

Masterarbeit

Thema: Domain Specific Language specification and implementation for TTCN3

Name: *El Mehdi Bennani*
Anschrift: *Anschrift*
Matrikelnummer: *11033253*
Studiengang: *MTI*

Erstprüfer: *Prof. Dr. Hans W. Nissen*
Zweitprüfer: *Prof. Dr. Hans W. Nissen*

Anfertigungszeitraum:
Fertigstellung/Abgabedatum:

Ich erkläre an Eides statt, dass ich die vorgelegte Abschlussarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort, Datum, Unterschrift: _____

Abstract

Hier kommt die Kurzfassung hin

Abstract

und das Abstract wenn benötigt

Anleitung:

Motivation des Textes: worin liegt die Bedeutung der entsprechenden Forschung, warum sollte der längere Text gelesen werden?

- Fragestellung: welche Fragestellung(en) versucht der Text zu beantworten, was ist der Umfang der Forschung, was sind die zentralen Argumente und Behauptungen?
- Methodologie: welche Methoden/Zugänge nutzt der Autor/die Autorin, auf welche empirische Basis stützt sich der Text?
- Methodologie: welche Methoden/Zugänge nutzt der Autor/die Autorin, auf welche empirische Basis stützt sich der Text?
- Ergebnisse: zu welchen Ergebnissen kam die Forschung, was sind die zentralen Schlussfolgerungen des Textes?
- Implikationen: welche Schlussfolgerungen ergeben sich aus dem Text für die Forschung, was fügt der Text unserem Wissen über das Thema hinzu?

Contents

Abstract	I
List of figures	V
List of tables	VI
List of abbreviations	VII
Topic	VII
Preface	1
Introduction	2
1 Background	4
1.1 Human brain and languages	4
1.2 Transformational system	4
1.2.1 as transition system	4
1.2.2 Broca's area	5
1.3 System specification in tten-3	5
1.3.1 System composition	5
1.3.2 System development	5
1.3.3 System verification	5
1.3.4 Zweite Überschrift	6
2 Language Syntax	9
2.1 express structural concepts	9
2.1.1 features to describe	9
2.1.2 Zweite Überschrift	9
2.2 Well-written	9
2.2.1 qualities	9
2.2.2 TLA+	10
2.2.3 implementation inheritance	10
2.3 The process of writing a correct specification	12
2.3.1 features to describe	12
2.3.2 Sorting the elements of the specification	12
2.3.3 The Hierarchical Structure	12
2.3.4 System as a formula	13

3	Role of abstractions in domain modeling	15
3.1	basis for description and analysis	15
3.1.1	Zweite Überschrift	15
3.1.2	Zweite Überschrift	15
3.2	Well-designed abstractions	15
3.2.1	qualities	15
3.2.2	implementation inheritance	16
3.3	System verification and validation	19
4	Marktanalyse	20
4.1	dfhdhgj	21
4.2	thsgfnfgnj	21
4.3	gfhhgjdrtu	21
5	Entwicklung des Modells	22
5.1	Erstes Unterkapitel	22
6	algebraic design	24
	xyhas to be defined	24
6.1	Parser combinators vs. parser generators	24
6.2	SBT build system	26
7	Ergebnisse	27
7.1	Präsentation	27
7.2	Schwierigkeiten	27
7.3	Auswertungen	27
8	Zusammenfassung	28
8.1	Zusammenfassung	28
8.2	Ausblick auf zukünftige Arbeiten	28
8.3	Schlusswort	28
	Bibliography	29
	Anhang	31
	Erklärung	32
	Glossar	33
	Index	35

List of Figures

Fig. 1.1:	Aufbau allgemein	6
Fig. 1.2:	Humanoide Roboter	8
Fig. 2.1:	Aufbau allgemein	10
Fig. 2.2:	Humanoide Roboter	14
Fig. 3.1:	Aufbau allgemein	16
Fig. 3.2:	Humanoide Roboter	18

List of Tables

Tab. 1.1:	Titel	7
Tab. 2.1:	Titel	11
Tab. 3.1:	Titel	17

Topic

Die Idee an einem Unterstützungs-Roboter für gehbehinderte Menschen zu forschen, entstand durch das medial wachsende Interesse an Laufrobotern. Diese sind bereits im Entertainmentbereich z. B. bei Spielzeugen gängig, werden aber inzwischen auch vermehrt für Militärzwecke eingesetzt....

Preface

Das Vorwort ist optional.

Introduction

*“Ἀρχὴ ἡμισυ παντός
Der Anfang ist die Haelfte des Ganzen.”* (Vgl.[Ari35])

Roboter¹

Bei RÄßdern und Ketten ist dies hingegen nicht der Fall, Sie brauchen großflÄßchige StÄßtzpunkte.

Das bedeutet, dass letztere Arten der Fortbewegung einen ununterbrochenen Kontakt zum Boden benÄßtigen. (Vgl.[JZ03])

Dadurch wurden die im Verlauf der Evolution optimierten Konstruktionen von Beinen, das Zusammenspiel von Sensorik und Aktorik und die Steuerung von Gehbewegungen weitestgehend analysiert und dienten somit dem besseren VerstÄßndnis der Laufmotorik².

An dieser Stelle sei darauf hingewiesen, dass dieses Forschungsgebiet einen stark interdisziplinÄßren Charakter besitzt...

¹Der Begriff Roboter (tschechisch: robot) wurde von Josef und Karel Capek Anfang des 20. Jahrhunderts durch die Science-Fiction-Literatur geprÄßgt. Der Ursprung liegt im slawischen Wort robota, welches mit Arbeit, Fronarbeit oder Zwangsarbeit Äßbersetzt werden kann. 1921 beschrieb Karel Capek in seinem TheaterstÄßck R.U.R. in Tanks gezÄßchtete menschenÄßhnliche kÄßnstliche Arbeiter. Mit seinem Werk greift Capek das klassische Motiv des Golems auf. Heute wÄßrde man Capeks KunstgeschÄßpfe als Androiden bezeichnen. Vor der PrÄßgung dieses Begriffes wurden Roboter zum Beispiel in den Werken von Stanislaw Lem als Automaten oder Halbautomaten bezeichnet.

²Die Laufmotorik beinhaltet die Bewegungsfunktion und deren Lehre, die FÄßhigkeit des KÄßrpers sich kontrolliert zu bewegen, die Gesamtheit der vom zentralen Nervensystem kontrollierten Bewegungen des KÄßrpers im Gegensatz zu den unwillkÄßrlichen Reflexen des KÄßrpers und die Unterscheidung in Grob- und Feinmotorik

Kapitel 1 Übersicht

1 Background

1.1 Human brain and languages

Lots of studies have tried to teach limited forms of human language to other great apes. Gorillas and chimpanzees have been able to pick up and communicate through symbols and sign languages and even through a limited form of combinations of the human protolanguage

possibilities to learn new languages, dolmetschen übersetzen...

sens for communication

1.2 Transformational system

sequential algorithms

input data ?? compute ?? output result...

partial correctness + termination + complexity

computational model: Turing machines, RAM, term rewriting, . . .

das muss weg!

Reactive systems (operating systems, controllers, . . .) ? environment ?? system ?
 safety: something bad never happens ? liveness: something good eventually happens
 ? computational model: transition systems

1.2.1 as transition system

initial state Please input!!

in detail! abstract (finite-state) model

state transitions, entire transition system obtained as product

all states reachable, no deadlocks

tla+handout.pdf -> the Notes: ? If (S, L) is a system specification then it should usually be machine closed: otherwise they require unbounded look-ahead and are non-implementable

Dritte Überschrift

understand the foundations of DSLs

1.2.2 Broca's area

systems and properties

Dritte Überschrift

understand the foundations of DSLs

1.3 System specification in ttcn-3

read and write models

1.3.1 System composition

concepts of refinement and (de-)composition

1.3.2 System development

System development

1.3.3 System verification

validate models and prove properties certificat ttcn-3

Text....(siehe Abb. 1.1)

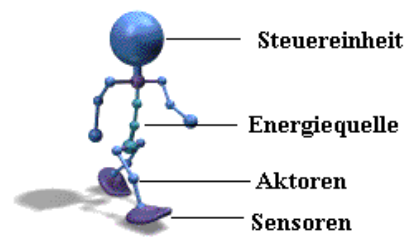


Figure 1.1: Aufbau und Komponenten von Robotern

Dritte Überschrift

- Die Bewegungsform der Achsen
- Anzahl und Anordnung der Achsen
- Die Formen des Arbeitsraums

... Arm zu strecken. (Vgl.[RS02])

1.3.4 Zweite Überschrift

Dritte Überschrift

Hier Text einfügen

Roboterfuß befindet

homogene

4 x 4 Matrix:

$$T = \begin{pmatrix} Ax & Ay & Az & 0 \\ Bx & By & Bz & 0 \\ Cx & Cy & Cz & 0 \\ Px & Py & Pz & 1 \end{pmatrix} \quad (1.1)$$

$$(\theta, d, a, \alpha) \quad (1.2)$$

verschiedene Matrizen:

$$T = \begin{pmatrix} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha & \arccos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & \arcsin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.3)$$

$${}^{n-1}T_n = \begin{pmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & a_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & a_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1.4}$$

$$T = T_1T_2T_3T_4T_5T_{tcp} \tag{1.5}$$

Text...(Siehe Tab. 1.1)

Titel:

Table 1.1: Titel

Überschrift 1	Überschrift 2
Text	Text
	Text
Text	Text
	Text

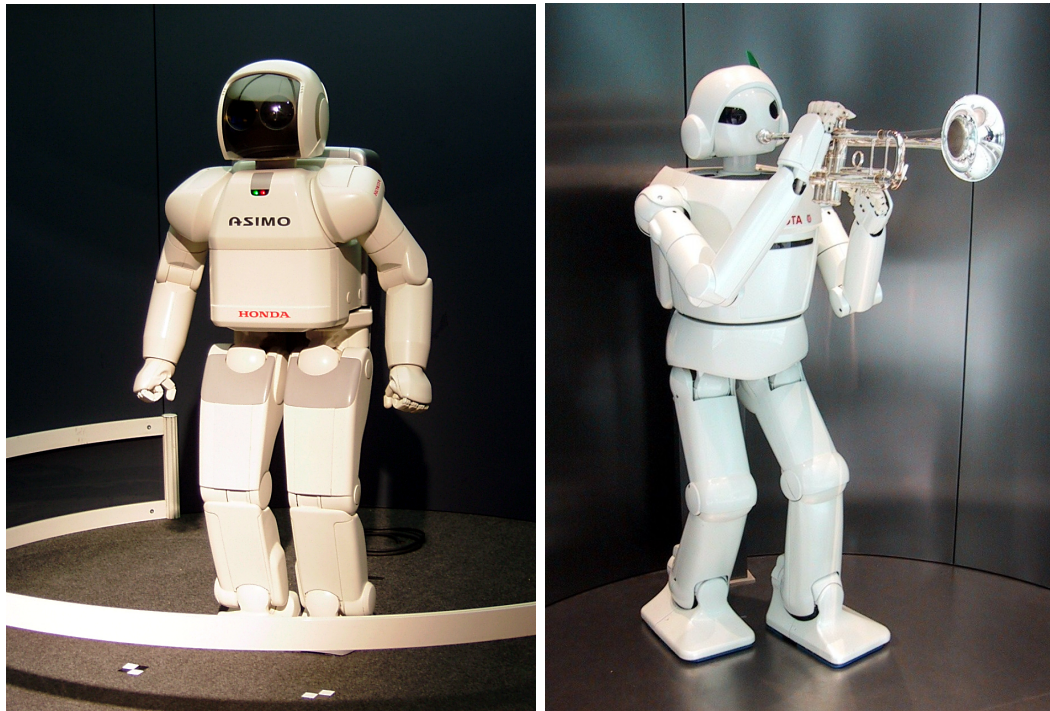


Figure 1.2: Humanoide Roboter Links der Aibo von Honda rechts der Roboter von Toyota

figure source:(http://de.wikipedia.org/wiki/Humanoider_Roboter)
Sichtung: 17.09.2010

In Abb. 1.2 sind...Text

Kraft \mathbf{F} , der Masse \mathbf{m} und der Beschleunigung \mathbf{a} kann mit der daraus resultieren Formel die Kraft, die wirkt, berechnet werden:

$$F = m * a \quad (1.6)$$

Folglich ist die Kraft das Produkt von Masse und Beschleunigung.

SI-Einheit der Kraft:

$$[F] = kg * \frac{m}{s^2} = Newton(N) \quad (1.7)$$

$$M = F * l = F * r * \sin \alpha \quad (1.8)$$

SI-Einheit des Drehmomentes:

$$[M] = Newtonmeter(N * m) \quad (1.9)$$

2 Language Syntax

2.1 express structural concepts

Why do we need a description? ? Designers: to communicate ideas ? Implementors: to build a conforming compiler ? Programmers: how do I write a while loop in language X?

2.1.1 features to describe

? syntax: legal sentences ? semantics ? static: checked at compile time ? dynamic: run time behaviour

a simple, compact and unambiguous notation: ? We need a formal description rather than an informal (natural) language description.

2.1.2 Zweite Überschrift

the ASCII character version

lexems identifiers and strings numbers are built-in primitive symbols

the language evolution

hier kann Kursiv geschrieben werden xy were sentences and equals was a verb xy are nouns and equals is an operation Just as the noun yx names the ... Many people think of all mathematical expressions as sentences, reading $1+1$ as Add 1 and 1. They think of $1 + 1 = 2$ as asserting

2.2 Well-written

how to write a module?

2.2.1 qualities

written in a declarative style that has the goal of making Text....(siehe Abb. 1.1)

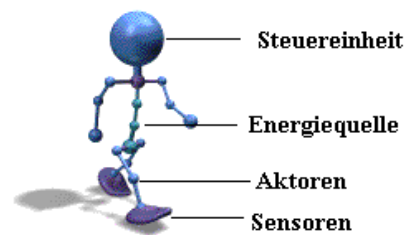


Figure 2.1: Aufbau und Komponenten von Robotern

2.2.2 TLA+

Propositional logic is the study of simple operations on Booleans (truth values)

all the possibilities that can occur, as probabilities all the cases that can be defined with the semantics

- introduce an extra level of indirection between your domain abstraction and the language of implementation.
- extensible abstractions
- Die Formen des Arbeitsraums

... extensible abstractions. (Vgl.[RS02]) as a manager translates the sentence into a set of separate sentences, checks the ones that are trivially true, and sends the others to one or more other ...

manager remembers if it has already .. an ..

This would be inelegant but feasible

being completely rigorous and using the notation of dsl2ttn3, you would have to write $G[1]$ and $G[2]$ instead of N and E . (If you were restricted to ordinary dsl notation, you would have no way to be rigorous.)

2.2.3 implementation inheritance

Dritte Überschrift

dsl in action subclasses become unnecessarily coupled to the implementation of your base class! how to distill an abstraction out of its nonessential details minimize accidental complexity in your abstractions Roboterfuß befindet

homogene
4 x 4 Matrix:

$$T = \begin{pmatrix} Ax & Ay & Az & 0 \\ Bx & By & Bz & 0 \\ Cx & Cy & Cz & 0 \\ Px & Py & Pz & 1 \end{pmatrix}$$

(2.1)

$$(\theta, d, a, \alpha)$$

(2.2)

verschiedene Matrizen:

$$T = \begin{pmatrix} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha & \arccos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & \arcsin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(2.3)

$${}^{n-1}T_n = \begin{pmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & a_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & a_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

(2.4)

$$T = T_1 T_2 T_3 T_4 T_5 T_{tcp}$$

(2.5)

Text...(Siehe Tab. 1.1)

Titel:

Table 2.1: Titel

Überschrift 1	Überschrift 2
Text	Text Text
Text	Text Text

2.3 The process of writing a correct specification

This is a procedure I use regularly when writing code that is at all tricky and that I want to be correct. But, getting this program correct was not important enough to warrant spending that much time. ???

2.3.1 features to describe

most obviously correct specifications could not be executed efficiently The thing to do in such a case is to write two versions of some definitions: one that is simpler and more obviously correct, and another that dsl can evaluate more efficiently.

2.3.2 Sorting the elements of the specification

We want to specify what it means to sort a set of elements. For simplicity, let's assume that each element is a record τ with a key component, and the elements are to be sorted in increasing order of their key value. Sorting a set means arranging its elements in a list. A sorting of a set S is therefore a list of elements of S that contains each element exactly once and is sorted according to the elements' key values. To describe this precisely, we must define: τ What a list of elements of S is. τ What it means for such a list to contain each element of S exactly once. τ What it means for the list to be sorted according to elements' key values.

However, if she defined a Turing machine T to be a 7-tuple, it would be impossible to remember if its initial state was $T[4]$ or $T[5]$. Programming languages solve this problem by introducing records.

the record

*Records in Programming Languages A record is called a struct in the C programming language. In more modern programming languages, an object is a record for which certain operations are defined. In addition to using $r.f$ to mean the value of field f of record r , these languages also use $r.O(\dots)$ to mean the value obtained (and side effects produced) by applying the operation O to the record r . If you program in an object-oriented language, you may miss some of its features when using TLA+. While those features are useful for writing programs, they are not needed for writing specifications, and their inherent mathematical complexity would make specifications that used them harder to understand.

2.3.3 The Hierarchical Structure

To write the spec, I had to decide how to represent the voters' ballots. The obvious way to represent a single voter's ballot is by a sequence whose i th element is the

name of the candidate the voter ranks number i . An obvious way to represent all the votes is as a set of such sequences. However, that's not right because two voters can cast identical ballots, and there is no concept of a set having "two copies" of an element. Here are three reasonable ways to represent the collection of votes: ? With a set whose elements are records or tuples with one component being the ranking and the other identifying the voter (with a randomly chosen identification if the vote is to be anonymous). ? With a bag (multiset) ? of rankings. ? With a sequence of rankings, arranged in an arbitrary order.

2.3.4 System as a formula

different specifications lead to the same Module definition one standard interpretation

Writing Structured specifications

dsl in action A module consists of module definition describing the module, followed by either: Writing Structured optional module control part ? A non-module control that is a sequence of steps, ending with a qed step, each of which may (but need not) have a control part. ? A module control, which is either the keyword obvious, the keyword omitted, or a by

after

conjunction... formula like $tla+$

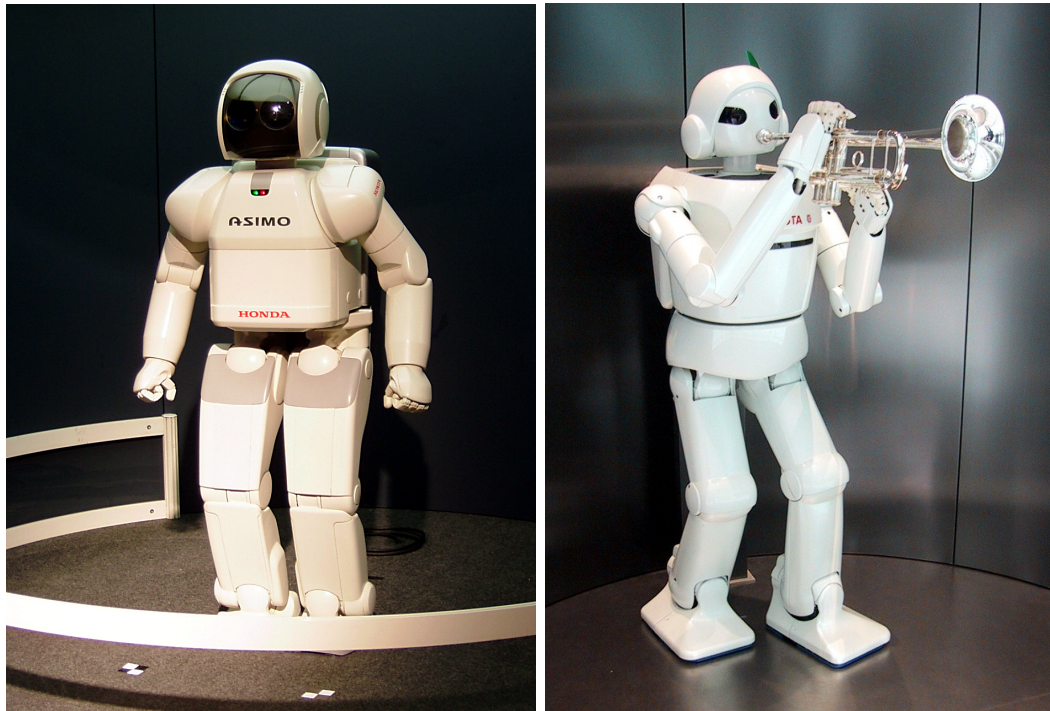


Figure 2.2: Humanoide Roboter Links der Aibo von Honda rechts der Roboter von Toyota

Bild-Quellen: (http://de.wikipedia.org/wiki/Humanoider_Roboter)
Sichtung: 17.09.2010

In Abb. 1.2 sind...Text

Kraft \mathbf{F} , der Masse \mathbf{m} und der Beschleunigung \mathbf{a} kann mit der daraus resultieren Formel die Kraft, die wirkt, berechnet werden:

$$F = m * a \quad (2.6)$$

Folglich ist die Kraft das Produkt von Masse und Beschleunigung.

SI-Einheit der Kraft:

$$[F] = kg * \frac{m}{s^2} = Newton(N) \quad (2.7)$$

$$M = F * l = F * r * \sin \alpha \quad (2.8)$$

SI-Einheit des Drehmomentes:

$$[M] = Newtonmeter(N * m) \quad (2.9)$$

3 Role of abstractions in domain modeling

3.1 basis for description and analysis

A DSL is nothing but a layer of abstraction over an underlying implementation model. Unless you get it right, your domain model won't be at the correct level of abstraction, and the linguistic representation of it in the DSL won't be either. well-engineered abstraction, do not contains nonessential details that don't belong to the core concerns of the abstraction design abstractions at the correct level to ensure that your model speaks the language of the domain abstraction needs to publish only the core essence to its clients, which makes it minimal from the outside looking in.

3.1.1 Zweite Überschrift

Every abstraction has a functionality to deliver to its clients minimality of abstractions. distillation of the abstraction

3.1.2 Zweite Überschrift

Extensibility and composability Extensibility is possible only through composability Engineering is about designing things in a modular way

Dritte Überschrift

hier kann Kursiv geschrieben werden Extensibility and composability Extensibility is possible only through composability Engineering is about designing things in a modular way Well-behaved abstractions can be composed to form higher-level abstractions

3.2 Well-designed abstractions

3.2.1 qualities

abstractions at the correct level to ensure that your model speaks the language of the domain. Only then will your code be as expressive as the language of the person who's an expert in the domain. Text....(siehe Abb. 1.1)

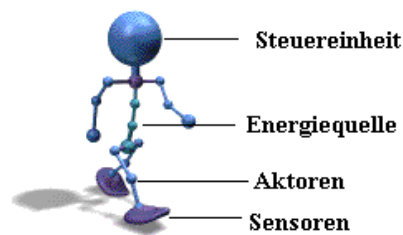


Figure 3.1: Aufbau und Komponenten von Robotern

how to design an abstraction

- introduce an extra level of indirection between your domain abstraction and the language of implementation.
- extensible abstractions
- Die Formen des Arbeitsraums

... extensible abstractions. (Vgl.[RS02]) difficult to extend `java.util.Map` using a true OO approach

Mixins let you do exactly what's needed here

Scala offers mixin implementation in the form of traits.

in the final implementation, the trait `xy` is mixed in dynamically during runtime object creation.

3.2.2 implementation inheritance

Dritte Überschrift

dsl in action subclasses become unnecessarily coupled to the implementation of your base class! how to distill an abstraction out of its nonessential details minimize accidental complexity in your abstractions Roboterfuß befindet

homogene
4 x 4 Matrix:

$$T = \begin{pmatrix} Ax & Ay & Az & 0 \\ Bx & By & Bz & 0 \\ Cx & Cy & Cz & 0 \\ Px & Py & Pz & 1 \end{pmatrix} \quad (3.1)$$

$$(\theta, d, a, \alpha)$$

(3.2)

verschiedene Matrizen:

$$T = \begin{pmatrix} \cos \theta & -\sin \theta \cos \alpha & \sin \theta \sin \alpha & \arccos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & \arcsin \theta \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(3.3)

$${}^{n-1}T_n = \begin{pmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & a_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & a_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

(3.4)

$$T = T_1 T_2 T_3 T_4 T_5 T_{tcp}$$

(3.5)

Text...(Siehe Tab. 1.1)

Titel:

Table 3.1: Titel

Überschrift 1	Überschrift 2
Text	Text Text
Text	Text Text

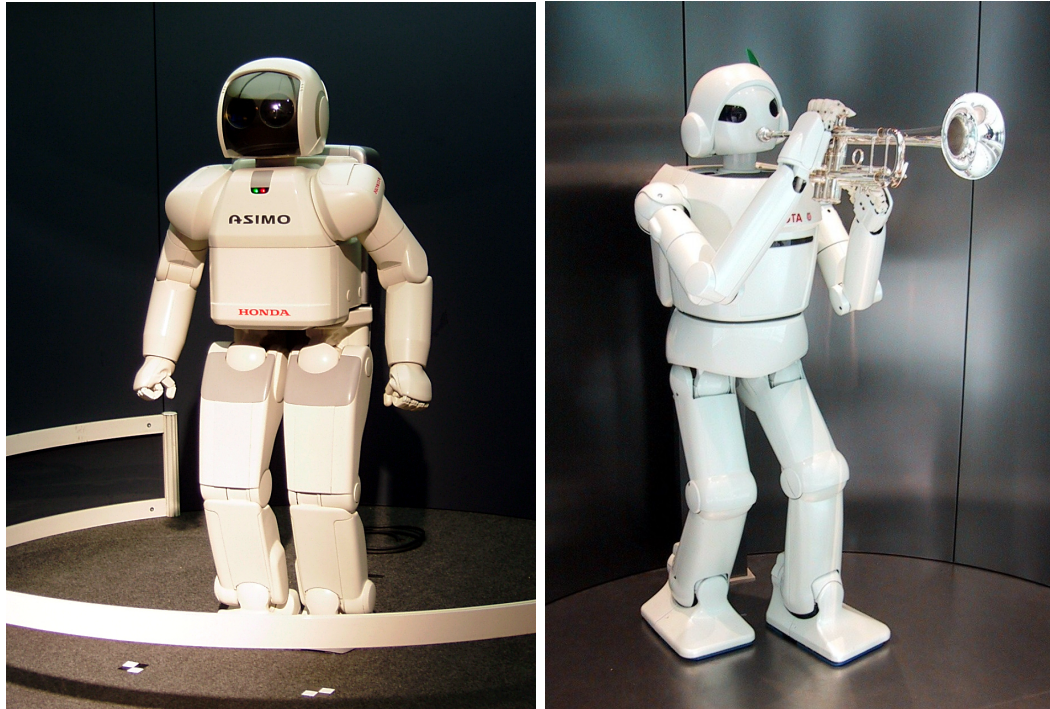


Figure 3.2: Humanoide Roboter Links der Aibo von Honda rechts der Roboter von Toyota

Bild-Quellen: (http://de.wikipedia.org/wiki/Humanoider_Roboter)
Sichtung: 17.09.2010

In Abb. 1.2 sind...Text

Kraft \mathbf{F} , der Masse \mathbf{m} und der Beschleunigung \mathbf{a} kann mit der daraus resultieren Formel die Kraft, die wirkt, berechnet werden:

$$F = m * a \quad (3.6)$$

Folglich ist die Kraft das Produkt von Masse und Beschleunigung.

SI-Einheit der Kraft:

$$[F] = kg * \frac{m}{s^2} = Newton(N) \quad (3.7)$$

$$M = F * l = F * r * \sin \alpha \quad (3.8)$$

SI-Einheit des Drehmomentes:

$$[M] = Newtonmeter(N * m) \quad (3.9)$$

3.3 System verification and validation

das musss weg!

Validation: are we building the right system ?

Verification: are we building the system right ?

4 Marktanalyse

Aufbau siehe Kapitel 1

4.1 dfhdhgj

Aufbau siehe Kapitel 1

4.2 thsgfnfgnj

Text

4.3 gfhfgjdrtu

Text

5 Entwicklung des Modells

Aufbau siehe Kapitel 1

5.1 Erstes Unterkapitel

1. Text
Text
2. Text
Text
3. Text
Text

Zuerst erfolgt zum besseren Verständnis die Deklaration der verwendeten Variablen

```
1 // die Motoren des rechten Beines werden wie in dem bisher
2 // verwendeten Schema zugewiesen
3 // Hüfte (C), Knie (B) und Fussgelenk (A)
4 public static Motor motorHip = new Motor(MotorPort.C);
5 public static Motor motorKnee = new Motor(MotorPort.B);
6 public static Motor motorAnkle = new Motor(MotorPort.A);
7 public static boolean sensorReached1 = false;
8
9 // der Ultraschallsensor wird an Port S1 erwartet
10 public static UltrasonicSensor sonicSensor1 =
11     new UltrasonicSensor(SensorPort.S1);
12
13 //Der TouchSensor wird an Port S2 erwartet und wird zum iterieren
14 // durch einen Bewegungsablauf verwendet
15 public static TouchSensor touchSensor =
16     new TouchSensor(SensorPort.S2);
```

1. Item 1

2. Item 2

3. Item 3

→ Item 1

→ Item 2

→ Item 3

6 algebraic design

*“Ἀρχὴ ἡμισυ παντός
Der Anfang ist die Hälfte des Ganzen.”* (Vgl.[Ari35])

Roboter¹

6.1 Parser combinators vs. parser generators

introduce a design approach called algebraic design

Parser combinators vs. parser generators

Das bedeutet, dass letztere Arten der Fortbewegung einen ununterbrochenen Kontakt zum Boden benötigen. (Vgl.[JZ03])

Dadurch wurden die im Verlauf der Evolution optimierten Konstruktionen von Beinen, das Zusammenspiel von Sensorik und Aktorik und die Steuerung von Gehbewegungen weitestgehend analysiert und dienten somit dem besseren Verständnis der Laufmotorik².

An dieser Stelle sei darauf hingewiesen, dass dieses Forschungsgebiet einen stark interdisziplinären Charakter besitzt...

¹Der Begriff Roboter (tschechisch: robot) wurde von Josef und Karel Capek Anfang des 20. Jahrhunderts durch die Science-Fiction-Literatur geprägt. Der Ursprung liegt im slawischen Wort robota, welches mit Arbeit, Fronarbeit oder Zwangsarbeit übersetzt werden kann. 1921 beschrieb Karel Capek in seinem Theaterstück R.U.R. in Tanks gezüchtete menschenähnliche künstliche Arbeiter. Mit seinem Werk greift Capek das klassische Motiv des Golems auf. Heute würde man Capeks Kunstgeschöpfe als Androiden bezeichnen. Vor der Prägung dieses Begriffes wurden Roboter zum Beispiel in den Werken von Stanislaw Lem als Automaten oder Halbautomaten bezeichnet.

²Die Laufmotorik beinhaltet die Bewegungsfunktion und deren Lehre, die Fähigkeit des Körpers sich kontrolliert zu bewegen, die Gesamtheit der vom zentralen Nervensystem kontrollierten Bewegungen des Körpers im Gegensatz zu den unwillkürlichen Reflexen des Körpers und die Unterscheidung in Grob- und Feinmotorik

Kapitelübersicht

6.2 SBT build system

build and run your Scala code using sbt, the build tool for Scala, and/or an IDE like IntelliJ or Eclipse. See the book's source code repo on GitHub for more information on getting set up with sbt. Sbt is very smart about ensuring only the minimum number of files are recompiled when changes are made. It also has a number of other nice features which we won't discuss here.

sbt uses a small number of concepts to support *flexible and powerful build definitions*. There are not that many concepts, but sbt is not exactly like other build systems and there are details you will stumble on if you haven't read the documentation

7 Ergebnisse

7.1 Präsentation

Aufbau siehe Kapitel 1

7.2 Schwierigkeiten

Text

7.3 Auswertungen

Text

8 Zusammenfassung

8.1 Zusammenfassung

Text

8.2 Ausblick auf zukünftige Arbeiten

Text

8.3 Schlusswort

Text

Bibliography

- [Ari35] ARISTOTELES: *Politik 5*, v.Chr. 335. 4. 1303 b 29.
- [JZ03] JZ, JENS ZIEGLER: *Evolution von Laufrobotersteuerungen mit Genetischer Programmierung*, 2003. <https://eldorado.tu-dortmund.de/handle/2003/2743>. Sichtung: 20.07.2010.
- [RS02] RS, ROLAND STENZELER: *Steuerungsarchitekturen für autonome mobile Roboter*, 2002. http://darwin.bth.rwth-aachen.de/opus3/volltexte/2002/408/pdf/Stenzel_Roland.pdf Sichtung: 04.09.2010 oder im Anhang(CD).

Anhang

Inhalt Anhang

Auf der mitgelieferten CD befindliche Dateien:

1. Video der Ergebnisse des entwickelten Transmovers
2. Marktanalyse
 - BIC Analyse
 - Fragebögen befragter Betroffener
3. Skizzen und Entwürfe
 - Skizze
 - Entwurf LEGO Designer
4. Programmablaufpläne
5. Excel Datei Auswertung Winkelmessung
6. Sequenzdiagramme
7. Quellcodes
 - Winkeleinmessung
 - Transmover LabVIEW Programme
 - Transmover JAVA Programme
 - Wii Remote Programme

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben.

Die Arbeit hat nach meinem Wissen mit gleichem Inhalt noch keiner anderen Prüfungsbehörde vorgelegen.

Gummersbach, 14. Oktober 2010

Thomas Karanatsios

Glossar

Autonom	das Programm welches implementiert ist arbeitet weitgehend unabhängig von Benutzer eingriffen.
Biometrie	Die Biometrie auch Biometrik genannt beschäftigt sich mit Messungen an Lebewesen und den dazu erforderlichen Mess- und Auswerteverfahren.
Bionisch	Adjektiv zur Beschreibung eines Organismus, dessen biologische Grundlage durch technische Möglichkeiten verbessert wurde.
Degrees of freedom	(DOF) bedeutung Freiheitsgrade = Der Freiheitsgrad bezeichnet einen Parameter eines Systems. Die Eigenschaft, ein Freiheitsgrad zu sein, ergibt sich für einen Parameter daraus, Mitglied in einer Menge von Parametern zu sein, die das System beschreiben.
Deliberativ	Deliberativ = erwägen, überlegen, sich entscheiden, beschließen ist eine semantische Funktion Verbmodus des Konjunktivs z. B. im Lateinischen (coniunctivus deliberativus), die eine überlegende Rückfrage als Reaktion auf eine Aufforderung ausdrückt.
Dynamik	Eine Dynamik steht für, das Teilgebiet der Mechanik, das sich mit der Wirkung von Kräften befasst
Endeffektor	Als Endeffektor wird in der Robotik das letzte Element einer kinematischen Kette bezeichnet. Bei Industrierobotern kann es sich hierbei zum Beispiel um eine Einheit zum Schweißen von Autokarosserien oder allgemein um einen einfachen Greifer handeln. Der im englischen als TCP (Tool Center Point) bezeichnete ausgezeichnete Punkt am Ende der kinematischen Kette ist das Zielsystem, für das die aus der gestellten Aufgabe resultierenden Positionierungsforderungen gelten. Aufgaben spezifisch kann der TCP dabei auch außerhalb des Roboters liegen, Beispiele wären der Fokus eines gegriffenen Lasers oder auch die Mitte des gerade transportierten Objekts.

Dorsalextension	steht für die Bewegung in den Zehengelenken in Richtung Fußrücken.
Extension	Die Extension (von lat. extensio "Streckung") ist die Streckung eines Gelenkes. Die gegenläufige Bewegung wird als Flexion bezeichnet.
Exoskelett	Ein Exoskelett ist eine Stützstruktur für einen Organismus, das eine stabile äußere Hülle um diesen bildet.
Energy Harversting	Als Energy Harvesting (wörtlich übersetzt Energie-Ernten) bezeichnet man die Erzeugung von Strom aus Quellen wie Umgebungstemperatur, Vibrationen oder Luftströmungen. Die Industrie entwickelt bereits heute Energiequellen für drahtlose Sensornetzwerke oder Anwendungen wie etwa Fernbedienungen an schwer erreichbaren Stellen. Energy Harvesting vermeidet bei Drahtlostechnologien Einschränkungen durch kabelgebundene Stromversorgung oder Batterien.
Flexion	Die gegenläufige Bewegung zur Extension wird als Flexion bezeichnet.
Inertialsystem	In der Physik ist ein Inertialsystem (von lat. iners "untätig, träge") ein Koordinatensystem, in dem sich kräftefreie Körper geradlinig, gleichförmig bewegen. In einem Inertialsystem gilt also das newtonsche Trägheitsgesetz in seiner einfachsten Form, nach der kräftefreie Körper ihre Geschwindigkeit in Betrag und Richtung beibehalten und Beschleunigungen proportional zur anliegenden Kraft erfolgen. Der Begriff Inertialsystem wurde erstmals von Ludwig Lange (1885) verwendet.
Inhibition	Das Wort Inhibition (lat. inhibere "unterbinden", "anhalten"; veraltend Inhibierung, deutsch Hemmung, Antonym Desinhibition, Desinhibierung) bezeichnet: in der Neurobiologie eine Abnahme der Erregbarkeit von Nervenzellen, siehe Inhibition (Neuron) in der Ethologie die Blockierung einer Verhaltensweise durch innere oder äußere Faktoren, siehe Bedingte Hemmung in der Digitaltechnik bezeichnet die Inhibition eine Schaltung aus einem UND- und einem NICHT-Glied, siehe Inhibition (Digitaltechnik)
Ipsilateral	Ipsilateral bedeutet "auf derselben Körperseite oder -hälfte gelegen". Das Gegenteil von ipsilateral ist kontralateral.

Widmung

Die Widmung ist optional! Text

Avec tout l'amour de mes parents,
Votre fils