarer Sieger hervorgegangen wird eine neue Spielrunde gestartet, indem NAO das selbe Symbol vorzeigt.

Hard-Strategie

Das folgende Sequenzdiagramm beschreibt die Hard-Strategie. Das Interessante ist die zeitliche Wahl des Symbols beim NAO.

NAO leitet zum starten der Spielrunde ein (6). Der Nutzer kann jetzt ein Symbol auswählen mit der es konkurrieren will (7). Der Nutzer präsentiert das gewählte Symbol zum Zeitpunkt des Aufdecken (8). Der NAO wiederum nicht, genau zu diesem Zeitpunkt erfasst NAO das gespielte Symbol des Nutzer und entscheidet sich mit dem Symbol anzutreten das die Spielrunde gewinnt (9). Ist das Symbol anhand der Auswertungsmatrix erfasst präsentiert er es dem Nutzer (10). Der NAO wertet die Spielrunde aus und teilt dieses dem Nutzer mit (11, 12). Ist noch kein klarer Sieger hervorgegangen wird eine neue Spielrunde gestartet.

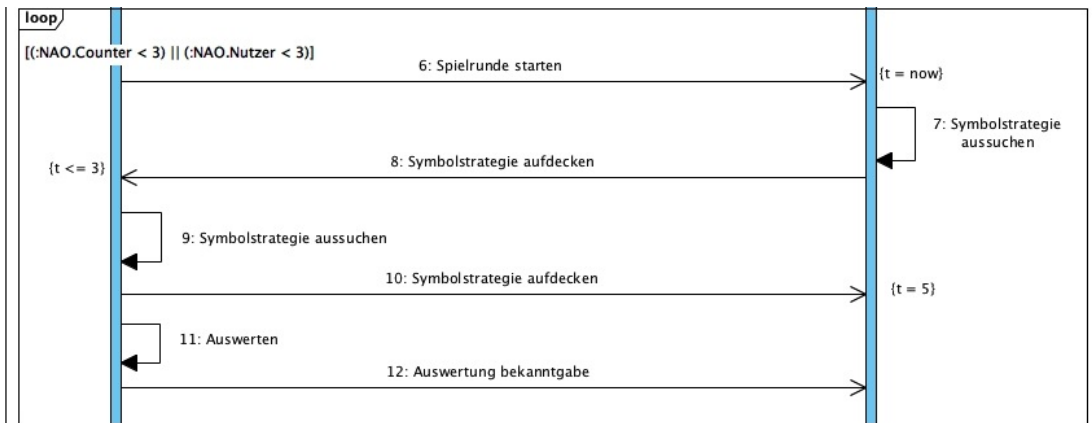


Abbildung 6.: Sequenzdiagramm: Ausschnitt "Hard-Strategie".

E. Hauptschritte des Szenario 1

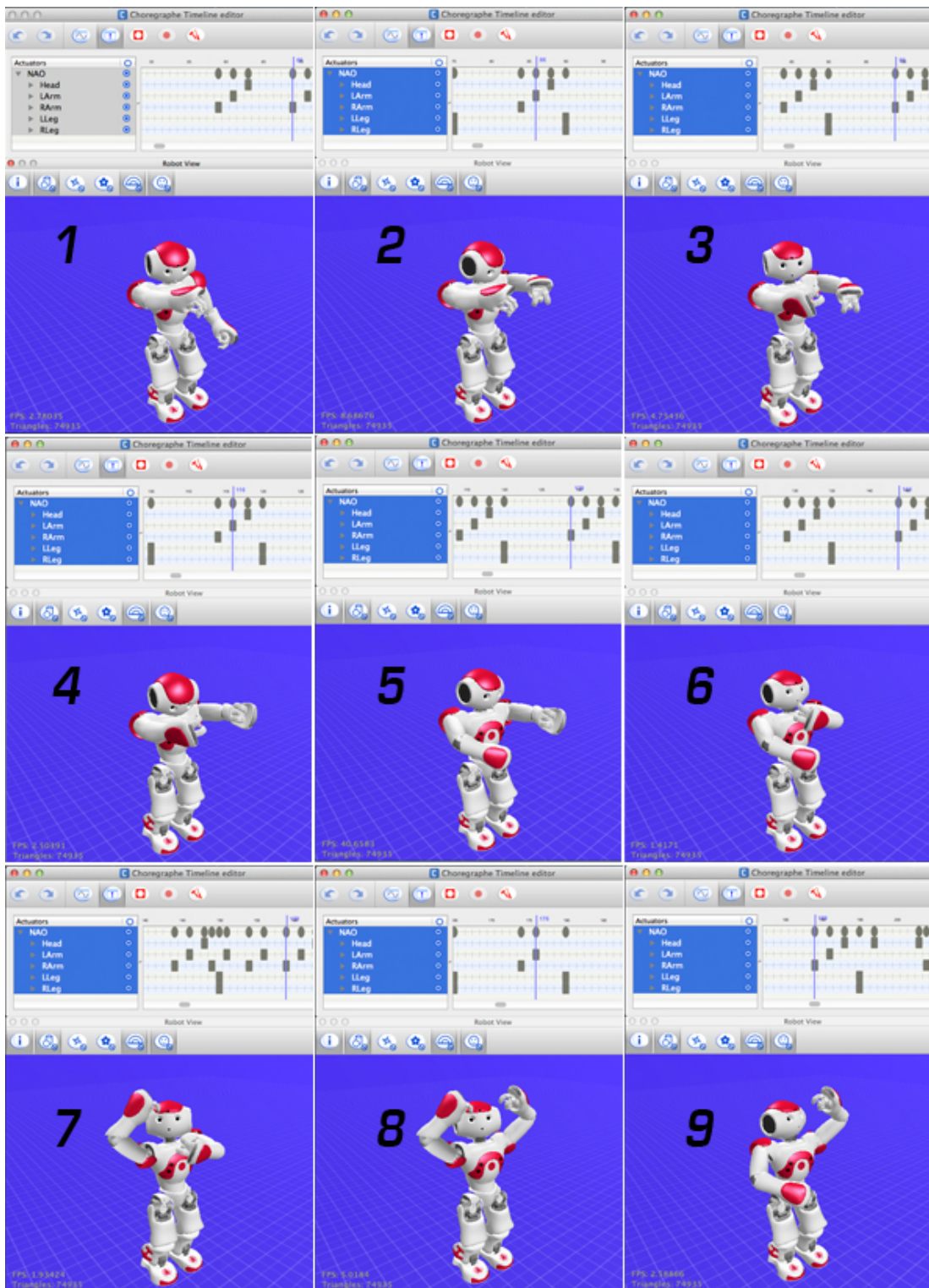


Abbildung 7.: Hauptschritte des Tanz Macarena.



Abbildung 8.: Hauptschritte des Tanz Macarena.

Abbildungsverzeichnis

2.1.	NAO-Roboter Familie: Die Torso-Modelle T2 (Links) und T14 (Mittig), die Humanoide-Modelle H21 und H25 (Rechts) [4].	4
2.2.	Computerarchitektur des NAO [5].	5
2.3.	Gelenkpositionierung des NAO [6].	6
2.4.	Softwarearchitektur des NAO.	8
2.5.	Auslesung der <i>autoload.ini</i> Datei und die damit verbundene Instanziierung der Module [9].	9
2.6.	Verzeichnisbaum mit der Instanz Broker als Wurzelknoten [9].	10
2.7.	Übersicht der Unterstützten Programmiersprachen [8].	12
3.1.	Übersicht der zu entwickelnden Szenarien.	15
3.2.	Ermittelte Anwendungsfälle der interaktiven Präsentation des NAO.	16
3.3.	Aktivitätsdiagramm: Tanzen.	20
4.1.	Übersicht der Systemarchitektur.	41
4.2.	Sequenzdiagramm: Auswahl der Schlusschoreographie.	46
4.3.	Schere (Links), Stein (Mittig), Papier (Rechts).	47
4.4.	Klassendiagramm des Szenario 2 "Schere, Stein, Papier".	50
4.5.	Abbildung des Klassendiagramm "Schere, Stein, Papier" auf Komponenten.	51
4.6.	Komponentendiagramm: "Schere, Stein, Papier".	52
4.7.	Sequenzdiagramm: "Random-Strategie".	53
4.8.	Klassendiagramm des Szenario 3 „Quiz-Show“.	54
4.9.	Abbildung des Klassendiagramm "Quiz-Show" auf Komponenten.	55
4.10.	Komponentendiagramm: "Quiz-Show".	56
4.11.	Sequenzdiagramm: Szenario 3 "Wer wird Millionär" mit Nutzung eines Joker.	57
4.12.	Komponentendiagramm: "Tanz erweitern".	58
4.13.	Sequenzdiagramm: Szenario 4 "Tanz erweitern".	59
5.1.	Timeline zur Aufnahme von Bewegungen.	62
5.2.	Erweiterungseditor der Timeline.	63

5.3.	Übersicht der Behavior des Szenario 2. Erstellt mit der IDE.	65
5.4.	Initialisierung.	66
5.5.	Auswertung des zu spielende Symbol von NAO.	68
5.6.	Hauptebene des Spiel "Quiz-Show".	71
5.7.	Aufbau der Fragen und der Antwortmöglichkeiten".	72
5.8.	Überblick des Ablaufes einer gestellten Frage.	73
5.9.	Überblick des Switch-Case.	73
5.10.	Erweiterte Timeline öffnen.	74
5.11.	Bewegungsfreigabe einzelner Körperregionen über den NAO-Simulator. . . .	75
6.1.	Versuchsaufbau für eine hohe Performance-Gewinnung.	86
1.	Aktivitätsdiagramm: Schere, Stein, Papier	i
2.	Aktivitätsdiagramm: Quizshow	ii
3.	Aktivitätsdiagramm: Tanzszenario erweitern	iii
4.	Modul Programmierung: Broker zu Proyx	vi
5.	Sequenzdiagramm: Ausschnitt "Easy-Strategie".	vii
6.	Sequenzdiagramm: Ausschnitt "Hard-Strategie".	viii
7.	Hauptschritte des Tanz Macarena.	ix
8.	Hauptschritte des Tanz Macarena.	x

Tabellenverzeichnis

2.1.	Verteilung der DoF im Körperbau des NAO	7
2.2.	Übersicht der gestellten Module im NAOqi API.	11
3.1.	Anwendungsfallbeschreibung: "Nutzer wahrnehmen".	17
3.2.	Anwendungsfallbeschreibung: "Moderation starten".	17
3.3.	Anwendungsfallbeschreibung: "Tanzszenario auswählen".	21
3.4.	Auszahlungsmatrix: Auswertungsregelung.	24
3.5.	Anwendungsfallbeschreibung: "Spielszenario auswählen (Schere, Stein, Papier)".	26
3.6.	Anwendungsfallbeschreibung: "Spielszenario auswählen (Quiz-Show)".	30
3.7.	Anwendungsfallbeschreibung: "Interaktivesszenario auswählen".	33
4.1.	Kriterien für die Implementierung.	36
4.2.	Auswertung der Evaluation anhand der ermittelten Kriterien.	40
4.3.	Übersicht der genutzten Hardwarekomponenten.	42
4.4.	Übersicht der gestellten Module im NAOqi API. Die blau Markierten Module sind szenarioübergreifende Module. Die grün Markierten Module werden ausschließlich im Szenario 1 verwendet. Die gelb Markierten Module werden im Szenario 2 genutzt.	43
4.5.	Übersicht der speziellen Hardwarekomponenten im Szenarien 1.	45
4.6.	Übersicht der speziellen Hardwarekomponenten im Szenarien 2.	48
5.1.	Sensoren um NAOs Steifheit der Gelenken zu lösen oder sperren.	62
5.2.	Die berechneten Beats pro Minute.	64
6.1.	Erfüllung der funktionalen Anforderung in der Konzeptrealisierung.	83
6.2.	Erfüllung der technischen Anforderung in der Konzeptrealisierung.	84
6.3.	Erfüllung der nicht-funktionalen Anforderung in der Konzeptrealisierung.	85
1.	Anwendungsfallbeschreibung: "Strategie wählen".	iv
2.	Anwendungsfallbeschreibung: "Joker einsetzen".	v

3. Anwendungsfallbeschreibung: "Interaktivesszenario erweitern". v

Listings

4.1. Quellcode "Hello World": Methoden aufruf anhand eines Proxy-Objekt	37
4.2. Quellcode "Hello World": Broker zu Main-Broker Konzept.	38
5.1. Easy Strategie	66
5.2. Medium Strategie	67
5.3. Quellcodeabschnitt Evaluation-Box: Inputsignal des NAO	68
5.4. Quellcodeabschnitt Evaluation-Box: Spielrunde Auswertung	69
5.5. Quellcodeabschnitt parallele Task beim Methodenaufrufe setzen	71

Abkürzungsverzeichnis

CPU	Zentraleinheit eines Computers
DoF	Freiheitsgrad
FSB	Front Side Bus
IDE	Integrierte Entwicklungsumgebung
Remote	Entfernt
SDK	Software Development Kit
UML	Unified Modeling Language
WLAN	Wireless Local Area Network

Glossar

Bussystem Ein Bussystem ist eine Datenleitung, welches Rechnerkomponenten verbindet.
Über diese Leitung ist es gestattet von mehreren Geräten ihre Daten zu übertragen.

DCM Ist ein spezielles Softwaremodul vom NAOqi. Kommuniziert ständig mit fast allen Hardwarekomponente (Sensoren, Aktoren), bis auf das Soundsystem und der Kameras.

Idle Leerlaufprozess.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 7. Oktober 2016 Asmatullah Noor