

Seminarmaterial:

Präsentation zur Sitzung 3

Coding & Robotik – Problemlösen, Modellieren und Programmieren am Beispiel von Lernrobotern

Autor:

Raphael Fehrmann, Horst Zeinz



V1 – 07/ 2020

Verwertungshinweis:

Die Medien bzw. im Materialpaket enthaltenen Dokumente sind gemäß der Creative-Commons-Lizenz „CC-BY-4.0“ lizenziert und für die Weiterverwendung freigegeben. Bitte verweisen Sie bei der Weiterverwendung unter Nennung der o. a. Autoren auf das Projekt „Lernroboter im Unterricht“ an der WWU Münster | www.wwu.de/Lernroboter/ . Herzlichen Dank! Sofern bei der Produktion des vorliegenden Materials CC-lizenzierte Medien herangezogen wurden, sind diese entsprechend gekennzeichnet.

Vorlage für einen entsprechenden Verweis:

Raphael Fehrmann, Horst Zeinz: Lehrmaterial zum Hochschulseminar „Lernroboter im Unterricht“;
Forschungsprojekt „Lernroboter im Unterricht“ an der Westfälischen Wilhelms-Universität Münster;
Abruf über: <https://www.uni-muenster.de/Lernroboter/seminar/>;
Lizenz: [CC-BY-4.0, www.creativecommons.org/licenses/by/4.0/deed.de](https://creativecommons.org/licenses/by/4.0/deed.de)

Kontakt zum Projekt:

Forschungsprojekt
«Lernroboter im Unterricht»

WWU Münster, Institut für
Erziehungswissenschaft

Prof. Dr. Horst Zeinz
» horst.zeinz@wwu.de

Raphael Fehrmann
» raphael.fehrmann@wwu.de

www.wwu.de/Lernroboter/

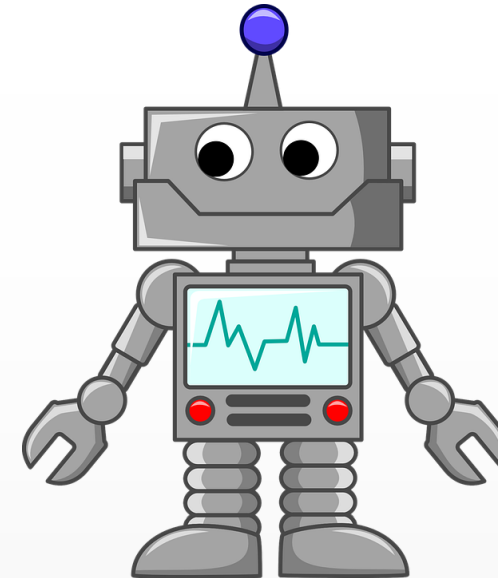
Das Projekt wird als
„Leuchtturmprojekt 2020“
gefördert durch die



**UNIVERSITÄTS
GESELLSCHAFT**
MÜNSTER

Sitzung 3:

Coding & Robotik – Problemlösen, Modellieren und Programmieren am Beispiel von Lernrobotern



Horst Zeinz | Raphael Fehrmann

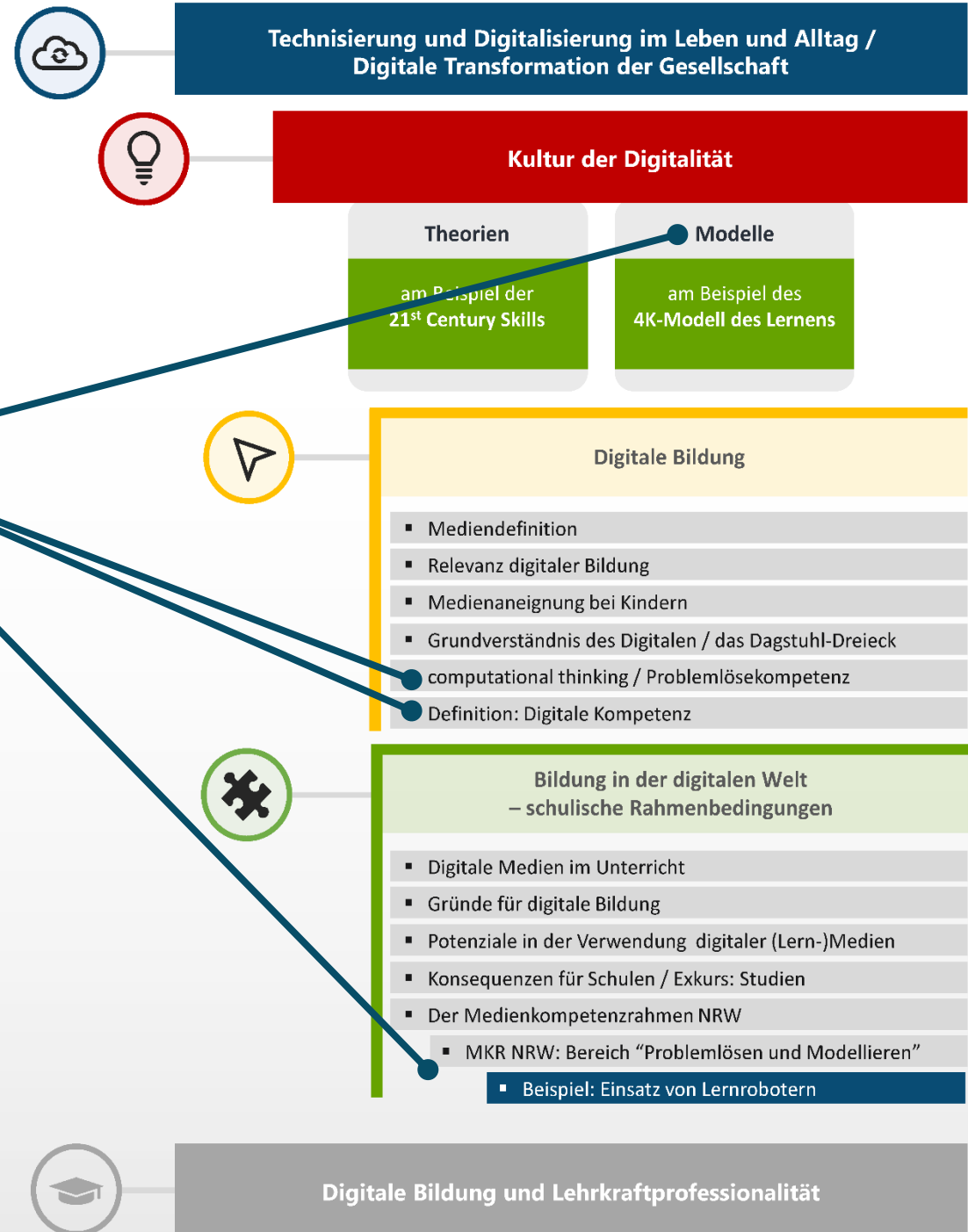
Ausblick auf die heutige Sitzung

Stundenthema:

Coding & Robotik:

Problemlösen, Modellieren und

Programmieren am Beispiel Lernroboter



Inhaltsverzeichnis



Roboter



Grundlagen der Algorithmik



Problembasiertes Lernen



Programmieren bzw. algorithmisches Denken als Problemlösen



Rückbezug zu Robotertechnik und Ausblick auf die Didaktik



- Welche Gedanken / Impressionen / Beispiele / Fragen... kommen Ihnen, wenn Sie den Begriff Robotik hören? Bitte sammeln Sie diese im etherpad! (Link im moodle hinterlegt)

Auswertung des etherpads



 Typ	Webanwendung	 Zielgruppen	Primarstufe, Sekundarstufe I, Sekundarstufe II, Erwachsenenbildung
 Betriebssystem	systemunabhängig	 Kosten	kostenfrei, s. Kommentar
 Installation	LP - nicht erforderlich SuS - nicht erforderlich	 Kommentar zu Kosten	ZUM-Wiki bietet kosten-freie, aber gänzlich öffentliche etherpads für Bildungskontexte an. Etherpad kann als kostenfreie OpenSource-Installation jedoch auch selbst auf eigenen Servern installiert und in Tools wie Nextcloud eingebunden werden.
 Internet-anbindung	LP - erforderlich SuS - erforderlich	 Setting	Partnerarbeit, Gruppenarbeit, Plenum
 Registrierung	LP - nicht erforderlich SuS - nicht erforderlich	 Aufwand der eigenen Einarbeitung	gering
 didaktischer Zweck	Kollaboration, Texterzeugung, Tabellenerzeugung		

Die vollständige Methodenkarte mit weiterführenden Hinweisen und den Links zum Produkt finden Sie im moodle sowie unter www.wwu.de/Lernroboter (CC-BY-lizenziert).



▪ Technischer Begriff: autonome Roboter

Roboter

- Karel Čapek, Anfang des 20. Jahrhunderts, Ursprung in der Science-Fiction-Literatur
- zurückgehend auf das damalig tschechische Wort „robot(a)“, übers.: (Zwangs-)Arbeit ([Link](#))
- zu verorten in der technischen Kategorie „Automaten“
- **Definition der RIA (Robot Industrie of America, 1979) / der Begriff ist länderspezifisch verschieden definiert:**
- Ein Roboter ist ein programmierbares Mehrzweck-Handhabungsgerät für das Bewegen von Material, Werkstücken, Werkzeugen oder Spezialgeräten. Der frei programmierbare Bewegungsablauf macht ihn für verschiedenste Aufgaben einsetzbar.
- *original: A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks.*

autonom

- eigenständig, eigenverantwortlich
- in Bezug auf die Robotik: Funktion des Roboters und Ausführung von Befehlen ohne direkte externe Unterstützung
- Roboter-Autonomie kann je nach Einsatzgebiet variieren

vgl. Wüst 2018



▪ Technischer Begriff: autonome Roboter – mobil oder stationär

mobil

- bei Robotern, die Umgebung erkunden können und über Arten der Fortbewegung verfügen (Fahren, Laufen, Fliegen, Schwimmen...)
- unbegrenzter Arbeitsraum, der über Sensoren erkundet wird
- wenn vernetzt, dann drahtlos
- Betrieb über Akkus
- Beispiele: Laufroboter, Flugroboter, Unterwasserroboter...

stationär

- bei Robotern, die an einem festen Standpunkt gebunden aufgerichtet und in Betrieb genommen werden,
- wegen Standortbindung i. d. R. an festen Arbeitsablauf eines Programms gebunden (Bsp.: Fertigungsstraße in der Automobilindustrie), begrenzter Arbeitsraum
- i. d. R. Industrieroboter ohne Sensoren



- **Interdisziplinäre Wissenschaft des Roboters:**

- Robotik, in verschiedensten Disziplinen vertreten (Informatik, Mathematik, KI, Kognitionswissenschaft, Psychologie,...)

- **Ethikbezogene Regularien**

1. Ein Roboter darf keinen Menschen verletzen.
2. Ein Roboter muss den Anweisungen gehorchen, die ihm von Menschen bspw. per Coding gegeben werden. Widerspricht eine Anweisung der ersten Regel, darf sie nicht ausgeführt werden.
3. Ein Roboter muss seine eigene Existenz so lange zu sichern versuchen, wie dies nicht der ersten oder zweiten Regel widerspricht.

in Anlehnung an Oubatti 2007
zurückgehend auf Issac Asimaov



▪ erste Roboter-ähnliche Konstruktion:

- dampfgetriebene fliegende Taube
- Erfinder: griechischer Mathematiker Archytas, 400 v. Chr.
- roboterähnliche Maschine (im Vergleich nicht „intelligent“, nur zur Ausführung einer Tätigkeit bestimmt)
- im Bereich der Automaten zu verankern
- weitere Automaten-Beispiele: Wasseruhren, Heronesball (Kugel, die sich dreht, wenn das unter ihr stehende Wasser erhitzt wird)
- Entwicklung erster Roboter gemäß dem heutigen Begriffsverständnis ab Beginn des 20. Jahrhunderts



Roomba 900 (Kategorie Haushaltsroboter)

- weitere Beispiele: Rasenmäher-, Wisch-, Poolreinigungs-Roboter

Bild: Tibor Antalóczy, iRobot Roomba 780,
CC-BY-SA, wikipedia.org

Pillo (Kategorie Medizinroboter)

- beantwortet Fragen, gibt Medikamente aus
- kann Nutzer durch Gesichtserkennungssoftware erkennen und medizin. Bedürfnisse speichern
- kann sich mit Ärzten oder dem Krankenhaus in Verbindung setzen

Bild: Herstellerfoto, pillohealth.com



Hexacopter (Kategorie Drohne)



- zum Überblick über bspw. Katastrophengebiete

Bild: pixabay.com, Autor: Pexels

Roboterhund Aibo (Sony, Kategorie sozial-intelligenter Roboter)

- unterstützt Personen mit besonderen Bedürfnissen (motorisch)
- und erinnert an wiederkehrende Aufgaben (z. B. Tabletteneinnahme)

Bild: Jörg Kantel, flickr, BY-NC-ND



ATLAS 2030 (Kategorie Exoskelett-Bot)

- ermöglicht Personen mit neuromuskulären Krankheiten das Gehen

Bild: [Herstellerfoto](#)

SpotMini (Kategorie Medizinroboter)

- vierbeiniger Helfer-Bot
- nach dem Vorbild eines Hundes
- Unterstützung des Menschen im Alltag (Dinge greifen, Treppen steigen, Hindernisse umgehen, Türen öffnen)

Bild: Nathan Proudlove, flickr, BY-NC



Pepper (Kategorie Sozialer Roboter)

- Fokus: Interaktion mit dem Menschen
- erster humanoider Roboter, der die Gefühle von Menschen erkennt und in Echtzeit reagieren kann
- zur Realisierung vielfältiger Aufgaben in Restaurants, Banken, Hotels, Krankenhäusern, Einkaufszentren
- imitiert Bewegungen des Menschen inkl. Körpersprache und emotionsähnlichen Reaktionen, „kann zuhören“
- „verwandt“ mit dem humanoiden Roboter NAO

Bild: Xavier Caré / Wikimedia Commons / CC-BY-SA



Paro (Kategorie Sozialer Roboter)

- Einsatzort: tiergestützte Therapie, in den Bereichen, in denen Patienten sich nicht um ein echtes Tier kümmern können oder keinen Zugang zu diesen haben können
- optisch dem Sattelrobbenbaby ähnlich
- Kunstpelz ist weich und strapazierfähig
- Kopf bewegt sich in Richtung der Schallquelle
- Augen blinzeln regelmäßig, schließen sich bei Streicheln des Fells
- Sensoren im Fell reagieren auf Berührung
- Gesicht, Körperbewegung und Töne drücken Gefühle aus



Soziale Roboter

- interagieren mit Menschen
- verstehen menschliche Kommunikation, reagieren auf diese

Weltraumroboter

- auf Bedingungen der Weltraumerkundung ausgelegt

Industrie- und Arbeitsroboter

- für wiederkehrende Aufgaben (Montage, Verpackung...)

Kollaborative Roboter

- Industrieroboter, die gefahrlos neben / mit Menschen arbeiten können
- Programmierung per Tablet-Eingabe oder per Nachahmung von Tätigkeiten („Sehen und Ausführen“)

Humanoide Roboter

- ähneln in Optik und Aufbau der menschlichen Silhouette
- besitzen höhere künstliche Intelligenz
- sammeln bspw. Erinnerungen, denken „selbstständig“

Schwarmroboter

- hunderte kleine Roboter, die vernetzt sind und gemeinsam eine Aufgabe bearbeiten (Assoziation: Insektenschwarm)

Medizinroboter

- Beispiele: Roboter-Rollstühle, Exoskelette

vgl. Buller et al. 2019, S. 26 f.



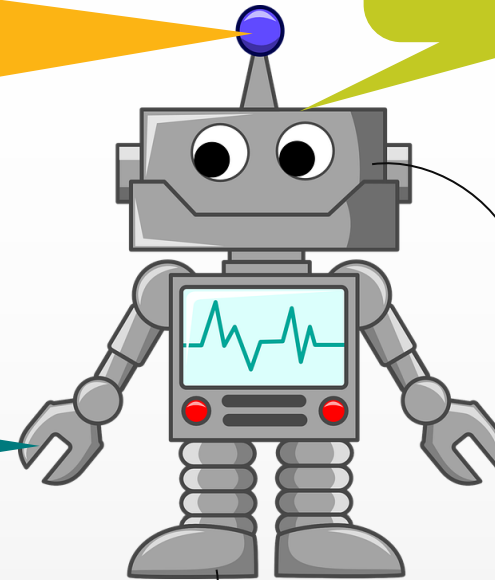
Sensoren zur Sammlung von Informationen aus der Umwelt (hier: Antenne, „Augen“ / Bewegungssensoren oder Kamera,...)

Körper / **Grundgerüst** als Hülle

Aktoren (inkl. Bewegungssystem und Interaktionssystemen, hier: Display, Greifer, „Füße“)

„**Computer-Gehirn**“ / Steuereinheit zur Informationsverarbeitung (CPU / Central Processing Unit, bestehend aus Leiterplatten)

Energiequelle (als Akku mit Ladestation bzw. Lademöglichkeit wie Solar oder dauerhafte Stromzufuhr)





Umwelt

Robotersystem

Sensoren

zur Orientierung in der Umwelt



Steuereinheit

zur Informationsverarbeitung



Aktoren

zur Ausführung von Aktionen



- empfangen Informationen aus der Umwelt als physikalisches Signal (Lichtsensoren: Helligkeitsveränderung, Temperatursensoren: Veränderung der Umgebungstemperatur, Ultraschallsensoren: zur Wahrnehmung von Gegenständen / Hindernissen...)
- geben Information als elektrisches Signal an Steuereinheit

- „Computer“, bestehend aus Leiterplatten
- enthält Programm mit spezifischen, algorithmischen Anweisungen („Wenn..., dann...“):
 - umfasst Gesamtablaufplan mit allen nötigen Algorithmen,
 - wertet Sensordaten aus und „interpretiert“ diese
- und steuert auf Basis der über die Sensoren empfangenen Informationen die Aktoren an

- vollziehen physikalische Aktion auf Basis der von der Steuereinheit weitergegebenen Informationen



- Prinzipien:
 - Sensoren, Steuereinheit und Aktoren wirken zusammen
 - Steuereinheit verfügt über ein Programm, bestehend aus algorithmischen Definitionen von Abläufen (Befehle)
 - Umsetzung des Programms über verschiedene Programmiersprachen in unterschiedlichen Darstellungen (textbasiert, grafisch)



Aktor

Algorithmus

Ort und Bedingungen, unter denen der Roboter arbeitet

bewegliche Maschine, die von einem Computer so gesteuert wird, dass sie Aufgaben ausführt (Wahrnehmung von Veränderungen in der Umwelt, autonome Reaktion auf diese)

Code

Sensor

anderer Begriff für „Programm“; algorithmische Anweisungen, die in Programmiersprache geschrieben werden

Programm

Programmieren / Programmierung

Schreiben von Programmen

Bauteil eines Roboters, das Informationen aus der Umwelt empfängt

Roboter

Umgebung

Reihe von Schritten, die ein Computer beim Lösen von Problemen oder während einer Aufgabe abarbeitet

bewegliches Teil eines Roboters, z. B. ein Motor oder ein Roboterarm

Reihe von Anweisungen, anhand derer ein Computer oder ein Roboter bestimmte Aufgaben ausführen kann

Roboter

Klärung zentraler Begrifflichkeiten



Aktor	bewegliches Teil eines Roboters, z. B. ein Motor oder ein Roboterarm
Code	anderer Begriff für „Programm“; algorithmische Anweisungen, die in Programmiersprache geschrieben werden
Programm	Reihe von Anweisungen, anhand derer ein Computer oder ein Roboter bestimmte Aufgaben ausführen kann
Roboter	bewegliche Maschine, die von einem Computer so gesteuert wird, dass sie Aufgaben ausführt (Wahrnehmung von Veränderungen in der Umwelt, autonome Reaktion auf diese)

Algorithmus	Reihe von Schritten, die ein Computer beim Lösen von Problemen oder während einer Aufgabe abarbeitet
Sensor	Bauteil eines Roboters, das Informationen aus der Umwelt empfängt
Programmieren / Programmierung	Schreiben von Programmen
Umgebung	Ort und Bedingungen, unter denen der Roboter arbeitet

Geringfügig adaptiert aus:
Buller et al. 2019, S. 152-155

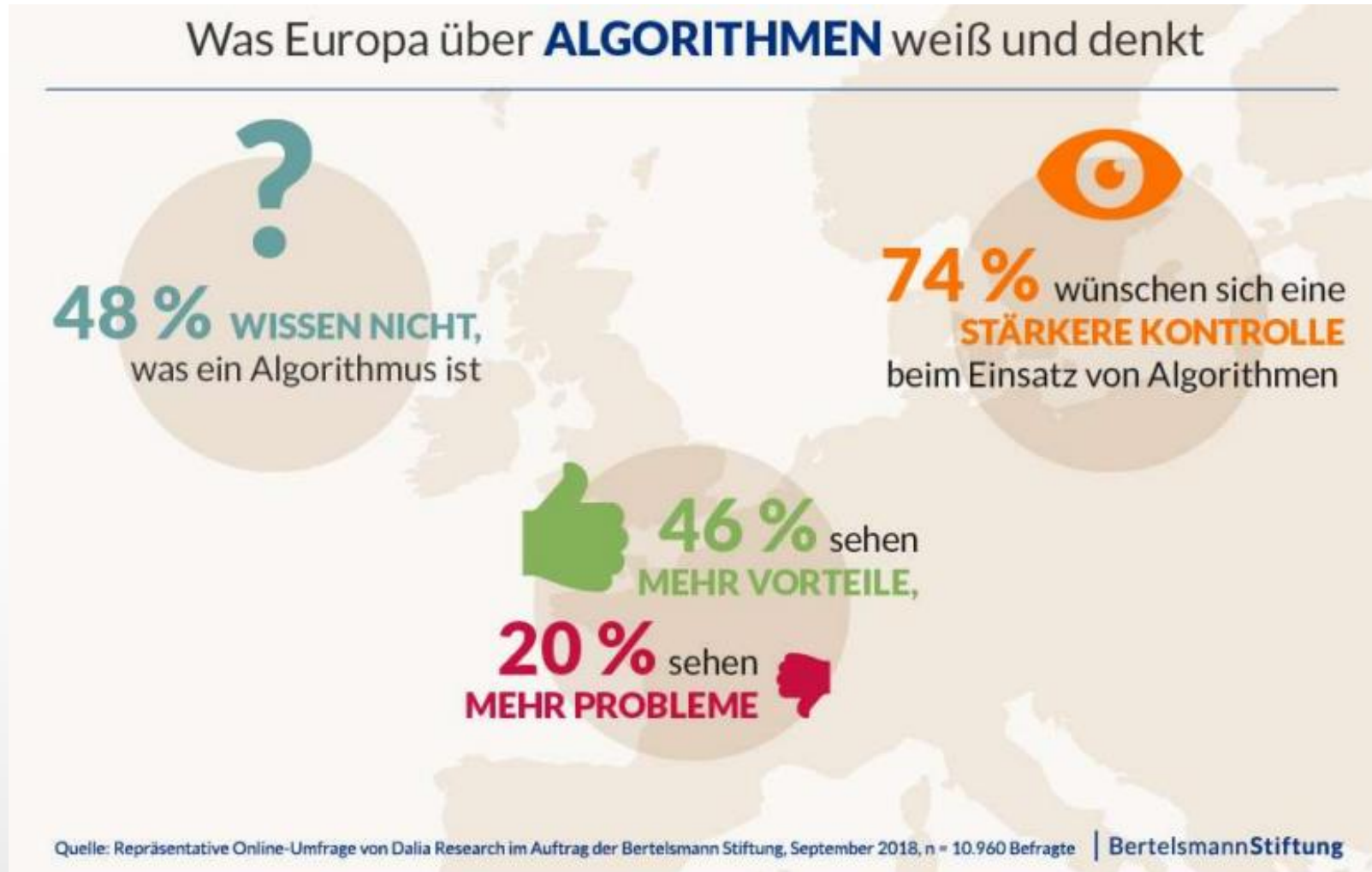


Ziel: Problemlösung durch Programmierung einer Maschine
/ eines Roboters unter Nutzung von Algorithmen

zum Weiterlesen, auch im
didaktischen Kontext: Specht 2019



- „Wer bin ich?“ (Wir spielen eine Runde im Plenum.)
- Erkenntnis:
 - Befragung nach Größe der Zielgruppe, von großen Gruppen (Geschlecht...) nach kleineren gruppenspezifischen Merkmalen (Berufsfeld...)
- Anwendung eines Sortier-Eingrenzungsalgorithmus
- Problemzerlegung, Abstraktion

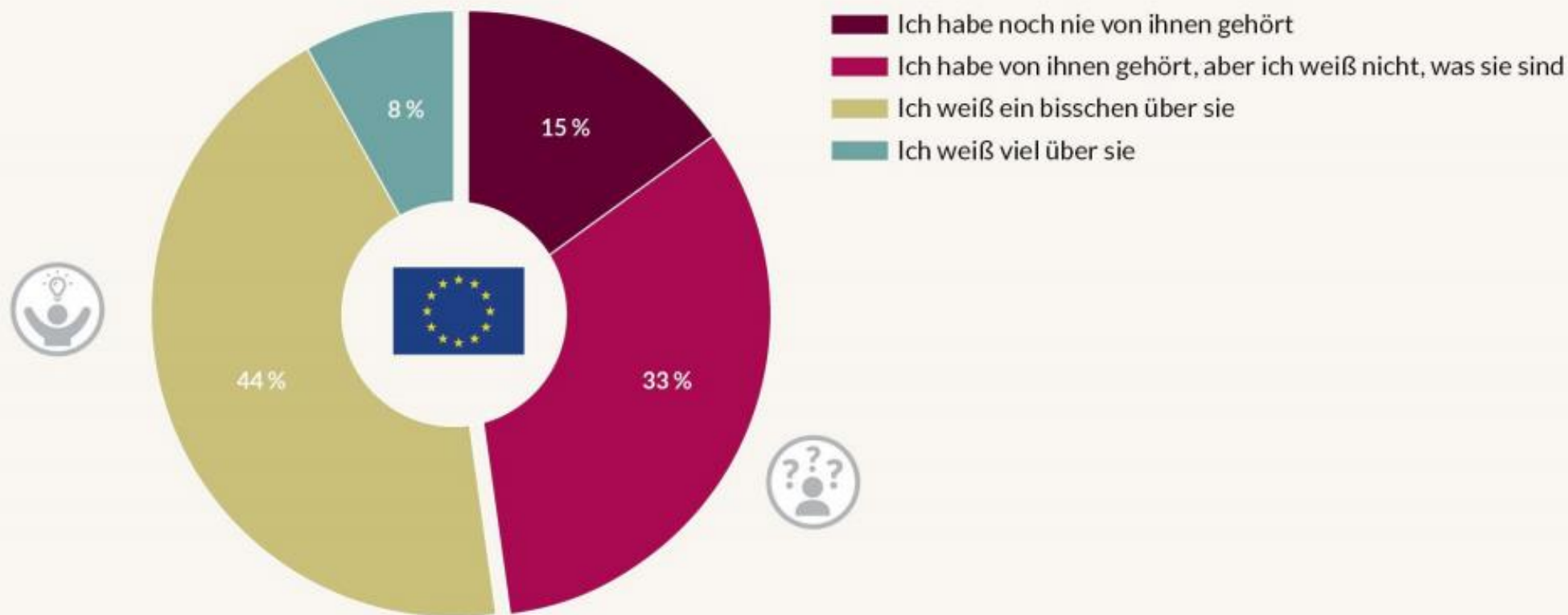


aus: Grzymek et al., 2019, S. 8
(veröffentlicht unter CC-Lizenzierung, hiesig eingesetzt
als Auszug aus dem Werk ohne Materialveränderung)²⁴



ABBILDUNG 1 Vertrautheit mit Algorithmen | EU

„Welche der folgenden Aussagen beschreibt am besten Ihre Vertrautheit mit Algorithmen?“



Quelle: Repräsentative Online-Umfrage von Dalia Research im Auftrag der Bertelsmann Stiftung, September 2018, n = 10.960 Befragte

| BertelsmannStiftung

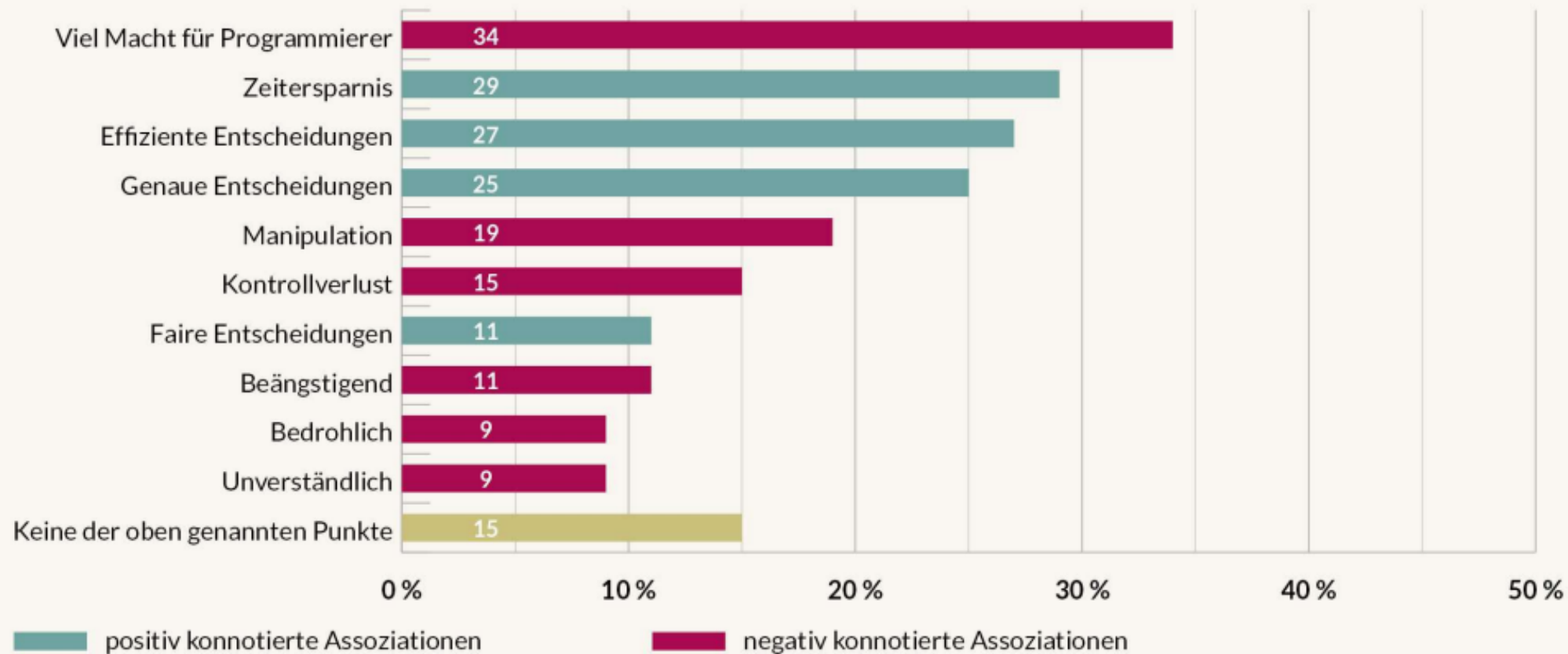
aus: Grzymek et al., 2019, S. 15
(veröffentlicht unter CC-Lizenzierung, hiesig eingesetzt
als Auszug aus dem Werk ohne Materialveränderung)

Grundlagen der Algorithmik – Grzymek et al. (2019): Was Europa über Algorithmen weiß und denkt



ABBILDUNG 4 Assoziationen zum Begriff Algorithmus | EU

„Was kommt Ihnen in den Sinn, wenn Sie das Wort ‚Algorithmus‘ hören?“



Quelle: Repräsentative Online-Umfrage von Dalia Research im Auftrag der Bertelsmann Stiftung, September 2018, n = 10.960 Befragte

| BertelsmannStiftung

aus: Grzymek et al., 2019, S. 21

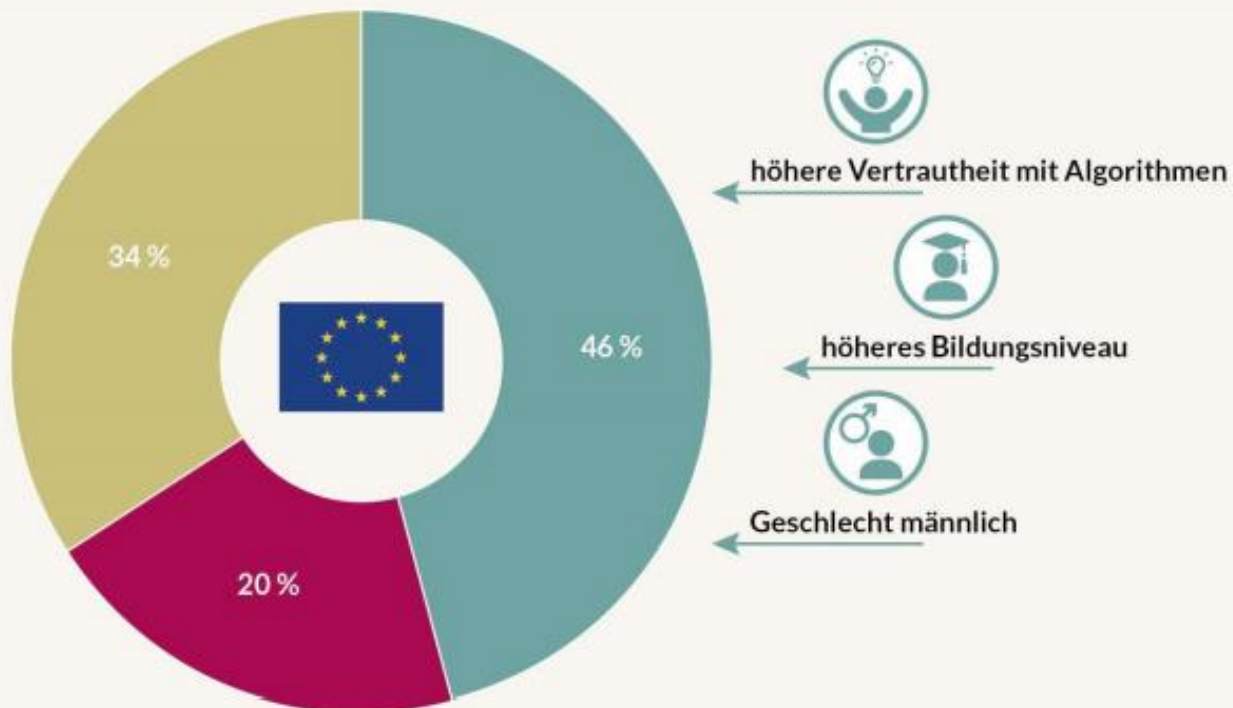
(veröffentlicht unter CC-Lizenzierung, hier eingesetzt
als Auszug aus dem Werk ohne Materialveränderung)



ABBILDUNG 6 Grundsätzliche Haltung zu algorithmischer Entscheidungsfindung | EU

„Glauben Sie, dass der Einsatz von Computeralgorithmen für die Entscheidungsfindung insgesamt mehr Vorteile oder mehr Probleme mit sich bringt?“

- Mehr Vorteile
- Mehr Probleme
- Ich weiß nicht



Quelle: Repräsentative Online-Umfrage von Dalia Research im Auftrag der Bertelsmann Stiftung, September 2018, n = 10.960 Befragte

| BertelsmannStiftung

aus: Grzymek et al., 2019, S. 23
(veröffentlicht unter CC-Lizenzierung, hiesig eingesetzt
als Auszug aus dem Werk ohne Materialveränderung)



„Die Menschen in Europa wissen wenig über Algorithmen. **48 Prozent der europäischen Bevölkerung wissen nicht, was ein Algorithmus ist.** Auch ist weniger als der Hälfte der europäischen Bevölkerung bekannt, dass Algorithmen bereits in vielen Lebensbereichen eingesetzt werden. Besonders niedrig sind dabei die Werte für Anwendungsfelder, wo die **Entscheidungen von Algorithmen potenziell folgenreich für die soziale Teilhabe sind, etwa bei der Kreditvergabe, der Bewerberauswahl und der medizinischen Diagnostik.** [...] Zudem tritt die europäische Bevölkerung Algorithmen **mit gemischten Gefühlen entgegen.** So assoziieren Europäerinnen und Europäer mit Algorithmen sowohl positive Aspekte wie Effizienz und Zeitersparnis als auch Negative wie Angst oder das Risiko der Manipulation. Insgesamt **überwiegt die optimistische Grundhaltung.**“



Querverweis auf den Film der bpb:
**„Algorithmisches Denken verstehen –
Interview mit Julia Kleeberger und Matthias Spielkamp“**

veröffentlicht unter der Lizenz CC-BY-SA über den Link:
<https://www.bpb.de/mediathek/265398/algorithmisches-denken-verstehen>



Querverweis auf die Aktion:
„IDEA QUICK SÖRT“

veröffentlicht unter der Lizenz CC-BY-NC-SA-4.0 über den Link:
idea-instructions.com/quick-sort/



- basierend auf dem informatischen Quick Sort
- ungeordnete Liste von Türmen (Zahlen), Problem: aufsteigend sortieren
- Algorithmisches Vorgehen:
 - Start: Würfeln zur Auswahl eines Startturms – hierdurch Einteilung der Elemente in zwei Teilmengen
 - Linie auf der Höhe des Turms zeichnen
 - Steine, die höher als Linie sind: kennzeichnen mit „nach rechts verschieben“
 - Steine, die niedriger als Linie sind: kennzeichnen mit „nach links verschieben“
 - Sortiere
 - Schleife: wiederhole und durchlaufe Schema für beide Teilmengen so lange, bis richtig sortiert – für höhere Türme und für niedrigere Türme
 - danach: Zusammenführung der Ergebnisse



Unter einem **Algorithmus** versteht man

- ein **Verfahren**, welches **eindeutig, endlich beschreibbar** und **mechanisch** ist und
- der **Lösung eines vordefinierten Problems** – bzw. der Lösung mehrerer ähnlicher Probleme, die derselben Problemklasse zugehören – dient. Hierfür umfasst der Algorithmus eine Reihe **präziser Anweisungen**.
- Zu jedem **Zeitpunkt des Verfahrens** muss der **Folgeschritt eindeutig** durch den vorangegangenen Schritt festgelegt.
- Nach der Eingabe der jeweiligen Daten und der Ausführung des Algorithmus **bricht das Verfahren nach endlich vielen Schritten** ab, es liefert das gesuchte Ergebnis, die Lösung des Problems.

vgl. Meyer 2012, S. 13 ff.



Eindeutigkeit

Ein Algorithmus muss in seiner Beschreibung **eindeutig** sein, er darf keine widersprüchliche Beschreibung haben.

Fintheit

Die Beschreibung des Algorithmus muss **endlich** sein, darf also nur eine begrenzte Anzahl von Zeilen umfassen und endlich viel Speicherplatz benötigen.

Ausführbarkeit

Jeder **Einzelschritt muss ausführbar** sein, die **Wirkung einer jeden Anweisung** muss eindeutig festgelegt sein.

vgl. Meyer 2012, S. 13 ff.



Terminierung

Nach endlich vielen Schritten muss der Algorithmus **enden und ein Ergebnis liefern**. Hierbei muss **jede mögliche Eingabe / jede Option berücksichtigt** sein.

Determiniertheit

Der Algorithmus muss **bei gleichen Voraussetzungen stets das gleiche Ergebnis liefern**.

Determinismus

Zu **jedem Zeitpunkt der Ausführung** besteht höchstens **eine Möglichkeit der Fortsetzung**. Der jeweilige Folgeschritt ist also eindeutig bestimmt.

vgl. Meyer 2012, S. 13 ff.



Beispiele aus der digitalen Anwendung:

- Partnersuche
- Google-Suche
- Werbeeinblendung
- Navigation / Wegeberechnung
(Dijkstra-Algorithmus, kürzester Weg)
- Bewerberauswahl
- Ampelschaltung
- ...

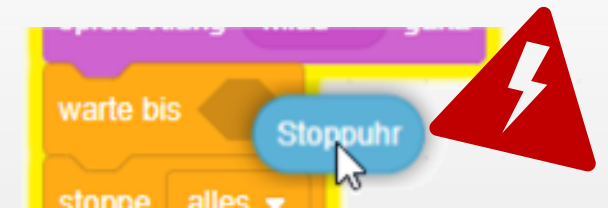
Gibt es evtl. Vorgehensweisen im Alltag, die Ähnlichkeiten zu algorithmischen Vorgehensweisen aufweisen, sich aber bzgl. der Eigenschaften ggf. unterscheiden?

Aktivitäten wie **Kochen, Backen, Zähneputzen oder der Weg zur Arbeit/Schule** weisen ebenfalls klar strukturierte Handlungsvorschriften auf.

Diese müssen und werden allerdings oftmals nicht ganz so genau den genannten Eigenschaften von Algorithmen entsprechen, sondern können an einigen Stellen auch abweichen und funktionieren trotzdem noch.



- Programmbibliothek
- fügt Web- und Softwareprodukten einen visuellen Code-Editor hinzu
- stellt einfache, ikonisierte, grafische Blöcke zur Darstellung von algorithmischen Programmierkonzepten bereit (Variablen, Schleifen etc.)
- syntaktische Korrektheit wird Editoren-seitig überprüft
- semantische Korrektheit wird spielerisch erfahren / nur syntaktisch korrekt zusammengehörige Blöcke können zusammengefügt werden





- Definieren der Ausgangslage und dem Ziel
- Entwickeln von möglichen Teilschritten, Abbildung dieser
- Formalisieren und Festhalten
- Bewerten der bisherigen Erkenntnisse, Überprüfung auf Korrektheit
- Testen anhand von Beispieldaten / Verifizieren bzw. das Begründen der korrekten Ausführung an bekannten Bedingungen
- Anwenden, Bewerten, Korrigieren, Verbessern, Weiterentwickeln inkl. der Überprüfung auf Effektivität



- **kreatives Denken ist erforderlich, um auf die Idee des Algorithmus zu kommen**
 - Problemzerlegung
 - Abstraktion
 - Mustererkennung
 - Anwendung von Algorithmen

- **Rückblick auf eigene Problemlösung und Algorithmen-Anwendung:**
 - Das Paritätsbit / Nutzung des Paritätsbits zur Fehlererkennung und -behebung
 - Bau einer Lego-Ente



- Problem als Barriere zwischen einem unbefriedigenden Ausgangszustand und einem erwünschten Zielzustand
- erfordert Suche nach einem (i. d. R. mehrstufigen) Lösungsweg*
- durch verschiedene **Merkmale** gekennzeichnet, hier am Beispiel der Definition von D. Dörner (1979):
 - „Bei der Problemlösung sind mehrere Variablen zu berücksichtigen.
 - Die Variablen sind keine unabhängigen Größen, sondern miteinander vernetzt.
 - Die Variablen und ihre Zusammenhänge sind nicht alle bekannt.
 - Das Zusammenspiel der Variablen unterliegt einer Eigendynamik, d. h. sie interagieren auch ohne Einwirken der Person.
 - Das Problem ist schlecht definiert: Der Anfangs- und Zielzustand oder die möglichen Lösungsalternativen sind nicht klar erkennbar.“ (Kerres 2018, S. 366)



- **Problemlösen** beschreibt eine **Lernhandlung**, bei der ein Anfangszustand in einen zu erzielenden Endzustand übertragen werden soll und die dabei durch bewusstes, intentionales Handeln und gedankliche Reflexion geprägt ist (vgl. Giest 2009, S. 79)
- „Problemlösekompetenz bedeutet, dass die Problemlöser einen **Lösungsweg suchen und sich auf den Lösungsprozess einlassen** müssen. Sie müssen bereits **Gelerntes aktivieren und organisieren**. Hierbei wird deutlich, dass beim systematischen Aufbau von Problemlösekompetenz das Lernen und Anwenden von heuristischen Regeln und das „**Eigenständig-Denken-Lernen**“ im Vordergrund steht. Die Kompetenz, Probleme zu lösen, brauchen Kinder, Jugendliche und Erwachsene demnach immer dann, wenn Lösungen nicht naheliegend oder offensichtlich sind und deshalb strategisches Vorgehen zur Lösungsfindung notwendig ist. Die Kompetenz, Probleme zu lösen, zeigt sich darin, dass über **geeignete Strategien zum Finden [von] Lösungsansätze[n] und -wege [n] verfügt** und über diese **Strategien reflektiert werden kann.**“ (vgl. Kipman 2020, S. 12)



- Im Kontext der heutigen, digital geprägten Welt ist Problemlösekompetenz höchstrelevant, denn:

„die Fähigkeit, über eine praktisch unendliche Menge von Objekten rational zu argumentieren, wird [...] mit zunehmender Komplexität der technischen Infrastruktur unserer Gesellschaft immer wichtiger, [...], um mit komplexen Systemen verständnisvoll umgehen [zu] können“ (Nievergelt 1999, S. 365).

vgl. Nievergelt, 1999



- vielfältige Ansätze vorhanden, hier am Beispiel des in der Literatur Meistverwendeten:
- Vierstufigkeit des Problemlöseprozesses nach Pólya (1995), vgl. bspw. Kipman 2020, S. 18 ff.; auf den außermathematischen Kontext adaptiert:

Rückschau / Bewertung der Problemlösung

- Wurde das Ziel erreicht?
- War der Problemlöseprozess effektiv?
- Besteht eine Möglichkeit zur Kontrolle (bspw. Ableitung über verschiedene andere Wege)?

Ausführen des Plans

- Sind die Lösungsschritte eindeutig und vollständig definiert?
- Sind die Lösungsschritte korrekt? Kann dies bewiesen werden?

Schritte des Problemlösens

Verstehen der Aufgabe / des Problems

- Was ist unbekannt?
- Was ist gegeben?
- Wie lautet die Bedingung?
- Was ist das Ziel?
- ...

Ausdenken eines Plans

- Sind **ähnliche Situationen** bzw. Problemstellungen bekannt?
- Wie kann das Problem in **Teilaspekte** gegliedert werden? Sind **Teilaspekte** der Problemstellung auf Basis vorhandenen Wissens **unmittelbar lösbar**?
- Wie kann das **weitere Vorgehen strukturiert** werden? Welche **einzelnen Schritte** müssen durchgeführt werden?
- Gibt es Dinge, die **mehrmals durchgeführt** werden müssen? (algorithmisch: Schleifen)
- Müssen Dinge **zwischenzeitlich abgefragt** und muss jeweils verschieden weiter verfahren werden? (algorithmisch: Wenn-Dann-Bedingungen)



Medienkompetenzrahmen:

„Der Bereich „Problemlösen und Modellieren“ verankert eine informatische Grundbildung als elementaren Bestandteil im Bildungssystem. Neben **Strategien zur Problemlösung** werden **Grundfertigkeiten im Programmieren** vermittelt sowie die **Einflüsse von Algorithmen und die Auswirkung der Automatisierung von Prozessen in der digitalen Welt reflektiert**.

Grundschul Kinder am Ende der Klasse 4 **erkennen algorithmische Muster und Strukturen in verschiedenen Kontexten**, z.B. bei Verkehrsschaltungen auf dem Schulweg, und können diese nachvollziehen und reflektieren. Sie **formalisieren und beschreiben Probleme** und entwickeln Problemlösestrategien. Dazu **planen und nutzen sie Algorithmen und Modellierungskonzepte** auch in einfachen Programmierumgebungen, z.B. bei Robotern, Microcontroller-Boards oder Programmier-Apps. Sie beurteilen die gefundenen Lösungsstrategien. Grundschul Kinder beschreiben und reflektieren die Einflüsse von Algorithmen auf die **digitalisierte Gesellschaft sowie die Auswirkungen der Automatisierung für die eigene Lebenswirklichkeit**, z.B. in Bezug auf Abläufe im Alltag.

Jugendliche am Ende der Sekundarstufe I erkennen algorithmische Muster und Strukturen in verschiedenen Kontexten, [...]“ MKR 2018a, S. 22f.

Problemlösung zudem verankert in: KMK 2016, „Problemlösen und Handeln“

MKR 2018a,b

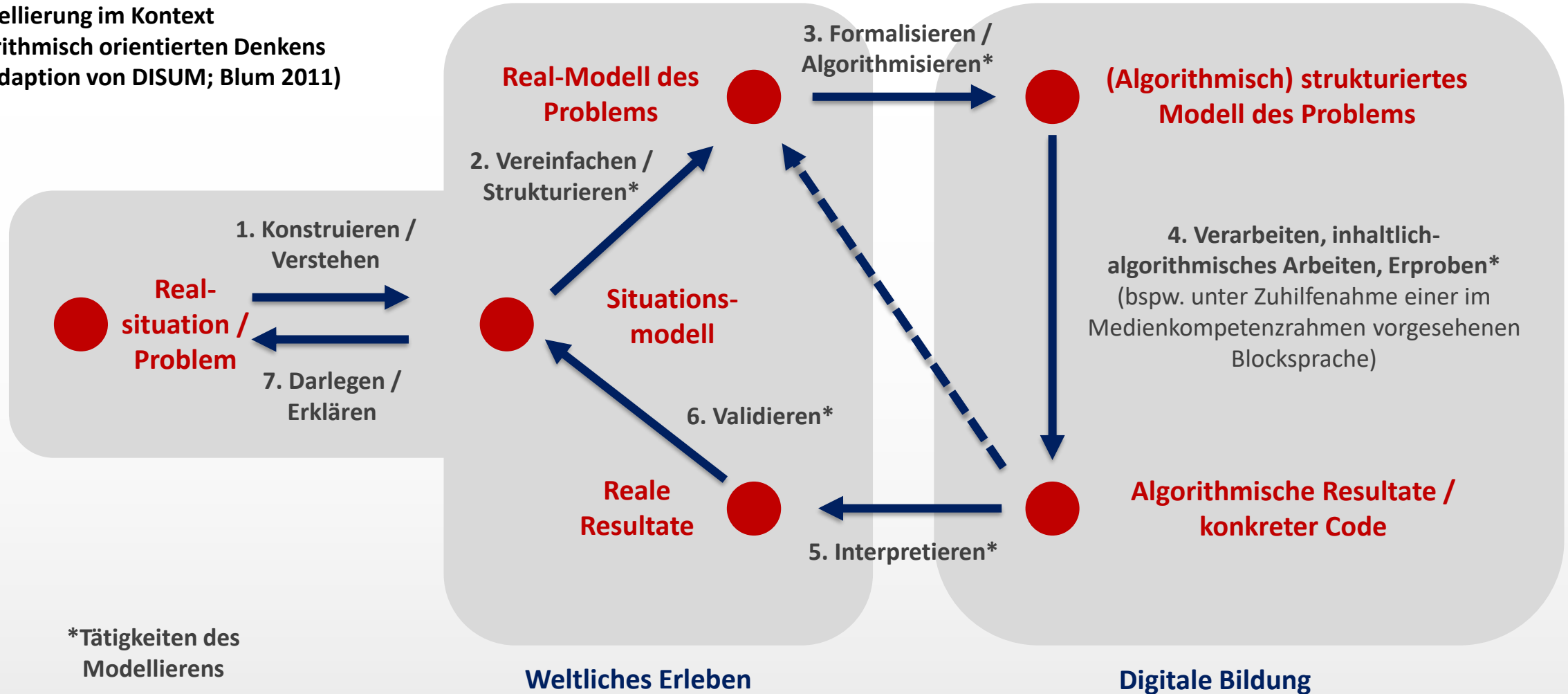


- Modellierung als ein Prozess zur Erstellung eines Modells als vereinfachtes Abbild der Wirklichkeit
- bekannte Modellierungskreisläufe:
 - mathematische Modellierung: Blum
 - informatische Modellierung: Humbert

Beide Modellierungskreisläufe werden zur fakultativen Betrachtung auf den folgenden zwei Folien auf unseren Kontext (computational thinking, Problemlösen mit Algorithmen) adaptiert dargestellt.



Modellierung im Kontext algorithmisch orientierten Denkens (in Adaption von DISUM; Blum 2011)



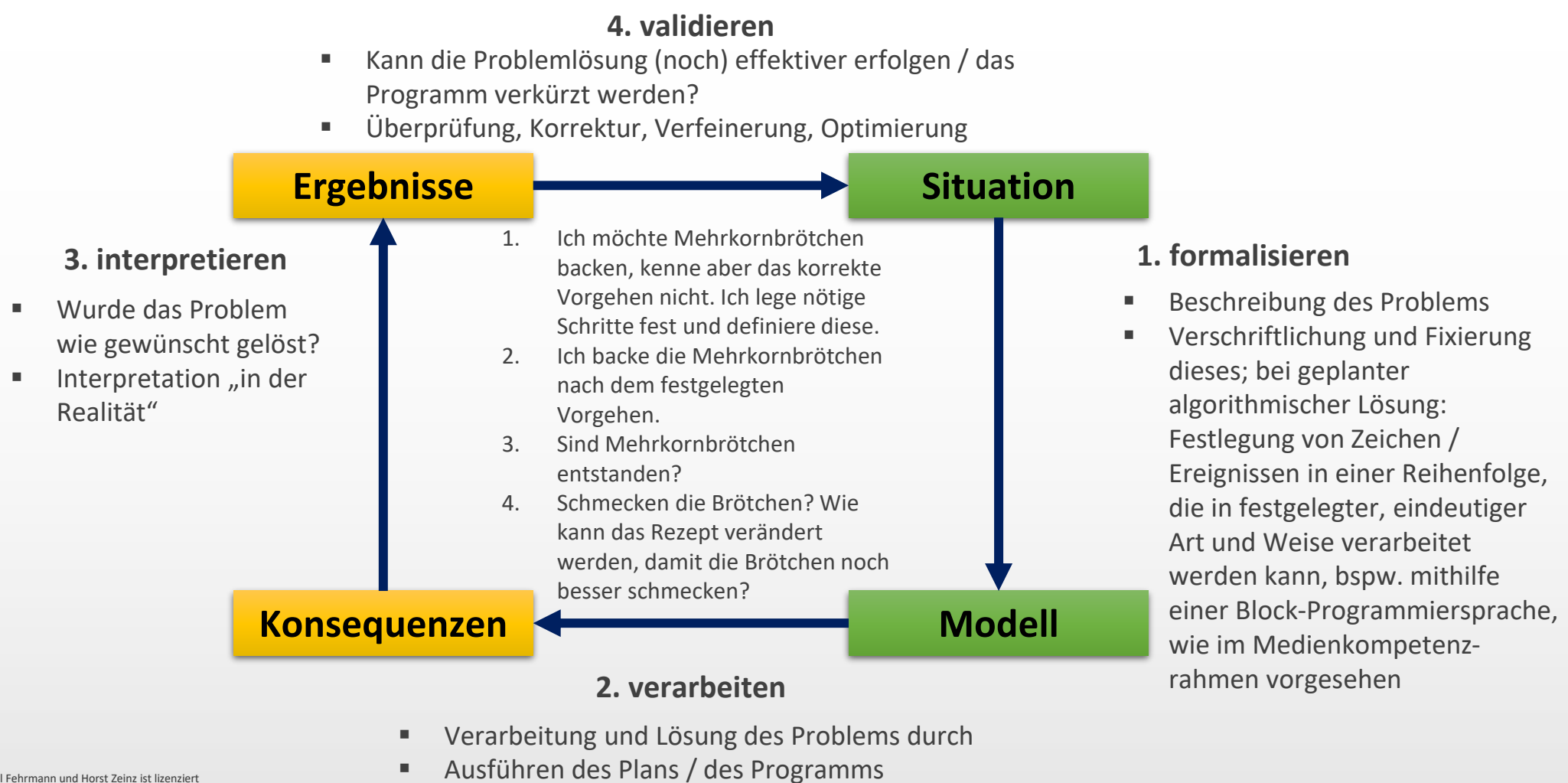
Erkunden von Realsituationen und Lösung von Sachproblemen mithilfe von allgemeiner digitaler Kompetenz, Problemlösekompetenz und der Kompetenz im Verstehen und Aufstellen von Algorithmen

Modellierung in Anlehnung an und Adaption des DISUM Modellierungskreislaufs von Blum (2011), vgl. bspw. Kaiser, Gabriele & Henn, Hans Wolfgang (2015): Werner Blum und seine Beiträge zum Modellieren im Mathematikunterricht. Wiesbaden: Springer, S. 17; auf den außermathematischen Kontext adaptiert





Lebensweltlicher Kontext
Algorithmischer Kontext



Modellierung in Anlehnung an und Adaption des informatischen Modellierungskreislaufs von Humbert (2006 - Didaktik der Informatik - mit Praxisproblemen Unterrichtsmaterial, 2., Aufl. Fäden der Informatik, Wiesbaden: B.G., Teubner Verlag. isbn: 978351-0112-9.) und das Beispiel der Roberta-Initiative, abrufbar unter: https://slidewiki.org/print/95582/_/95582/593764-2/



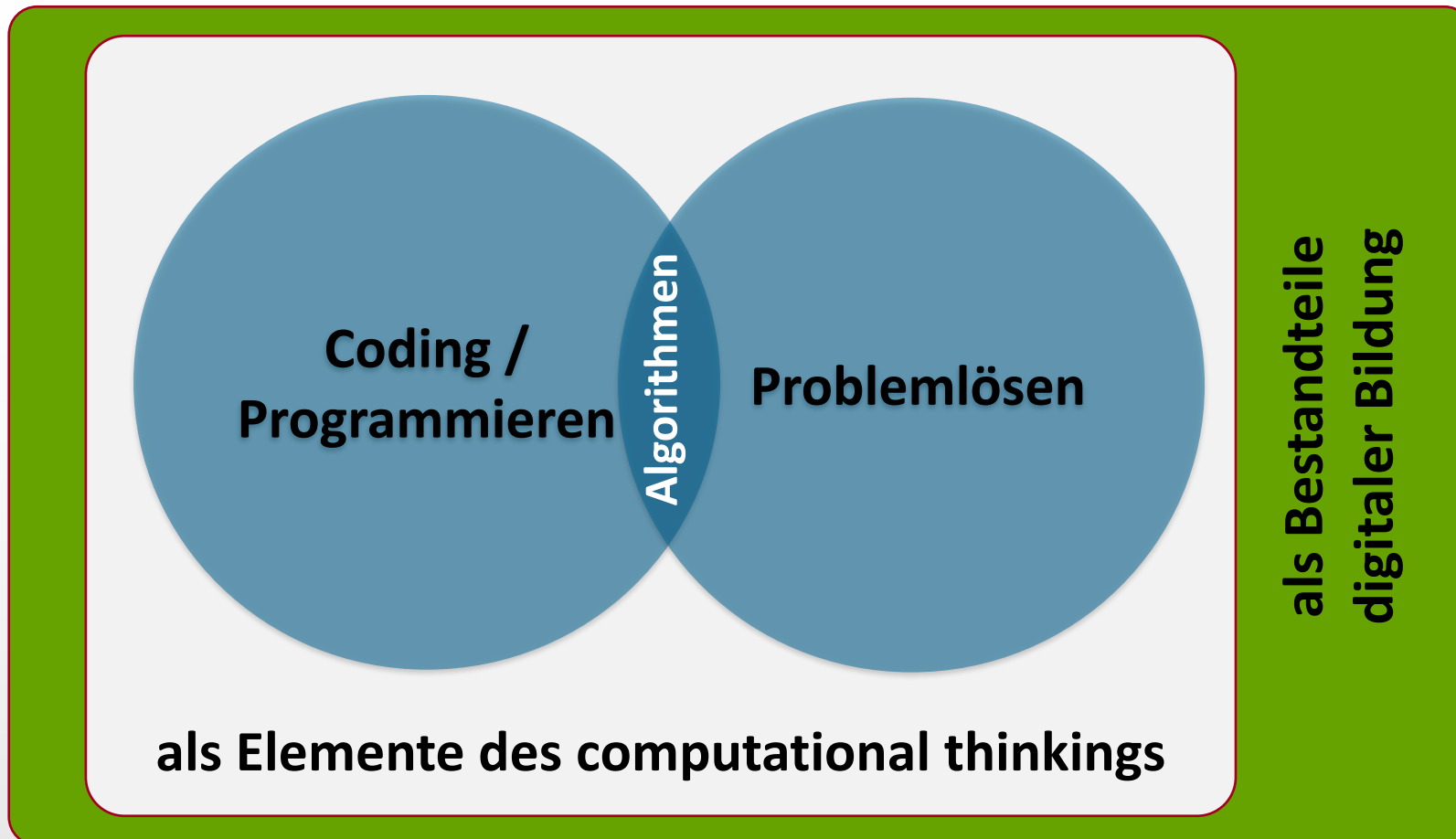
- **Programmieren bzw. algorithmisches Denken als Problemlösen**

- **Ziel: Verständnis entwickeln, ohne Programmieren zu können**

(vgl. auch Studienergebnisse aus ICILS 2018 / Eickelmann et al., 2019)



- Zusammenhang zum computational thinking und zu digitaler Bildung



Programmieren bzw. algorithmisches Denken als Problemlösen

Der „computational thinker“



Ist mein Lösungsweg der effizienteste Weg?
Habe ich die schnellste Lösung gefunden?
Braucht mein Lösungsweg die wenigsten Ressourcen?
Führt mein Lösungsweg zur richtigen Antwort?
Ist mein Lösungsweg anwendbar auf weitere Probleme?



- interaktives Informatiksystem des Lernroboters als Element von Spielumgebungen (vgl. Nievergelt 1999, S. 365 ff.)
- vielseitiges, funktionales Werkzeug mit umfangreichen Sensoren und Aktoren des Messens, Steuerns und Regelns (vgl. Nievergelt 1999, S. 365 ff.)
- ermöglicht „eingeschränkte Formen des Programmierens in der einfachsten Gestalt“ (Nievergelt 1999, S. 368), begleitet von einfachen Modellierungen



- Lernroboter bieten einen **didaktisch reduzierten Einstieg**, ohne dass die Schüler*innen über Vorkenntnisse verfügen müssen, und schaffen rasche Erfolgserlebnisse durch geringe Einstiegshürden (vgl. Stiftung Haus der kleinen Forscher 2018, S. 301).
- Lernroboter bieten **verschiedene Zugänge** zu den Inhaltsbereichen der Programmierung sowie der Problemlösung (u. a. grafische Codebausteine, Systeme zur Fehlerkorrektur, begrenzte, einfache, strukturierte und jederzeit erweiterbare Programmlogik).
- Lernroboter sind **thematisch vielseitig einsetzbar** (fachspezifisch sowie fachübergreifend).



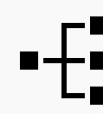
- Abstrakte Algorithmen werden durch die Verwendung von Tastenbefehlen oder Blocksprache haptisch fassbar und konkret.
- motivationssteigernde, gestalterische Erarbeitung und Ausgestaltung ist möglich
- Praxisnähe, direkte Problemerkennung und -lösung auf Basis der direkten Rückmeldung / Ausführung des Algorithmus
- Erfahrungen im problemlösenden Denken werden gesammelt.

vgl. / Empfehlung zum Weiterlesen: Romeike 2017, Brandhofer 2017b, 2017c

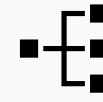


„Program or be programmed?“ (Douglas Roshokoff, Buchtitel 2011)

- kein Programmieren, um Programmieren zu können, sondern Programmieren mit Lernrobotern, um **programmierte Geräte reflexiv und mündig zu nutzen**
- Ziel ist das **Verstehen und Reflektieren**, nicht das eigentliche Programmieren in fachlich hoher Auflösung
- Ablösen einer überwiegend passiven Nutzung digitaler Angebote hin zu einer **mitgestaltenden, durchschauenden und problemlösungsorientierten Anwendung**
- → **computational thinking** → **Digitale Kompetenz**



- Mitchel Resnick (*12. Juni 1956), US-amerikanischer Professor für Lernforschung (Learning Research) und Direktor des Okawa Center und Direktor der Lifelong Kindergarten-Group am MIT Media Lab; 2007 Entwicklung der Programmiersprache Scratch
- entwickeltes Kompetenzmodell: „low floor – wide walls – high ceiling“



high ceiling

nach oben offen, keine obere Grenze (Does it scale with growth?)

- Roboter „wachsen vom Anspruch her mit“ (grafische Programmieroberflächen bis hin zur Anwendung von Programmiersprachen wie javascript)
- Komplexität der mithilfe des Systems zu lösenden Problemstellungen und Lösungsmöglichkeiten ist unbegrenzt

wide walls

verschiedenste Zugangsweisen (How inclusive is it? How many different use-cases does it serve?)

- thematisch vielseitig einsetzbar (u.a. Genderaspekt, Verbindung verschiedener Kompetenzbereiche), auch durch verschiedene Arten der Programmierung
- Einsatz für unterschiedliche Frage- und Problemstellungen im Kontext verschiedener Fachbereiche / Fächer

low floor

leichter Einstieg (How easy is it to learn?)

- keine Einstiegshürden
- keine Vorkenntnisse nötig
- schnelle Erfolgserlebnisse



Doebeli's BFG ([Link](#))



- Ergänzung der kollaborativen Arbeitsbereiche (Glossar für Fachbegriffe)
- Video / Roboter-Einsatz im Alltag (ZDF, Planet Schule): Medizintechnik
(Video steht im moodle bereit)
- Vertiefungstext: Kleinhanß, Christian (2017): Einen Rechner braucht es nicht. Unterrichtsideen für erste Schritte in Richtung Programmieren. In: Computer und Unterricht (107), S. 19-21.



Literaturverzeichnis

- Brandhofer, Gerhard (2017b): Code, Make, Innovate! Legitimation und Leitfaden zu Coding und Robotik im Unterricht. Ein Pladoyer für einen Blick hinter die Kulissen des Digitalen, für Coding, Computational Thinking, Robotik und Making in der Schule. In: R&E-Source - Open Online Journal for Research and Education. Online verfügbar unter <https://journal.ph-noe.ac.at/index.php/resource/article/view/348/422>, Tag des letzten Zugriffs: 14.01.2020.
- Brandhofer, Gerhard (2017c): Programmieren in der Schule im Zeitalter der Digitalität. In: Schule aktiv! (Oktober), S. 4–5. Online verfügbar unter https://www.phdl.at/fileadmin/user_upload/5_Ueber_uns/2_Institute/Medienbildung/Publikationen/coding_2017.pdf, Tag des letzten Zugriffs: 14.01.2020.
- Buller, Laura; Gifford, Clive; Mills, Andrea (2019): Roboter. Wie funktionieren die Maschinen der Zukunft? München: DK.
- Eickelmann, Birgit; Bos, Wilfried; Gerick, Julia; Goldhammer, Frank; Schaumburg, Heike; Schwippert, Knut; Senkbeil, Martin & Vahrenhold, Jan (2019): ICILS 2018 Deutschland - Computer- und informationsbezogene Kompetenzen von Schülerinnen und Schülern im zweiten internationalen Vergleich und Kompetenzen im Bereich Computational Thinking. Münster: Waxmann. Zugriff über URL: https://kw.uni-paderborn.de/fileadmin/fakultaet/Institute/erziehungswissenschaft/Schulpaedagogik/ICILS_2018__Deutschland_Berichtsband.pdf, Tag des letzten Zugriffs: 12.11.2019.
- Giest, Hartmut (2008): Experimentieren und Problemlösen als Lernhandlungen. In: Grundschulunterricht Sachunterricht (2), S. 4–9. Online verfügbar unter <https://www.oldenbourg-klick.de/zeitschriften/grundschulunterricht-sachunterricht/2008-2/experimentieren-und-problemlosen-als>, Tag des letzten Zugriffs: 14.01.2020.

Literaturverzeichnis

- Giest, Hartmut (2009): Zur Didaktik des Sachunterrichts. Aktuelle Probleme, Fragen und Antworten. Potsdam: Universitätsverlag Potsdam. Zugriff über URL: https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/3197/file/giest_didaktik.pdf, Tag des letzten Zugriffs: 31.05.2019.
- Grzymek, Viktoria & Puntschuh, Michael (2019): Was Europa über Algorithmen weiß und denkt. Ergebnisse einer repräsentativen Bevölkerungsumfrage. Hg. v. Bertelsmann Stiftung. Gütersloh. Online-Bezug über URL: <https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/WasEuropaUEberAlgorithmenWeissUndDenkt.pdf>, Tag des letzten Zugriffs: 15.11.2019.
- Kerres, Michael (2018): Mediendidaktik: Konzeption und Entwicklung mediengestützter Lernangebote. Berlin: Walter de Gruyter GmbH.
- Kipman, Ulrike (2020): Problemlösen. Begriff – Strategie – Einflussgrößen – Unterricht – (häusliche) Förderung. Wiesbaden: Springer-Gabler.
- Köpp, Sabine; Kuhlen, Britta & Voll, Sabine (2018): Mein Medienheft 3 / 4 – Digitale Medien. Stuttgart, Leipzig: Ernst Klett.
- Kreis Soest, Medienzentrum (2018): Methodenkatalog zum Medienkompetenzrahmen NRW. Soest: Kreis Soest. Online-Bezug über URL: https://www.kreis-soest.de/bildung_integration/bildung/medienzentrum/materialien/materialien.php.media/386821/Methodenkatalog_web.pdf, Tag des letzten Zugriffs: 31.10.2019.

Literaturverzeichnis

- Nievergelt, Jürg (1999): Roboter programmieren - ein Kinderspiel - Bewegt sich auch etwas in der Allgemeinbildung? In: Informatik Spektrum, 22.10.1999, S. 364-375. Bezug über URL: http://www.johanneum-lueneburg.de/dokumente/upload/Nievergelt_RoboterProgrammierenEinKinderspiel.pdf, Tag des letzten Zugriffs: 31.05.2019.
- Oubbati, Mohamed (2007): Robotik. Skript zur Vorlesung. Ulm: Universität Ulm. Online-Bezug über URL: https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.130/Arbeitsgruppen/Robotics/Robotik/Robotik-Skript_07-08.pdf, Tag des letzten Zugriffs: 20.11.2019.
- Resnick, Mitchel; Robinson, Ken (2017): Lifelong Kindergarten. Cultivating creativity through projects, passion, peers, and play. Cambridge, Massachusetts, London: The MIT Press.
- Romeike, Ralf (2017): Wie informatische Bildung hilft, die digitale Gesellschaft zu verstehen und mitzugestalten. In: Eder, Sabine; Mikat, Claudia; Tillmann, Angela (Hrsg.): Software takes command – Herausforderungen der „Datafizierung“ für die Medienpädagogik, in: Theorie und Praxis, S. 105-118. München: kopaed. Bezug über URL: https://computingeducation.de/pub/2017_Romeike_GMK2016.pdf , Tag des letzten Zugriffs: 15.11.2019.
- Wüst, Klaus (2004): Grundlagen der Robotik. Skript zur Vorlesung. Gießen: Technische Hochschule Mittelhessen. Online-Bezug über URL: <https://homepages.thm.de/~hg6458/Robotik/Robotik.pdf>, Tag des letzten Zugriffs: 20.11.2019.

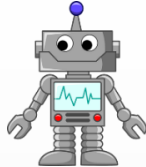

Urheber-Nachweis bei Grafiken

Diese Folie gehört zum Material und darf nicht entfernt werden.

Sofern die in der Präsentation abgebildeten Grafiken einer Urheberrechtseinschränkung unterliegen oder im direkten Projektkontext entwickelt wurden, wurde die Quelle der Entlehnung unter- oder oberhalb der Grafik vermerkt. Sofern kein Vermerk an der Grafik vorliegt, wurde diese

- vom Autor der Präsentation selbst erstellt oder
- dem Portal pixabay.com im Rahmen einer Pixabay-Lizenz entnommen – diese Grafiken unterliegen damit keinem Kopierrecht und können kostenlos für kommerzielle und nicht kommerzielle Anwendungen in digitaler oder gedruckter Form ohne Bildnachweis oder Quellenangabe verwendet werden (Bildliste siehe nachfolgende Folie).
- Einzelne Infografiken können zudem aus kostenfreien und unter der Bedingung der Rückverlinkung auf den Anbieter freigegebenen Folien der Portale presentationload.de und smiletemplates.com entstammen. Die vom Anbieter geforderte Rückverlinkung wird hiermit umgesetzt. Weitere Infografiken können zudem aus dem Office-Integrierten Piktogramm-Set entstammen.

Urheber-Nachweis bei Grafiken | pixabay-Bildliste

Bild	Titel	Urheber	Link	Lizenz	Ursursungsportal
	Android	OpenClipart-Vectors	https://pixabay.com/de/vectors/android-k%C3%BCnstliche-doodle-roboter-159109/	Pixabay License, freie kommerzielle Nutzung, kein Bildnachweis nötig	pixabay.com
	Luftbild Kamera Drohne Flucht	Pexels	https://pixabay.com/de/photos/luftbild-kamera-drohne-flucht-1866742/	Pixabay License, freie kommerzielle Nutzung, kein Bildnachweis nötig	pixabay.com

Weitere Informationen zum Projekt „Lernroboter im Unterricht“ finden Sie fortlaufend unter www.wwu.de/Lernroboter/.

Lernroboter

im Unterricht

Das Projekt „Lernroboter im Unterricht“ wird als „Leuchtturmprojekt 2020“ gefördert durch die



**UNIVERSITÄTS
GESELLSCHAFT
MÜNSTER**



Herzlichen Dank
für Ihre Aufmerksamkeit !