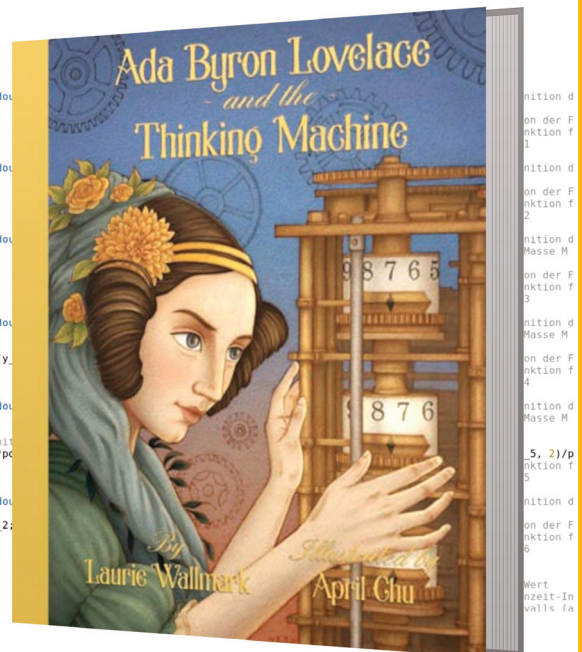


Schwerpunktthema Algorithmisches Denken und Programmieren

lernen & lehren

Elektrotechnik – Informationstechnik
Metalltechnik – Fahrzeugtechnik

```
11 #include <stdio.h>
12 #include <math>
13
14 double f_1(double y_1, dou
15     double wert;
16     wert = y_4;
17     return wert;
18 }
19
20 double f_2(double y_1, dou
21     double wert;
22     wert = y_5;
23     return wert;
24 }
25
26 double f_3(double y_1, dou
27     double M = 1;
28     double wert;
29     wert = y_6;
30     return wert;
31 }
32
33 double f_4(double y_1, dou
34     double M = 1;
35     double wert;
36     wert = 2*M*y_4*y_5/(y
37     return wert;
38 }
39
40 double f_5(double y_1, dou
41     double M = 1;
42     double wert;
43     // Eigentliche Definit
44     wert = -2*pow(M, 2)*p
45     return wert;
46 }
47
48 double f_6(double y_1, dou
49     double wert;
50     wert = -2*y_5*y_6/y_2;
51     return wert;
52 }
53
54 int main(){
55     double tau;
56     double a = 0;
57     double b = 800;
```



Ist Software Text oder Maschine?

Algorithmisches Denken und Programmierung

Programmierenlernen unter Berücksichtigung der Cognitive Load Theory (CLT)

Unterrichtspraxisprojekt: Betriebsdatenvisualisierung mittels Node-RED und MQTT

Fundamentale Kompetenzen der industriellen Steuerungstechnik

Grundkurs Robotik am OSZ TIEM Berlin Spandau

Usability in der Anwendungsentwicklung

Berufliches Lehren in der Krise?

Verhindern Lernmanagementsystem, Künstliche Intelligenz und Distance Learning einen produktiven Lehr-Lern-Prozess?



Am 8. und 9. März 2024 fand unter dem Titel „Berufliches Lehren in der Krise?“ am Berufsschulcampus Unstrut/Hainich in Mühlhausen die 33. Fachtagung der BAG Elektrotechnik, Informationstechnik, Metalltechnik und Fahrzeugtechnik statt. Trotz des Streiks der Lokführer fanden sich Akteurinnen und Akteure aus allen Bereichen der Berufsbildung zusammen, um sich mit der Fragestellung auseinanderzusetzen, inwieweit Lernmanagementsysteme, Künstliche Intelligenz und „Distance Learning“ einen produktiven Lehr-Lernprozess unterstützen oder verhindern.

Teilweise zeitgleich zur 13. Fachtagung des Arbeitskreises Versorgungstechnik, welche mit Teilnehmerinnen und Teilnehmern vom 7. bis zum 8. März ebenfalls in den Räumen des Berufsschulcampus mit dem Titel „Energie-wende – Auswirkungen auf die Ausbildung im Handwerk SHK“ stattfand, startete die Fachtagung der BAG Elektrometall bereits morgens mit einem Pre-Workshop unter der Moderation von Nikolaus Steffen von der Europa-Universität Flensburg.

Im anschließenden Teil der Fachtagung wurden neben den klassischen Formen von Workshops weitere Formate angeboten:

Das erste hieß **Young Teachers**.

So fand zum ersten Mal ein Symposium „Young Teachers“ statt. Dabei wurde eine neue Initiative zur Vernetzung von (angehenden) Lehrkräften vorgestellt. Angesprochen sind Studierende, Lehrkräfte im Vorbereitungsdienst sowie alle Lehrkräfte, die Interesse an diesem Format haben und daran teilnehmen möchten.

Der von Andreas Lindner initiierte **Markt der Möglichkeiten** war das zweite innovative Format.

Hier fanden sich drei Gruppen von Lehrerinnen und Lehrern ein, die Einblicke in die digitale Umsetzung von Unterrichtskonzepten ausstellten. Neben der digitalen Abbildung der Unterrichtskonzepte in Lernmanagementsystemen, die vom Berufskolleg in Hilden im Überblick und der Berufsschule für Fertigungstechnik in München am Beispiel der Harmonisierung zweier Lernfelder im zweiten Ausbildungsjahr der Industriemechaniker gezeigt werden konnten, wurde noch die Darstellung eines Fahrzeugs, das anhand mehrerer digitaler Fotos in einem 360°-Rundgang von den Auszubildenden erkundet werden kann, von einer Arbeitsgruppe der TU Berlin präsentiert.

Mitgliederversammlung und Tagungsfest

Der Tag endete im Anschluss an die Mitgliederversammlung mit dem Tagungsfest im „1. Deutschen Bratwurstmuseum“. Den Auftakt bildete eine äußerst unterhaltsamen Führung durch das Museum, bei der sich die Teilnehmerinnen und Teilnehmer von der weltweit herausragenden Stellung der Thüringer Bratwurst überzeugen und genug Appetit für die anschließend aufgetischten Spezialitäten entwickelt konnten.

Zukunftswerkstatt „Berufliches Lehren in der Krise?“

Das für BAG-Fachtagungen neuartige Format der Zukunftswerkstatt wurde von den Tagungsteilnehmenden gut angenommen. Die Teilnehmenden kamen aus Schulleitungen, Universitäten sowie berufsbildenden Schulen, wie bei der zunächst durchgeführten Vorstellung entlang von Leitfragen ersichtlich wurde.

Workshop 1: Fachschule für Technik vor neuen Herausforderungen.

Workshop 2: Ersetzen digitale Formate die Lehre in der Berufsbildung?

Workshop 3: Berufspädagogen und -pädagoginnen - Berufseinstieg ohne Qualifikation?

Schlussvortrag

Veröffentlichungen und Ausblick

Zum Jahresende wird es außerdem in der wbv-Reihe „Berufsbildung, Arbeit und Innovation“ eine Veröffentlichung geben, in der diese und weitere Beiträge in ausführlicher Form nachgelesen werden können

Einen ausführlichen Überblick über alle Plenums- und Workshopbeiträge bieten die Abstracts auf der BAG-Homepage (www.bag-elektrometall.de).

Inhalt

SCHWERPUNKT: ALGORITHMISCHES DENKEN UND PROGRAMMIEREN

Editorial

- 46 Ist Software Text oder Maschine?
Axel Grimm/Nikolaus Steffen

Schwerpunkt

- 48 Algorithmisches Denken und Programmierung - Allgemeinbildende Informatik an der Schnittstelle zur gewerblich-technischen Berufsbildung
Timo Hardebusch/Sven Jacobs/Steffen Jaschke
- 54 Programmierenlernen unter Berücksichtigung der Cognitive Load Theory (CLT)
Peer Stechert

Praxisbeiträge

- 62 Unterrichtsprojekt: Betriebsdatenvisualisierung mittels Node-RED und MQTT
Jesse Schlüter/Nikolaus Steffen
- 69 Integration von OT und IT - PLCnext Technology
Götz Jäckel
- 74 Fundamentale Kompetenzen in der industriellen Steuerungstechnik – Versuch einer didaktischen Strukturierung der Computer-Bedientechniken im Maschinenbau und angrenzender Bereiche
Andreas Lindner
- 82 Hinweis auf Lern-/Lehrmaterial zum Themenschwerpunkt: Grundkurs Robotik am OSZ TIEM Berlin Spandau
Axel Grimm/Nikolaus Steffen
- 83 Usability in der Anwendungsentwicklung
Christoph Dreeßen/Axel Grimm

Ständige Rubriken

- I–IV BAG aktuell 2/2024
- 88 Verzeichnis der Autorenschaft
- U3 Impressum

Beiträge in der Zeitschrift lernen & lehren sollen diskriminierungs- und gendersensibel verfasst sein. Das generische Maskulinum wird daher nicht verwendet. Solange im amtlichen Regelwerk der deutschen Rechtschreibung keine verbindlichen Sonderzeichen (Genderstern, Unterstrich, Doppelpunkt etc.) zur Einbeziehung nicht-binärer Geschlechtsidentitäten vorgegeben sind, wird auf deren Verwendung in der Zeitschrift verzichtet. Stattdessen werden konsequent andere Möglichkeiten zur Vermeidung diskriminierender Sprache und zur Sichtbarmachung von Diversität genutzt. Dazu zählt primär die Nutzung von geschlechtsneutralen Formulierungen. Wenn dieses nicht möglich ist, werden ausschließlich Beidnennungen (z. B. Schülerinnen und Schüler) verwendet.



Editorial

Ist Software Text oder Maschine?



AXEL GRIMM



NIKOLAUS STEFFEN

In einer Welt, die zunehmend von modernster Technologie und digitaler Innovation geprägt ist, können die Kompetenzen, algorithmisch Denken und Programmieren zu können, ein entscheidender Bestandteil für den persönlichen und beruflichen Erfolg und die gesellschaftliche Teilhabe sein. Computer und Software sind in nahezu allen Bereichen des Lebens präsent – im Privaten sowie im Beruflichen. Die KMK griff algorithmisches Denken und das Programmieren können u. a. in der Strategie „Bildung in der digitalen Welt“ (2016) auf und erhob dieses zumindest implizit zum „Bildungsgegenstand“ – einführend in der Allgemeinbildung, vertiefend in der beruflichen Bildung. Der Kompetenzerwerb von „digitalen“ Kompetenzen soll nach KMK-Strategie integrativ in allen Fachcurricula erfolgen, unabhängig von der noch anhaltenden, länderinternen und -übergreifenden Diskussion, „Informatik“ als eigenständiges Fach einzuführen. Das reflektierte Beherrschen des „Digitalen“ wird zu einer unverzichtbaren Schlüsselkompetenz für alle erklärt, auch und insbesondere für alle Lernenden in berufsbildenden Schulen und betrieblichen Ausbildungsstätten.

Worin liegt die besondere Relevanz des algorithmischen Denkens und Programmieren Könnens? Durch Programmieren werden Anweisungen erstellt, die von Computern ausgeführt werden, um bestimmte Aufgaben zu erledigen oder Probleme zu lösen. Diese Anweisungen werden in einer speziellen Sprache verfasst, die der Computer versteht, und in Form von Code geschrieben. Programmieren lernen bedeutet daher die Sprache zu erwerben und zu verstehen, durch die die Technologie funktioniert, und gegebenenfalls durch eigenes Programmieren technische Systeme zu verändern und mitzugestalten.

Doch als was lassen sich Computerprogramme eigentlich verstehen, sind sie „Text“ oder „Maschine“?

„Software ist naturgemäß weder das eine (Text) noch das andere (Maschine). Denn wenn ich ein Stück Software nehme und auf den Tisch lege und den Einschaltknopf für die Maschine suche, schon dann scheitere ich. Wenn ich deswegen anfangen, die Schwärzungen auf Papier zu lesen und vorzutragen, wie ich das mit Texten zu tun gewohnt bin, so scheitere ich ein zweites Mal. Software ist eben Software, zunächst und vor allem. Erst wenn wir ihr diese ihre Eigenständigkeit und Besonderheit zugestehen, kommen wir in die Lage, die Funktionen genauer zu beschreiben, die mit ihr aufgekomen sind“ (NAKE 2001, S. 737).

Diese Besonderheit von Software, die FRIEDER NAKE hervorhebt, und ihr Stellenwert an der Schnittstelle zwischen „Mensch und Maschine“ prägen und verändern nach wie vor Arbeits- und Bildungsprozesse in allen Bereichen. Die ehemals „mechanisierte“ Maschine ist übergegangen in die „computerisierte“ Maschine, mit der Verlagerung von Funktionen aus der anschaulichen, „begreifbaren“ Hardware in die Blackbox. Die Funktion der Blackbox lässt sich ohne ein Basisverständnis der Programmierung, unter Anwendung von algorithmischem Denken, nicht mehr erschließen.

Algorithmisches Denken, Computational Thinking oder auch informatisches Denken sollen hier nicht in ihren Spezifika unterschieden werden, sondern es soll ganz im Gegenteil darauf hingewirkt werden, dass dies als ein gemeinsamer Denkansatz verstanden werden kann, der trotz eventueller Divergenz im Detail für den Bildungsbereich ein hohes Maß an

Konvergenz aufweist. Es geht nämlich dabei nicht primär um das „produktorientierte“ Erlernen von Programmiersprachen mit ihrer je eigenen Syntax, Grammatik und Semantik und dann ein möglichst spezifisches Coden, sondern auch und insbesondere um die „konzeptorientierte“ Entwicklung von Denkweisen, die Problemlösungen durch Algorithmen und abstraktes Denken ermöglichen. Hieran kann dann u. a. auch der Anschluss zu wichtigen, „sicherheitsrelevanten“ Fragen erfolgen, wie z. B. an die Frage, warum eine Software nie umfassend sicher sein kann. Die Frage der Anlagensicherheit erweiterte sich mit der aufkommenden und jetzt flächendeckenden digitalen Vernetzung von einer Safety- zunehmend zu einer Security-Problematik, was sich nach Einschätzung beider Autoren noch nicht hinreichend in ihrer Relevanz in den Bildungsplänen aller gewerblich-technischen Berufe wiederfindet.

Die fast allen bekannte Torsteuerung, die so gerne in der Automatisierungstechnik aber auch in der Informationstechnik/Informatik Gegenstand beruflichen Unterrichts ist und bspw. durch Automatikbetrieb, Handbetrieb und Tippbetrieb durchaus komplex werden kann, lässt sich praktisch veranschaulichen und in ihren Funktionen und Abläufen durch die Lernenden in Prosa beschreiben. Ohne dass auch nur ein Codeschnipsel programmiert werden muss, kann durch Problemzerlegung, Mustererkennung und Abstraktion, unter Zuhilfenahme der fünf Planungselemente der Steuerungstechnik (Funktionsbeschreibung, Technologieschema, Ablaufschema, SPS-Beschaltung und Symboltabelle) auf algorithmisches Denken hingearbeitet werden, indem zunächst komplexe Probleme in kleinere, leichter handhabbare Teilaufgaben zerlegt werden. Danach werden Muster identifiziert, die helfen, Probleme zu verstehen und Lösungen zu entwickeln. Um sich auf die wesentlichen Aspekte eines Problems zu konzentrieren, erfolgt eine Reduzierung der Details ggf. mit einer Fokussierung auf das singuläre „Basissystem“ Torsteuerung. Letztlich lässt sich durch das gedankliche oder verschriftlichte Entwickeln von Schritten oder Anweisungen zur Lösung des Problems, die in einer bestimmten Reihenfolge ausgeführt werden müssen, algorithmisches Denken entwickeln und die Torsteuerung mit den drei Betriebsarten technologie- und syntaxunabhängig durchdringen, gedanklich realisieren und letztendlich (z. B. in GRAFCET) allgemeingültig in verständlicher Form visualisieren. Daran anschließend ist es wichtig, die Systemgrenze zu erweitern hinsichtlich der Berücksichtigung von potentiellen Schnittstellen nach „außen“ in die Vernetzungsrealität, denn singuläre Systeme sind obsolet – selbst wenn es sich „nur“ um eine Torsteuerung handelt. Ob die tatsächliche Umsetzung der Steuerungsaufgabe durch grafische oder textbasierte

Programmiersprachen, Low- oder No-Code realisiert wird, ist dann unabhängig von dem bildungsnahen Denkansatz des algorithmischen Denkens.

Ob Kompetenzen im Bereich „algorithmisches Denken“ ohne „Programmieren lernen“ bzw. „Programmieren lernen“ ohne „algorithmisches Denken“ entwickelt werden können, soll hier nicht weiter vertieft werden. Angenommen wird hier, dass sich beide Bereiche bedingen und es für die gewerblich-technische Berufsbildung nicht sinnvoll erscheint, diese unabhängig voneinander zu betrachten.

In einer zunehmend technologieorientierten Gesellschaft ist der Alltag vielerorts von sichtbaren und unsichtbaren Algorithmen durchdrungen. Reflektiertes Handeln in Bezug darauf verweist auf eine allgemeine und berufliche Bildung, die lebensbegleitend Kompetenzen entwickelt, die dazu dienen sollen, technische Systeme zu verstehen, zu durchdringen, kritisch zu bewerten und somit letztendlich gestaltend in der Berufs- und Lebenswelt mitzuwirken. Das Thema „Künstliche Intelligenz“ – als nächste große Entwicklungsphase der Digitalisierung – verdeutlicht, dass Reflexionen hier von besonderer Bedeutung sind. Denn auch hier gilt: Vor Algorithmen braucht niemand Angst zu haben, aber vielleicht vor manchen Menschen, die sie erdenken und verwenden.

NACHWORT:

Der hier zitierte FRIEDER NAKE setzte sich in der Informatik seit Jahrzehnten sehr intensiv u. a. mit der Algorithmik, der Semiotik und der Computerisierung als „Maschinisierung von Kopfarbeit“ in Abgrenzung zur Mechanisierung als „Maschinisierung von Handarbeit“ auseinander. Als Leseempfehlung sei hier der folgende Sammelband ausgesuchter Aufsätze genannt, die sich insb. durch ihren essayhaften Schreibstil auszeichnen und zum Nachdenken in dieser Thematik anregen:

DISTELMEYER, J.; EHRMANNTRAUT, S.; MÜLLER, B. (Hrsg.) (2021): Algorithmen & Zeichen: Beiträge von Frieder Nake zur Gegenwart des Computers. Berlin: Kulturverlag Kadmos. Online abrufbar: <https://doi.org/10.25969/mediarep/19385>

Literatur:

KMK (2016): Bildung in der digitalen Welt: Strategie der Kultusministerkonferenz. Hrsg.: Sekretariat der Kultusministerkonferenz, Berlin.

NAKE, F. (2001): Das algorithmische Zeichen. In: W. Bauknecht, W. Brauer, T. Mück (Hrsg.): Tagungsband der GI/OCG Jahrestagung 2001. Bd. II, S. 736 – 742.

Algorithmisches Denken und Programmierung

- Allgemeinbildende Informatik an der Schnittstelle zur gewerblich-technischen Berufsbildung



TIMO HARDEBUSCH



SVEN JACOBS



STEFFEN JASCHKE

Um ein umfassendes Bild der IT-bezogenen Kompetenzen zu zeichnen, die Facharbeiterinnen und Facharbeiter in gewerblich-technischen Berufen benötigen, werden in diesem Beitrag verschiedene Bildungsbereiche betrachtet. Diese umfassen die allgemeinbildende Informatik, IT-spezifische Kompetenzen und Medienbildung. Es werden dazu die spezifischen Kompetenzbereiche des algorithmischen Denkens und der Programmierung der KMK dargestellt.

EINLEITUNG UND MOTIVATION

Die Arbeitswelt befindet sich in einem kontinuierlichen Wandel, der maßgeblich durch die fortschreitende Digitalisierung z. B. durch cyberphysische Systeme, Big Data und die zunehmende Integration von Technologien wie Künstlicher Intelligenz (KI) geprägt ist. Diese Entwicklung bringt tiefgreifende Veränderungen mit sich, die sowohl Chancen als auch Herausforderungen für die Arbeitswelt darstellen. Es entstehen neue Berufsbilder, während traditionelle Berufe einem Wandel unterliegen. Die Digitalisierung hat zu einer Transformation von Arbeitsprozessen, hin zu flexibleren und effizienteren Abläufen, geführt. Dies betrifft nahezu alle Branchen, von der Fertigung über Dienstleistungen bis hin zu Verwaltung und Bildung. Die effektive Nutzung digitaler Werkzeuge ist somit zu einer Kernkompetenz in der modernen Arbeitswelt geworden. KI-Technologien revolutionieren die Art und Weise, wie Entscheidungen getroffen, Daten analysiert und Aufgaben automatisiert werden. Von einfachen Chatbots bis hin zu komplexen, datengesteuerten Analysesystemen verändern KI und Informatik viele Berufsfelder.

Dies hat konkrete Implikationen auf verschiedene Ebenen der beruflichen Bildung (Ausbildung, Weiterbildung, Lehramtsausbildung etc.) sowie für die all-

gemeine Bildung und wird multiperspektivisch diskutiert. Im Kontext dieser Diskussion stellt sich u. a. die Frage, inwieweit das Erwerben von Programmierkenntnissen noch relevant ist, insbesondere jenseits der IT-Berufe. Vielmehr werden neue Kompetenzen hinsichtlich Low-Coding und Prompt-Engineering für die berufliche Bildung als relevant angesehen (vgl. SEUFERT & SPIRGI 2023). Dadurch würden Fachkräfte in die Lage versetzt, Programmcode zu generieren und in eigene Anwendungen zu überführen, ohne eine Programmiersprache und die Grundlagen von Algorithmen und Datenstrukturen zu erlernen. Klar ist aber auch, dass Fachkräfte ohne entsprechende Kompetenzen nicht in der Lage wären, KI-generierten Programmcode zu analysieren und zu bewerten, denn es geht bei der Entwicklung von Software um mehr als die reine Codierung. Vielmehr ist eine Kompetenz zum algorithmischen Denken von Bedeutung, deren Betrachtung ebenso wie die der Programmierkompetenz in den gewerblich-technischen Fachrichtungen hoch volatil ist. Sie richtet sich insbesondere nach den technologischen Möglichkeiten, neuerdings KI-getrieben und in stetigem Wandel.

Letztlich unterliegen die Eingangsvoraussetzungen von Auszubildenden einem ebenso starken wie heterogenen Wandel. In der Praxis zeigt sich eine

große Diskrepanz zwischen formal geforderten und real vorhandenen Medienkompetenzen von Auszubildenden bei Ausbildungsbeginn (DIHK 2019), was wahrscheinlich auch für den Kompetenzbereich algorithmisches Denken übertragen werden kann. Die allgemeinbildende Informatik gewinnt nicht zuletzt deshalb erheblich an Bedeutung, wenngleich sich die curriculare Umsetzung in den Schulen aufgrund des föderalistischen Bildungssystems bundesweit stark unterscheidet (weitestgehend fehlendes Pflichtfach Informatik). Dieser Beitrag erhebt nicht den Anspruch, Antworten auf dringende Fragen von informatischer Bildung zu geben. Vielmehr sollen das Spannungsfeld und der aktuelle Stand der allgemeinbildenden Informatik skizziert werden, was Auswirkungen auf das Handeln und die Erwartungen in der beruflichen Bildung hat. Zunächst ist dazu eine begriffliche Einordnung notwendig.

ALGORITHMISCHES DENKEN, PROGRAMMIERUNG UND COMPUTATIONAL THINKING

In der Diskussion um die notwendigen Kompetenzen heutiger und künftiger Fachkräfte werden Kompetenzen aus verschiedenen Bereichen adressiert und oftmals wenig differenziert. So werden die Begriffe IT-Kompetenz, Informatikkompetenz und Medienkompetenz oftmals synonym verwendet oder in Teilen zusammengefasst z. B. zu Medien- und IT-Kompetenz (vgl. HÄRTEL et al. 2018). Im Grunde geht es um Digital Literacy: „Die Begriffe digitale (Medien-) Kompetenz, Digital Literacy und ICT-Literacy bezeichnen die Fähigkeit, entsprechende Technologie zu verstehen und sie kompetent anwenden zu können“ (SWK 2022, S. 15). Eine detaillierte Betrachtung dieser Digitalkompetenz gestaltet sich schwierig, denn je mehr Digitalisierung in unsere Lebenswelt dringt, desto weniger lässt sich die Bedeutung einer digitalen Kompetenz noch ausmachen (vgl. KERRES 2023). Die Bandbreite von Begrifflichkeiten verdeutlicht die unterschiedlichen disziplinären Sichten auf den Gegenstand. Im Folgenden wird sich auf die Kultusministerkonferenz (KMK) bezogen, denn der KMK-Kompetenzrahmen für eine Bildung in der digitalen Welt „dient allen Bundesländern als Orientierung auch unter Berücksichtigung veränderter Bedingungen des Lehrens und Lernens im Kontext des digitalen Wandels in allen Fächern“ (KMK 2023, S. 2).

Für die Lebens- und Arbeitswelt werden sechs Kompetenzbereiche unterschieden (vgl. KMK 2016):

1. Suchen, Verarbeiten und Aufbewahren,
2. Kommunizieren und Kooperieren,
3. Produzieren und Präsentieren,
4. Schützen und sicher Agieren,

5. Problemlösen und Handeln,

6. Analysieren und Reflektieren.

Algorithmisches Denken und Programmierung ist explizit im Anforderungsbereich „5.5. Algorithmen erkennen und formulieren“ verortet und umfasst das Erkennen und die Formulierung algorithmischer Strukturen in den verwendeten Tools sowie die Planung und Verwendung von strukturierten algorithmischen Sequenzen zur Lösung von Problemen (vgl. KMK 2016, S. 18). Weitere Digitalkompetenzen sind für die berufliche Bildung zentral. Das sichere Agieren in digitalen Umgebungen und der Schutz von persönlichen Daten sowie der Privatsphäre sind allgegenwärtig und auch das Verstehen und Reflektieren von Medien in der digitalen Welt, was den Einsatz von generativer KI einschließt, gewinnt zunehmend an Bedeutung.

Computational Thinking

Im Sinne einer umfassenden Medien-, Digital- und IT-Kompetenz ist eine Betrachtung von algorithmischem Denken und Programmierung begrifflich zu kurz gefasst. In der Informatikdidaktik wird eine übergeordnete Kompetenz, das informatische Denken bzw. Computational Thinking adressiert, welche die eben benannten Konzepte beinhaltet, aber auch die rahmenden Modellierungs- und Analyseprozesse umfasst. Nach GROVER und PEA sind das Logik und logisches Denken, Algorithmik und algorithmisches Denken, Muster und Mustererkennung, Abstraktion und Generalisierung, Evaluation sowie Automatisierung (vgl. 2018). Computational Thinking ist zusammengefasst als der Denkprozess zur Formulierung eines Problems und zur Beschreibung von Lösungswegen zu verstehen, sodass Menschen oder Maschinen diese ausführen können (vgl. WING 2006, S. 33). Es ist also nicht auf die Entwicklung in einer Programmiersprache beschränkt.

Die Diskussion um Kompetenzen im digitalen Zeitalter offenbart eine heterogene Landschaft von Begrifflichkeiten und Konzepten. Die Verschwommenheit der Grenzen zwischen den Disziplinen und die Notwendigkeit einer integrierten Perspektive leiten über zu der Frage, wie das Bildungssystem auf die Anforderungen der Digitalisierung reagieren kann und welche Rolle einem allgemeinbildenden Informatikunterricht dabei zukommt.

DIGITALISIERUNG UND ALLGEMEINBILDENDE INFORMATIK

Parallel zur Entwicklung der Digitalisierung der Lebens- und Arbeitswelt wurde im allgemeinbildenden Kontext diskutiert, welchen Stellenwert die Medien- didaktik und welchen der allgemeinbildende Infor-

matikunterricht zur Ausprägung einer umfassenden IT- und Medienkompetenz einnehmen muss. Nach dem sogenannten Dagstuhl-Dreieck (siehe Abb. 1) sollen Phänomene, Gegenstände und Situationen der digitalen Welt von verschiedenen Seiten betrachtet werden. In der Diskussion von Informatikerinnen und Informatikern, Fachdidaktikerinnen und Fachdidaktikern sowie Mediendidaktikerinnen und Mediendidaktikern hat sich das Modell als geeignet erwiesen, welches die gesellschaftlich-kulturelle, anwendungsbezogene und technologische Perspektive differenziert.

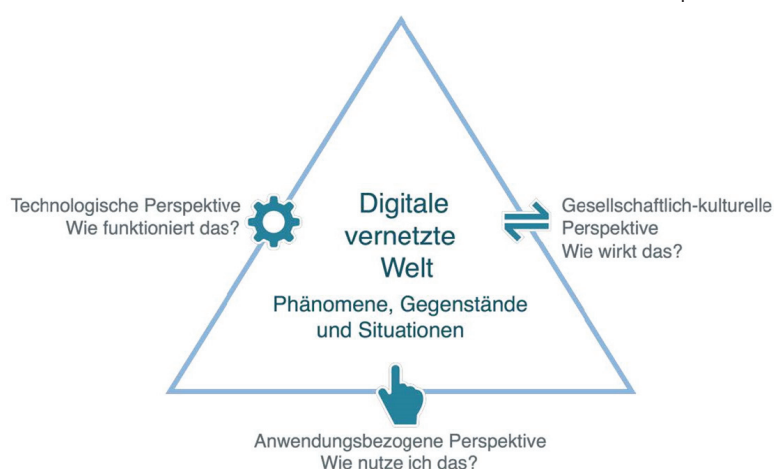


Abb. 1: Dagstuhl-Dreieck als multiperspektivisches Modell zur didaktischen Aufbereitung der digitalen Welt (vgl. BRINDA et al. 2016, S. 3)

Um bei dem Beispiel der künstlichen Intelligenz zu bleiben: Hier existieren verschiedene informatikdidaktische Medien, Mittel und Methoden, um die Funktionsweise von Black-Box-Phänomenen im Zentrum des Dagstuhl-Dreiecks im Unterricht zu behandeln. So wurde SoekiaGPT (vgl. HIELSCHER 2023) speziell für diesen Anwendungsfall in didaktisch reduzierter Form entwickelt. Während die gesellschaftlich-kulturelle sowie die anwendungsbezogene Perspektive durchaus in verschiedenen Fächern betrachtet werden kann und soll, bleibt die technologische Betrachtung der Informatik vorbehalten.

Konsens besteht also auch darin, dass algorithmisches Denken und Programmierung, insbesondere die technologische Perspektive, nicht nur fachfremd oder fächerintegrativ, also außerhalb der Informatik, unterrichtet werden können, sondern ein verpflichtender Informatikunterricht zu forcieren ist, denn die Art der Informatiklehre ist entscheidend für den Kompetenzerwerb der Schülerinnen und Schüler (siehe Abb. 2).

Stand der allgemeinbildenden Informatik in Deutschland

Wenngleich die Notwendigkeit einer informatischen Bildung zur Ausprägung von informatischem Denken und des Verständnisses der technologischen Perspektive geklärt ist, so zeigt sich bundesweit ein heterogenes Bild an verpflichtendem und fakultativem Informatikunterricht.

Im Informatikmonitor (vgl. HELLMIG et al. 2023) wird regelmäßig erhoben, in welchen Bundesländern Informatikunterricht als Pflichtfach, Wahlfach, Wahlpflichtfach oder Abiturfach in welchem Stundenumfang angeboten wird. In Abbildung 3 wird visualisiert, wie viele Schulstunden das Fach Informatik in einer Schullaufbahn der Sekundarstufe I verpflichtend unterrichtet wird. Zum Zeitpunkt der Erhebung im Jahr 2023 umfasste das Spektrum eines durchgängigen Informatikunterrichts in der Spitze die Klassen 5 bis 10 in Mecklenburg-Vorpommern bis hin zu keinem Informatikunterricht in Bremen.

Um eine Vorstellung von den zu erwartenden Kompetenzen der Schülerinnen und Schüler zu erhalten, welche die allgemeinbildenden Schulen verlassen, werden im Folgenden exemplarische Kompetenzen zu algorithmischem Denken und Programmierung des Kernlehrplanes Sekundarstufe I aus NRW aufgeführt. Diese stellen die Mindesterwartungen dar. Darüberhinausgehende Kompetenzen können aber im Übergang in die berufliche Bildung je nach Bundesland nicht zwingend erwartet werden.

Konkretisierte Kompetenzerwartungen (Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen 2021, S. 18):

„Die Schülerinnen und Schüler

- überführen Handlungsvorschriften in einen Programmablaufplan oder ein Struktogramm
- implementieren Algorithmen unter Berücksichtigung des Prinzips der Modularisierung
- bewerten einen als Quelltext, Programmablaufplan oder Struktogramm dargestellten Algorithmus hinsichtlich seiner Funktionalität
- implementieren Algorithmen in einer visuellen Programmiersprache“

Einerseits ist zu erkennen, dass die Kompetenzerwartungen im Bereich Algorithmik und Programmierung kognitiv herausfordernd sind. Andererseits stellt sich die Frage, welche Fertigkeiten beim Verlassen der allgemeinbildenden Schulen nach fünf und mehr

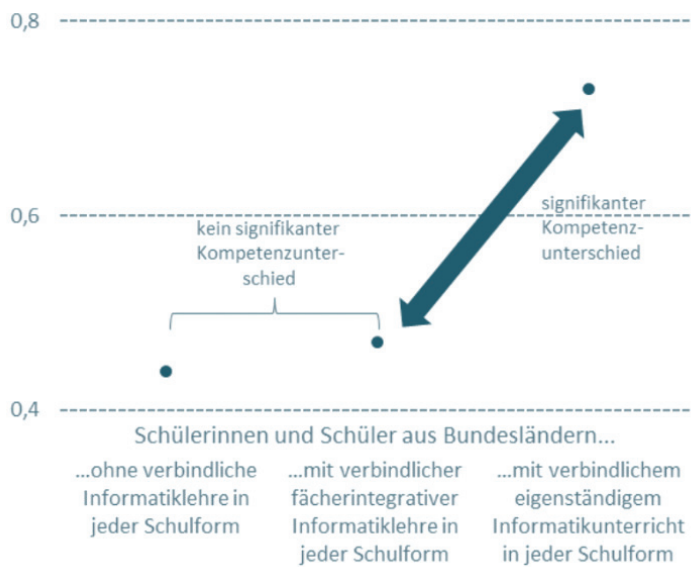


Abb. 2: Unterschiede im Bereich digitaler und informatikbezogener Kompetenzen von Schülerinnen und Schülern der Klasse 9 in Abhängigkeit von der vorhandenen Informatiklehre (vgl. SUESSENBACH et al. 2022, S. 5)

Jahren noch anwendbar sind. Eine weitere Verortung zum Ende der Sekundarstufe I scheint daher dringend angezeigt, was in einigen Bundesländern (siehe Abb. 3) der Fall ist. Letztlich stellt sich dann die Frage, welche Themen und Kompetenzen Teil der berufsschulischen Ausbildung sein müssen oder bereits in der Allgemeinbildung verortet sind bzw. sein sollten.

KOMPETENZANFORDERUNGEN IN METALL- UND ELEKTROBERUFEN

In der Diskussion um die notwendigen Kompetenzen heutiger und künftiger Fachkräfte werden Kompetenzen aus verschiedenen Bereichen adressiert, aber oftmals wenig differenziert. Neben den oben benannten allgemeinen Medienkompetenzen ist die notwendige Ausprägung abhängig von den zu Grunde liegenden realen Arbeitsprozessen (Arbeitsprozessorientierung: (a) die Gegenstände der (Berufs-) Arbeit, (b) die Methoden, Werkzeuge und Organisationsformen der (Berufs-)Arbeit und (c) die gesellschaftlichen, betrieblichen und subjektiven Anforderungen an die (Berufs-)Arbeit und Technik (vgl. SPÖTTL & BLINGS 2011, S. 38). Während Kompetenzen zum algorithmischen Denken und der Programmierung den IT-Berufen immanent sind, beschränkt sich die explizite Benennung in Kernlehrplänen der weiteren Fachrichtungen auf einzelne Aspekte oder aber ist lediglich als Teil der Standardberufsbildposition digitalisierte Arbeitswelt zu fassen. Die Standardberufsbildpositionen umfassen mindestens zu

erreichende Kompetenzformulierungen für alle Ausbildungsberufe ab August 2021.

Nach der Neuordnung der IT-Berufe 2020 existieren zwei neue Fachrichtungen im Beruf „Fachinformatiker/in“, und zwar „Daten- und Prozessanalyse“ sowie „Digitale Vernetzung“. Erstere ist an der Fachrichtung Anwendungsentwicklung angelehnt, fokussiert aber auf die Analyse und Bereitstellung von Daten und Datenquellen sowie deren Nutzung zur Optimierung von Arbeits- und Geschäftsprozessen. Letztere ist an die Fachrichtung Systemintegration angelehnt und fokussiert auf die Analyse und Planung insbesondere vernetzter Systeme und deren Betriebes hinsichtlich Systemverfügbarkeit.

Auch wenn algorithmisches Denken und Programmierung in besonderem Maße Teil der Fachrichtung Anwendungsentwicklung sind, finden sich in den fachrichtungsübergreifenden Lernfeldern 1 bis 9 folgende Kompetenzen für alle IT-Berufe (vgl. KMK 2019, S. 16, 19):

- „Die Schülerinnen und Schüler implementieren die Anpassung der Anwendung, auch im Team und erstellen eine Softwaredokumentation,
- Sie testen die Funktion der Anwendung und beurteilen deren Eignung zur Bewältigung der gestellten Anforderungen.
- Sie evaluieren den Prozess der Softwareentwicklung. [...]
- Die Schülerinnen und Schüler implementieren arbeitsteilig, auch ortsunabhängig, ihr Konzept mit vorhandenen sowie dazu passenden Entwicklungswerkzeugen und Produkten.“ (vgl. KMK 2019, S. 16, 19)

Als Entsprechungen in den Ausbildungsrahmenplänenentwürfen werden u. a. folgende Berufsbildpositionen benannt (KMK 2019, S. 41, 43):

- „Programmiersprachen, insbesondere prozedurale und objektorientierte Programmiersprachen, unterscheiden
- Algorithmen formulieren und Anwendungen in einer Programmiersprache erstellen
- Programmspezifikationen festlegen, Datenmodelle und Strukturen aus fachlichen Anforderungen ableiten sowie Schnittstellen festlegen
- Programmiersprachen auswählen und unterschiedliche Programmiersprachen anwenden“.

In den gewerblich-technischen Fachrichtungen abseits der IT-Berufe zeigt sich hinsichtlich der im Kontext von Digitalisierung und algorithmischen Denken

zu erwerbenden Kompetenzen ein heterogenes Bild. Mit der Neuformulierung der Standardberufsbildpositionen durch das Bundesinstitut für Berufsbildung sollen einheitliche und verbindliche Grundlagen für alle Ausbildungsberufe geschaffen werden, um die Teilhabe innerhalb einer digitalisierten Arbeitswelt sicherzustellen (vgl. BIBB 2021). Dabei liegt der Fokus auf einer integrativen Vermittlung allgemeiner, vorwiegend anwendungsbezogener Kompetenzen im Umgang mit digitalen Medien wie Datenschutz, Sicherheit und Informationsrecherche:

- „Mit eigenen und betriebsbezogenen Daten sowie mit Daten Dritter umgehen und dabei die Vorschriften zum Datenschutz und zur Datensicherheit einhalten, [...]“
- Risiken bei der Nutzung von digitalen Medien und informationstechnischen Systemen einschätzen und bei deren Nutzung betriebliche Regelungen einhalten, [...]“
- Informationen in digitalen Netzen recherchieren und aus digitalen Netzen beschaffen sowie Informationen, auch fremde, prüfen, bewerten und auswählen, [...]“
- Lern- und Arbeitstechniken sowie Methoden des selbstgesteuerten Lernens anwenden, digitale Lernmedien nutzen und Erfordernisse des lebensbegleitenden Lernens erkennen und ableiten.“ (BIBB 2021, S. 13 f.)

Ein expliziter Verweis auf algorithmisches Denken und Programmierung fehlt – im Gegensatz zum oben genannten KMK-Strategiepapier – jedoch. Da durch

INFORMATIKUNTERRICHT IN DER SEKUNDARSTUFE I (Stand: Oktober 2023)



Abb. 3: Übersicht über die Bundesländer mit Informatikunterricht in der Sekundarstufe I (HELLMIG et al., 2023, S.4)

die Standardberufsbildpositionen lediglich Mindestanforderungen festgesetzt werden, verschiebt sich die Frage nach der technologischen Perspektive des Dagstuhl-Dreiecks weiterhin in die fachbereichsspezifischen Kompetenzerwartungen der jeweiligen Rahmenlehrpläne mit den zu antizipierenden Folgen: Zum einen unterscheiden sich Umfang und Tiefe der formulierten Erwartungen deutlich zwischen den Fachrichtungen, zum anderen konzentriert sich die Auswahl der Kompetenzen regelmäßig auf für die jeweilige Domäne besonders relevante Inhalte (bspw. Netzwerksysteme für

Mechatroniker/-innen) sowie Spezialwissen oder ist in mögliche Zusatzqualifikationen (bspw. „Programmierung“) ausgelagert. Zuletzt beschränkt sich die verpflichtende Wirkung der modernisierten Standardberufsbildpositionen auf ab August 2021 in Kraft tretende Rahmenlehrpläne und Ausbildungsordnungen.

FAZIT

Es bleibt demnach auf absehbare Zeit eine Fragmentierung bestehen, die eine Herausforderung für den Erwerb kohärenter und umfassender Kompetenzen im Bereich des algorithmischen Denkens über alle Fachrichtungen hinweg darstellt. Die Einführung eines allgemeinbildenden und verpflichtenden Informatikunterrichts ist unter diesen Voraussetzungen kein Luxus, sondern eine Notwendigkeit, um sicherzustellen, dass jede Person – unabhängig von ihrem zukünftigen Berufsweg – die Werkzeuge und das Ver-

ständnis besitzt, um erfolgreich an einer digitalisierten Arbeits- und Lebenswelt teilzuhaben.

Angeichts der Tatsache, dass Automatisierung und KI fortschreitend auch in Nicht-IT-Berufen Einzug halten, wird es zunehmend wichtiger, dass jede und jeder über ein Grundverständnis informatischer Konzepte verfügt. Dies schafft eine gemeinsame Basis, die es ermöglicht, nicht nur mit Veränderungen Schritt zu halten, sondern auch aktiv an der Gestaltung einer digitalen Zukunft teilzunehmen.

Literaturverzeichnis

- BRINDA, T.; DIETHELM, I.; GEMULLA, R.; ROMEIKE, R.; SCHÖNING, J.; SCHULTE, C. (2016): Dagstuhl-Erklärung: Bildung in der digital vernetzten Welt. Gesellschaft für Informatik, abrufbar unter: <https://dagstuhl.gi.de/dagstuhl-erklaerung> (zuletzt abgerufen am 29.04.2024)
- BIBB (BUNDESINSTITUT FÜR BERUFSBILDUNG) (Hrsg.) (2021): Vier sind die Zukunft: Digitalisierung. Nachhaltigkeit. Recht. Sicherheit. Die modernisierten Standardberufsbildpositionen anerkannter Ausbildungsberufe. Verlag Barbara Budrich, abrufbar unter: <https://www.bibb.de/dienst/publikationen/de/17281> (zuletzt abgerufen am 29.04.2024)
- DIHK (DEUTSCHER INDUSTRIE- UND HANDELSKAMMERTAG) (Hrsg.) (2019): Ausbildung 2019. Ergebnisse einer DIHK-Online-Unternehmensbefragung, abrufbar unter: <https://www.dihk.de/de/themen-und-positionen/fachkraefte/aus-und-weiterbildung/ausbildung/dihk-ausbildungsumfrage-wissen-was-zu-tun-ist-11430> (zuletzt abgerufen am 03.03.2024)
- GROVER, S.; PEA, R. (2018): Computational Thinking: A Competency Whose Time Has Come. In: S. Sentance, E. Barendsen, C. Schulte (Eds.), Computer Science Education: Perspectives on Teaching and Learning in School, 19-38
- HÄRTEL, M.; AVERBECK, I.; BRÜGGEMANN, M.; BREITER, A.; HOWE, F.; SANDER, M. (2018): Medien- und IT-Kompetenz als Eingangsvoraussetzung für die berufliche Ausbildung – Synopse. Verlag Barbara Budrich, abrufbar unter: <https://www.bibb.de/dienst/publikationen/de/9223> (zuletzt abgerufen am 29.04.2024)
- HELLMIG, L.; SCHIECKOFF, B.; SCHWARZ, R.; SÜSSENBACH, F. (2023): Informatik-Monitor 2023/24. Zur Situation des Informatikunterrichts in Deutschland. Oktober 2023. Gesellschaft für Informatik, abrufbar unter: https://informatik-monitor.de/fileadmin/GI/Projekte/Informatik-Monitor/Informatik-Monitor_2023-24/PDF-Versionen/Informatik-Monitor_2023-24_Final.pdf (zuletzt abgerufen am 29.04.2024)
- HIELSCHER, M. (2023): SoekiaGPT - ein didaktisches Sprachmodell. In: Informatische Bildung in Schulen, 1(1). Gesellschaft für Informatik, abrufbar unter: <https://doi.org/10.18420/IBIS-01-01-04> (zuletzt abgerufen am 29.04.2024)
- KERRES, M. (2023): Bildung in der digitalen Welt: (Wie) Kann digitale Kompetenz vermittelt werden? In: N. McElvany, R. Lorenz, M. Becker, H. Gaspard, F. Lauer-mann (Hrsg.), Lernen in und für die Schule im Zeitalter der Digitalität – Chancen und Herausforderungen für Lehrkräfte, Unterricht und Lernende. Waxmann.
- KMK (KULTUSMINISTERKONFERENZ) (2016): Bildung in der digitalen Welt: Strategie der Kultusministerkonferenz: Beschluss der Kultusministerkonferenz vom 08.12.2016 in der Fassung vom 07.12.2017, abrufbar unter: https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2018/Digitalstrategie_2017_mit_Weiterbildung.pdf (zuletzt abgerufen am 29.04.2024)
- KMK (KULTUSMINISTERKONFERENZ) (2019): Rahmenlehrplan für die Ausbildungsberufe Fachinformatiker und Fachinformatikerin, IT-System-Elektroniker und IT-System-Elektronikerin (Beschluss der Kultusministerkonferenz vom 13.12.2019), abrufbar unter: https://www.kmk.org/fileadmin/Dateien/pdf/Bildung/BeruflicheBildung/rfp/Fachinformatiker_19-12-13_EL.pdf (zuletzt abgerufen am 29.04.2024)
- KMK (KULTUSMINISTERKONFERENZ) (2023): Jahresbericht der Kultusministerkonferenz zur Bildung in der digitalen Welt (Beschluss der KMK vom 07./08.12.2023). Berichtszeitraum: 01.08.2022 bis 31.07.2023, abrufbar unter: https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2023/2023_12_07-Jahresbericht-Bildung-in-der-digitalen-Welt_2022-2023.pdf (zuletzt abgerufen am 29.04.2024)
- MINISTERIUM FÜR SCHULE UND WEITERBILDUNG DES LANDES NORDRHEIN-WESTFALEN (2021): Kernlehrplan für die Sekundarstufe I - Klasse 5 und 6 in Nordrhein-Westfalen. Informatik, abrufbar unter: https://www.schulentwicklung.nrw.de/lehrplaene/lehrplan/260/si_kl5u6_if_klp_2021_07_01.pdf (zuletzt abgerufen am 29.04.2024)
- SEUFERT, S.; SPIRGI, L. (2023). Programmieren im Zeitalter der generativen KI: eine transversale Kompetenz in der Berufsbildung. In: A. Barabasch, S. Fischer (Hrsg.): bwp@ Spezial 20: Die Förderung von transversalen Kompetenzen in der Berufsbildung, 1-26, abrufbar unter: https://www.bwpat.de/spezial20/seufert_spirgi_spezial20.pdf (zuletzt abgerufen am 29.04.2024)
- SPÖTTL, G.; BLINGS, J. (2011). Ein Baustein für ein transnationales Berufsbildungskonzept. Frankfurt, Main. Lang.
- SUESSENBACH, F.; SCHRÖDER, E.; WINDE, M. (2022): Informatik für alle! Diskussionspapier, abrufbar unter: <https://www.stifterverband.org/medien/informatik-fuer-alle> (zuletzt abgerufen am 29.04.2024)
- SWK (STÄNDIGE WISSENSCHAFTLICHE KOMMISSION DER KULTUSMINISTERKONFERENZ) (Hrsg.) (2022): Digitalisierung im Bildungssystem: Handlungsempfehlungen von der Kita bis zur Hochschule. Gutachten der Ständigen Wissenschaftlichen Kommission der Kultusministerkonferenz (SWK), abrufbar unter: <https://doi.org/10.25656/01:25273> (zuletzt abgerufen am 29.04.2024)
- WING, J. M. (2006): Computational Thinking. Communications of the ACM 49(3), 33–35, abrufbar unter: <https://doi.org/10.1145/1118178.1118215> (zuletzt abgerufen am 29.04.2024)

Programmierenlernen unter Berücksichtigung der Cognitive Load Theory (CLT)



PEER STECHERT

Ziel dieses Beitrages ist es, vor dem Hintergrund der Cognitive Load Theory (CLT) Schwierigkeiten aufzuzeigen, vor denen gerade Auszubildende ohne nennenswerte Programmierkenntnisse zu Beginn ihrer Ausbildung stehen. Der Artikel stellt Strategien vor, die beim Programmierenlernen helfen können, u. a. „Program-Tracing“. Außerdem wird vor dem Hintergrund der CLT die Lernförderlichkeit von Programmierkonventionen betrachtet. Abschließend wird ein Ausblick auf Veränderungen durch KI-Werkzeuge vorgenommen.

EINLEITUNG

Computational Thinking gilt als eine Kernkompetenz des 21. Jahrhunderts und zielt darauf ab, Menschen in die Lage zu versetzen, Methoden der Informatik für die Lösung von Problemen in einer Vielzahl an Domänen einzusetzen. Das Erlernen einer Programmiersprache stellt einen wichtigen Zugang zu Computational Thinking im Rahmen einer gewerblich-technischen Berufsausbildung dar. Gleichzeitig durchdringt Informatik – und insbesondere das Programmieren – mehr und mehr die gewerblich-technischen Ausbildungsberufe und ihre Fachrichtungen Elektrotechnik, Informationstechnik, Metalltechnik und Fahrzeugtechnik (vgl. KMK 2019, S. 103). Der Prozess des Programmierenlernens kann jedoch herausfordernd sein.

ANFORDERUNGEN AN DIE PROGRAMMIERUNG IN UNTERSCHIEDLICHEN BERUFLICHEN FACHRICHTUNGEN

Die vier beruflichen Fachrichtungen – Elektrotechnik, Informationstechnik, Metalltechnik und Fahrzeugtechnik – haben unterschiedliche Schwerpunkte, die auch die Anforderungen an das Programmieren beeinflussen. JENEWEIN (2020) hält mit Blick auf die ländergemeinsamen Anforderungen der KMK (2019) für die Elektrotechnik fest: „In der Fachrichtung Elektrotechnik [...] besteht eine Orientierung der inhaltlichen Anforderungen auf Inhaltsbereiche, die sich vorwiegend an elektrotechnischen Anwendungsfeldern orientieren“ (JENEWEIN 2020, S. 130). Dem stellt er das Fachprofil Informationstechnik/Informatik gegenüber: „Stellt man diese beiden Fach-

profile gegenüber, so ist rasch zu erkennen, dass im Bereich der Informationstechnik inhaltliche und semantische Überschneidungen existieren, gleichzeitig jedoch die Fachrichtung Informationstechnik/Informatik sich deutlich aus der Tradition der klassischen Ingenieurwissenschaft Elektrotechnik abhebt und sich fast vollständig an der Inhaltsstruktur des Unterrichtsfachs Informatik orientiert“ (JENEWEIN 2020, S. 131). Die Fachrichtung Informationstechnik/Informatik beinhaltet neben allgemeinen Grundlagen wie Mathematik und Elektrotechnik auch Kernbereiche der Informatik, darunter Algorithmen, Datenstrukturen, Datenmodellierung, Programmierung und Rechnerstrukturen. Zusätzlich werden spezialisierte Themen wie IT-Systementwicklung, -betrieb, Sicherheit und betriebswirtschaftliche Anwendungen behandelt (vgl. KMK 2019).

Zu den Studieninhalten der Metalltechnik mit Programmier-Bezug gehören Elektrotechnik, Informationstechnik sowie Mess-, Steuer- und Regelungstechnik (vgl. KMK 2019, S. 79). Für die Fahrzeugtechnik werden berufsfeldrelevante Mess-, Steuer- und Regelungstechnik sowie fahrzeugspezifische Elektrotechnik, Elektronik und Mechatronik genannt (vgl. KMK 2019, S. 95). In den drei Fachrichtungen Elektrotechnik, Metalltechnik und Fahrzeugtechnik ist Programmierung somit integraler Bestandteil von eingebetteten Systemen, Steuerungssystemen und Mikrocontroller-Anwendungen. Typischerweise kommen maschinennahe Sprachen wie C oder die nach IEC 61131 für Speicher programmierbare Steuerungen zum Einsatz, wobei grundlegende Programmierkonzepte auch hier von Bedeutung sind. Die

Integration von Sensor- und Aktordaten erfordert spezifisches Know-how.

COGNITIVE LOAD THEORY IM KONTEXT DES PROGRAMMIERENS

Gedächtnismodell

Die kognitive Belastungstheorie (CLT), die 1988 von SWELLER (vgl. SWELLER, MERRIËNBOER & PAAS 2019) eingeführt wurde, bietet einen Rahmen zur Untersuchung von Lernprozessen. Diese Theorie fokussiert darauf, wie Lernprozesse an das menschliche Gedächtnissystem und informationsverarbeitende Prozesse angepasst werden können.

CLT basiert auf drei Grundannahmen:

1. Lernen erfordert kognitive Anstrengung.
2. Wissenserwerb impliziert das Konstruieren, Erweitern oder Umstrukturieren von Schemata im Langzeitgedächtnis.
3. Lernen benötigt Kapazitäten im Arbeitsgedächtnis.

Das Gedächtnismodell (siehe Abb. 1) unterscheidet zwischen dem Langzeitgedächtnis, wo Informationen als Schemata gespeichert sind, und dem Arbeitsgedächtnis, das eine begrenzte Kapazität zur gleichzeitigen Verarbeitung von Informationen hat. Im Allgemeinen wird davon ausgegangen, dass visuell-räumliche und sprachliche Informationen sowohl über getrennte Kanäle aufgenommen als auch in unterschiedlichen Komponenten verarbeitet werden. Die phonologische Schleife verarbeitet sprachliche Informationen und kann sich Worte nur für einen begrenzten Zeitraum (ca. zwei Sekunden) merken. Das

räumlich-visuelle Notizbuch ist ein für das vorübergehende Speichern von räumlichen und visuellen Informationen verantwortliches System mit begrenzter Kapazität von ca. fünf Elementen.

Das Grundproblem beim Programmieren ist die hohe Anzahl interagierender Elemente. Viele Herausforderungen des Programmierlernens lassen sich entsprechend durch die CLT erklären: Die Notwendigkeit, verschiedene Aspekte wie Datentypen, Variablendeklaration und bedingte Anweisungen gleichzeitig zu verstehen, übersteigt oft die Kapazität des Arbeitsgedächtnisses. In Abbildung 2 ist beispielsweise ein Programm zu sehen, das je nach Belegung der Variablen x und y den Wert der Variable z ändert. Bereits um diesen kurzen Programmode nachvollziehen zu können, ist es notwendig, die acht rechts aufgeführten Begriffe und Konzepte zu verstehen, womit für ungeübte Lernende die Kapazitätsgrenze des Arbeitsgedächtnisses erreicht bzw. überschritten wird. Geübtere Lernende nehmen Ausdrücke wie „ $\text{int } x = 10;$ “ hingegen als ein zusammengehöriges Konstrukt (Chunk) wahr, in dem die drei Konzepte Datentyp, Variable und Initialisierung zusammenspielen und nur einen Platz im Arbeitsgedächtnis belegen.

Unterschiedliche Arten kognitiver Last und Effekte

Insgesamt ist die CLT sehr gut empirisch untersucht und mit ihr lassen sich eine Reihe von Effekten erklären, die Einfluss auf das Lernen haben. Diese Effekte wiederum lassen sich lose den folgenden drei Arten der kognitiven Last zuordnen (vgl. SWELLER, MERRIËNBOER & PAAS 2019):

- Themenbezogene Last (auch intrinsische Last; intrinsic load): Schwierige Themen sollen durch

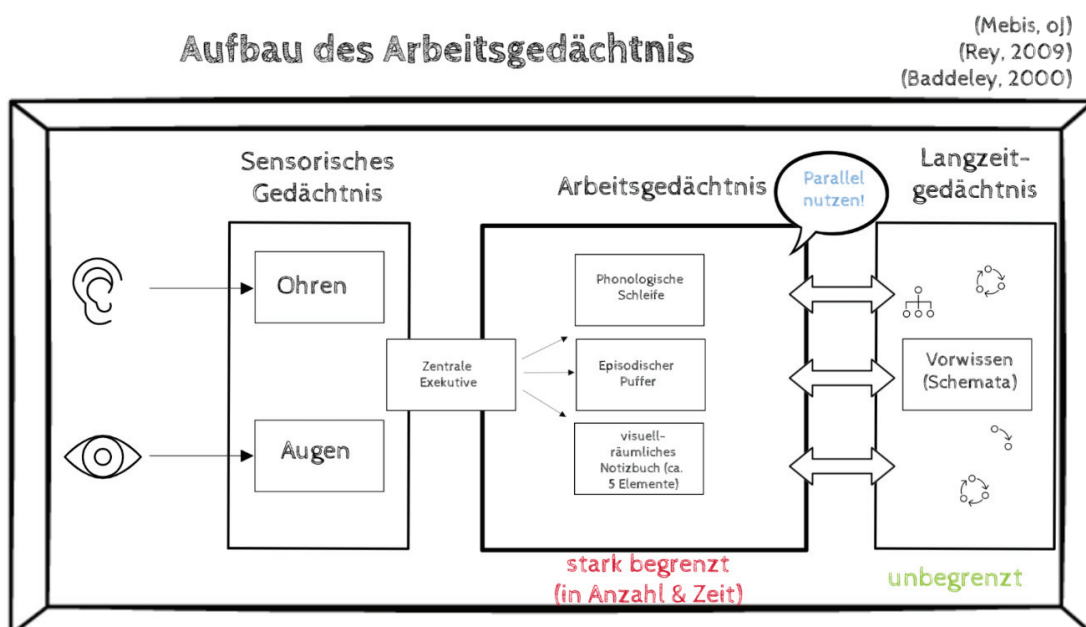


Abb. 1: Gedächtnismodell (STECHE 2022)

Programmieren und kognitive Last

Programmieren erzeugt eine hohe kognitive Last, die das gleichzeitige Lernen wichtiger Konzepte verhindert. (Mason & Cooper, 2012)

Einfache Programme motivieren wenig, komplexe Programme sind unerreichbar.

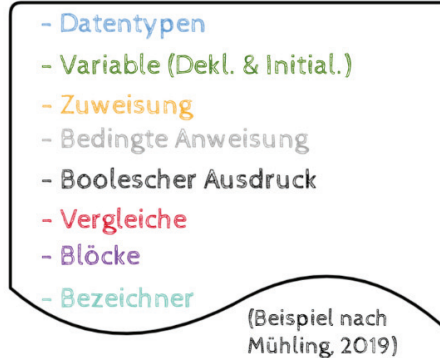
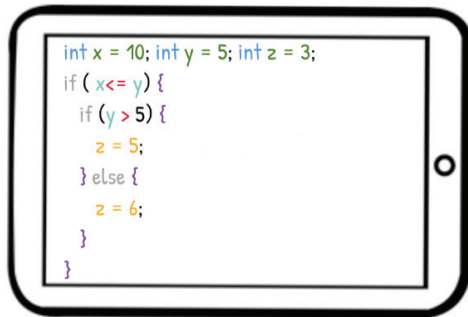


Abb. 2: Bereits kleine Codeschnipsel erfordern vielfältiges Vorwissen (STECHEIT 2022)

gezielte didaktische Reduktion und Gestaltung zugänglich gemacht werden.

- Lernbezogene Last (auch lernrelevante Last; germane load): Auch der Lernprozess kann unterschiedlich hohe kognitive Belastung verursachen. Zur Reduktion muss der Fokus der Lernaktivitäten gesteuert werden. Kognitive Aktivierung ist hier das Schlagwort.
- Materialbezogene Last (auch extrinsische bzw. sachfremde Last; extraneous load): Es soll möglichst wenig Ablenkung durch das Lernmaterial erzeugt werden, wozu beim Programmieren auch die Wahl der Programmiersprache und der Entwicklungsumgebung zählt.

Neuere Forschungsergebnisse deuten darauf hin, dass die lernbezogene Last (germane load) die gesamte kognitive Last nicht erhöht, sondern zwischen den beiden anderen Lasten umverteilt:

„Currently, we assume that rather than contributing to the total load, germane cognitive load redistributes working memory resources from extraneous activities to activities directly relevant to learning by dealing with information intrinsic to the learning task“ (SWELLER, MERRIËNBOER & PAAS 2019, S. 264).

Die CLT hilft, verschiedene Effekte zu erklären, die das Lernen beeinflussen und von denen einige im Folgenden mit Programmierbezug vorgestellt werden (siehe Abb. 3, nächste Seite).

Der Split-Attention-Effect im Lernkontext: Der Split-Attention-Effect beschreibt, wie Lernende Informationen aus verschiedenen Quellen zusammenführen müssen. Insbesondere mit Blick auf das räumlich-visuelle Notizbuch als Teil des Arbeitsgedächtnisses (siehe Abb. 1) führt dies zu einer erhöhten kognitiven

Belastung. Um diese zu reduzieren, sollten Lernmaterialien so gestaltet werden, dass Diagramme und visuelle Darstellungen unmittelbar mit ihren Erklärungen verknüpft sind. Ein Beispiel hierfür ist die (auch räumlich nahe, gleichzeitig sichtbare) Visualisierung von Quellcode durch Klassendiagramme.

Redundancy-Effect: Der Redundancy-Effect betont die kognitive Belastung durch überflüssige oder sich wiederholende Informationen. In Programmen, die eine Lehrkraft vorgibt, ist es daher vorteilhaft, auf sich wiederholende Quelltext-Anteile zu verzichten und stattdessen auf Wiederverwendung durch Auslagerung in Funktionen oder mittels Objektorientierung zu setzen.

Transient-Information-Effect: Dieser Effekt hebt hervor, dass nicht immer alle relevanten Informationen im Gedächtnis behalten werden können. Datenblätter und technische Referenzen, wie die Elemente von UML (Unified Modeling Language)-Diagrammen oder SQL (Structured Query Language)-Schlüsselwörter, können als effektive Hilfsmittel dienen, um die kognitive Last zu reduzieren. Concept Maps zur Reflexion über den Lernfortschritt im Klassenverband sind ebenfalls nützlich, um Zusammenhänge zwischen verschiedenen Konzepten zu verdeutlichen.

Worked-Examples-Effect: Bevor Schülerinnen und Schüler eigenständig größere Aufgaben oder Probleme lösen, sollten ausgearbeitete Lösungsbeispiele (Worked-Examples) sowohl Hinweise zum Lösungsweg als auch zur Darstellungsform der erwarteten Lösung geben. Worked-Examples können beispielsweise Best-Practices aufzeigen wie die Nutzung geeigneter Datenstrukturen oder ein geeignetes Maß an Dokumentation und Kommentaren.

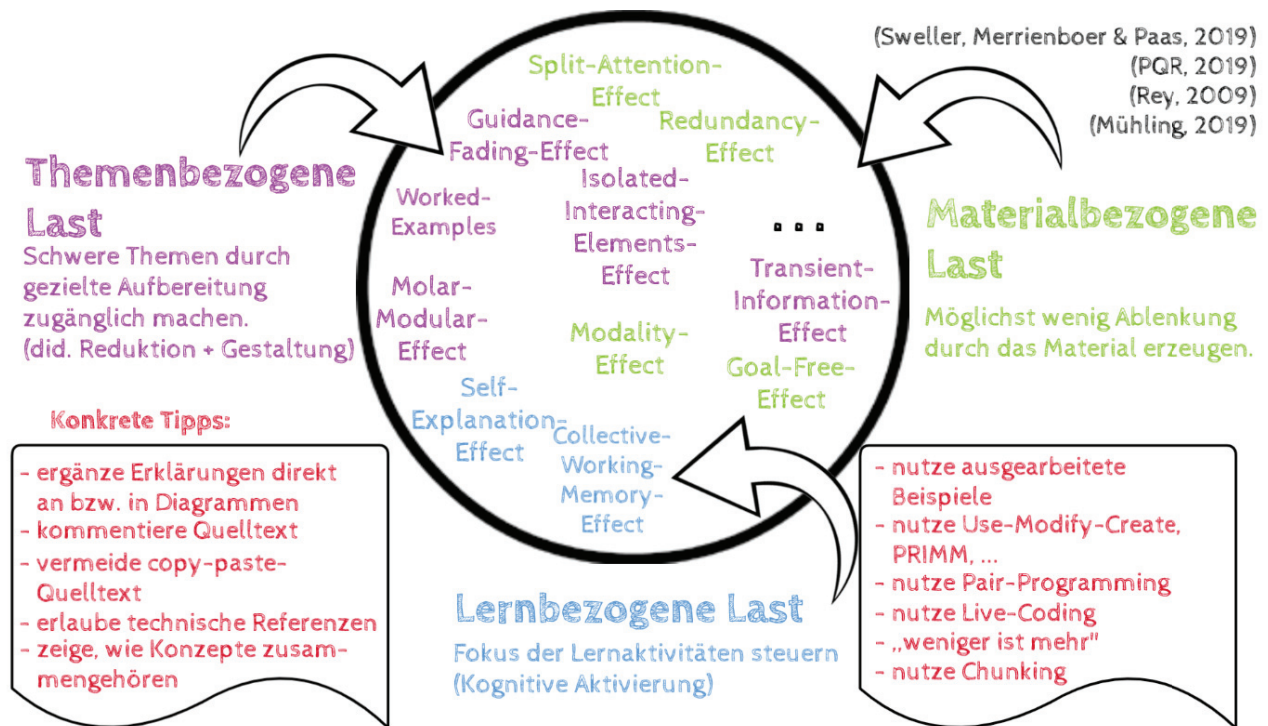


Abb. 3: Themenbezogene, materialbezogene und lernbezogene Last mit korrespondierenden Effekten (STECHE 2022)

Guidance-Fading-Effect: Viele Unterrichtsmethoden in der Informatik basieren auf dem sogenannten Guidance-Fading-Effect. Das heißt, die Lehrkraft greift am Anfang stark strukturierend in den Lernprozess ein und zieht sich bei fortgeschrittenen Lernenden immer weiter aus dem Lernprozess zurück. Unterrichtsmethoden wie Use-Modify-Create oder dessen Variante PRIMM (Predict-Run-Investigate-Modify-Make) unterstützen diesen Ansatz durch schrittweise zunehmende Eigenständigkeit. Bei beiden Methoden bekommen die Lernenden ein lauffähiges Programmgerüst von der Lehrkraft, das sie nach und nach analysieren, modifizieren und letztendlich zu etwas Eigenem umbauen sollen. Diese Ansätze nehmen so die Last, ein Programm „from scratch“ neu zu erstellen, lassen aber gleichzeitig viele Freiheiten bei der Modifizierung und dem Neugestalten, und gelten deshalb auch als „selbstdifferenzierend“.

Collective-Working-Memory-Effect im kooperativen Lernen: Dieser Effekt zeigt auf, dass kognitive Belastung innerhalb einer Gruppe verteilt werden kann. Methoden wie Pair-Programming mit den (regelmäßig wechselnden) Rollen Driver und Navigator im Informatikunterricht können hierbei besonders effektiv sein.

Modality- und Self-Explanation-Effect: Das Arbeitsgedächtnis hat eine begrenzte Kapazität an Elementen, die gleichzeitig verarbeitet werden können. Der Modality-Effect betont die Bedeutung von verbalen und visuellen Informationskanälen, die jeweils in der Anzahl der verarbeitbaren Elemente beschränkt sind,

aber deren parallele Nutzung die Gesamtanzahl erhöht. Die Methode „Live Coding“ durch die Lehrkraft kann beide Kanäle effektiv nutzen. Die Lernenden sehen der Lehrkraft beim Programmieren zu und hören gleichzeitig ihre Erläuterungen („lautes Denken“). In eine ähnliche Richtung geht auch der Self-Explanation-Effect. Hierbei sollen Schülerinnen und Schüler bei der Lösung von Problemen selbst laut denken.

Molar-Modular-Effect und Isolated-Interacting-Element-Effect: Diese Effekte unterstreichen die Bedeutung von Modularität und der sequenziellen Integration von Lernelementen. Es ist effektiver, sich auf einzelne, isolierte Elemente zu konzentrieren und diese später in einer Concept Map zu vernetzen, als sofort das Gesamtsystem in den Blick zu nehmen.

Chunking als Unterstützung für die Schemabildung: Abschließend wird das Chunking als eine Methode zur Reduktion der kognitiven Last und Förderung der Schemabildung hervorgehoben. Gemeint ist damit das Zusammenführen von isolierten Wissenselementen zu einem zusammengehörenden Element. Beispielsweise nutzen Software-Entwurfsmuster und Architekturmodelle diesen Effekt, um komplexe Zusammenhänge zusammenzufassen. Aber auch die Komponenten bzw. Blöcke von visuellen „Low Code“-Programmiersprachen kapseln viele Einzelelemente und reduzieren damit die kognitive Last.

Fazit

Die effektive Gestaltung von Lernmaterialien und -methoden unter Berücksichtigung der kognitiven Belastungstheorie kann das Programmierenlernen im Bereich der Informatik deutlich verbessern. Klar ist aber auch, dass die in der beruflichen Bildung eingesetzten komplexen professionellen Entwicklungsumgebungen direkt die kognitive Last erhöhen. Somit beeinflussen sowohl die Wahl der Werkzeuge als auch die Wahl der (Unterrichts-) Methoden die kognitive Last. Hervorzuheben sind Live-Coding und Worked-Examples aber auch vor dem Hintergrund der Cognitive-Apprenticeship-Theorie: Beides sind Methoden, die im Sinne von Meister-Lehrlings-Verhältnissen kognitive Prozesse der Expertinnen und Experten für die Lernenden sichtbar machen.

Kritische Bewertung der CLT

Obwohl die CLT wertvolle Einblicke in den Lernprozess bietet, gibt es auch Kritikpunkte, wie die Vernachlässigung der Rolle von Motivation und Emotion im Lernprozess: Einfache Programme motivieren wenig, komplexe Programme sind unerreichbar (siehe Abb. 2). Zudem steht die Theorie insofern mit Ansätzen der Lernfelddidaktik in Konflikt, als sie nahelegt, vollständige berufliche Handlungssituationen erst spät im Lernprozess zu thematisieren.

PROGRAMME VERSTEHEN DURCH TRACING UND UNTER NUTZUNG VON PROGRAMMIERKONVENTIONEN

Fachdidaktische Prinzipien für den Informatikunterricht und das Programmierenlernen

In England hat das National Centre for Computing Education in Zusammenarbeit mit der Raspberry Pi Foundation zwölf fachdidaktische Prinzipien für effektiven Informatikunterricht definiert. Diese sind (ncce.io/pedagogy; vgl. STECHERT 2023):

1. Lead with concepts,
2. Unplug, unpack, repack,
3. Create projects,
4. Challenge misconceptions,
5. Structure lessons,
6. Work together,
7. Model everything,
8. Add variety,
9. Make concrete,
10. Read and explore code first,
11. Get hands-on,
12. Foster program comprehension.

Mit Blick auf das Programmierenlernen in den gewerblich-technischen Berufen sind sicherlich sowohl die Durchführung von Softwareprojekten (3), Pair-Programming (6), die Arbeit mit Microcontrollern und Making (11) sowie die Strukturierung von Unterricht mit der Methode Use-Modify-Create (5) hervorzuheben. Programmierspezifisch sind darüber hinaus das Lesen von Quellcode (10) sowie die Förderung des Programmverstehens (12) u. a. mittels Tracing und Debugging. Die Literatur zu Fehlvorstellungen beim Programmierenlernen (4) ist äußerst umfangreich und sollte zur Diagnose des individuellen Lernfortschritts von den Lehrkräften herangezogen werden (vgl. SORVA 2023). Exemplarisch werden im Folgenden das Program-Tracing (12) sowie die Rolle des Programmlesens in Verbindung mit Programmierkonventionen (10) betrachtet, die eng miteinander verknüpft sind.

Program-Tracing

Es besteht kein Zweifel, dass Programmieren für jeden eine äußerst lohnende und zufriedenstellende Erfahrung sein kann. Allerdings sollten Lehrkräfte es nicht zu eilig haben, ihre Schülerinnen und Schüler ihr erstes eigenständiges Programm schreiben zu lassen, da die kognitive Belastung schnell zu hoch sein könnte. Im Deutschunterricht der Grundschule würden Schülerinnen und Schüler auch nicht zum Schreiben aufgefordert werden, bevor sie die Buchstaben und Grundlagen des Lesens gelernt haben. Ebenso gibt es zahlreiche Belege dafür, dass das Lesen eines Codes, den andere geschrieben haben, die Fähigkeit der Schülerinnen und Schüler verbessern und fördern kann, später selbst einen besseren Code zu schreiben.

Schülerinnen und Schüler sollten somit in die Lage versetzt werden, einen Code zu interpretieren, zu verstehen und zu manipulieren. Code lesen, genauer Code-Tracing, kann dazu beitragen, die kognitive Belastung der Lernenden zu verringern. Durch die Konzentration der Lernenden auf bestehende und funktionierende Programme und die Beantwortung spezifischer Fragen, könnten Lehrkräfte vermeiden, dass Schülerinnen und Schüler unnötig überfordert werden. Wenn die Lernenden die Möglichkeit haben, Code zeilenweise nachzuvollziehen, können sie den Code und seine Funktion verstehen, bevor sie ihn in Aktion sehen. In Abb. 4 wird eine „Notional Machine“ (Gedankenmodell vom Computer; Papiercomputer) bei der Ausführung eines Python-Quellcodes gezeigt. Hilfreich bei der Reduktion der kognitiven Belastung ist darüber hinaus die Einhaltung von Programmierkonventionen.

Zur Rolle von Programmierkonventionen beim Programmierenlernen

Lesbarer Code für eine geringere kognitive Last

Die Lernförderlichkeit von Programmierkonventionen liegt in der Strukturierung und Lesbarkeit des Codes und kann wiederum durch die CLT begründet werden. Die Einhaltung von Konventionen erleichtert nicht nur das Verständnis des eigenen Codes, sondern auch die Zusammenarbeit mit anderen Entwicklerinnen und Entwicklern. In Python sind Programmierkonventionen oft als PEP 8 (Python Enhancement Proposal 8) bekannt.

Einige konkrete Beispiele für Programmierkonventionen in Python sind:

- *Einrückung:* Python verwendet Einrückungen, um Blöcke von Codes zu definieren. PEP 8 – der Style Guide for Python Code – empfiehlt die Verwendung von 4 Leerzeichen pro Einrückungsebene.
- *Variablennamen:* Namen sollten klar und beschreibend sein. Für Variablen und Funktionen wird in Python üblicherweise „lowercase“ (alles klein geschrieben mit Unterstrichen) verwendet.
- *Maximale Zeilenlänge:* PEP 8 empfiehlt, dass Zeilen nicht länger als 79 Zeichen sein sollten, um die Lesbarkeit zu verbessern, insbesondere wenn ein Code in verschiedenen Umgebungen bearbeitet wird.
- *Import-Anweisungen:* Importe sollten am Anfang einer Datei stehen und in separaten Zeilen. Zu-

dem wird empfohlen, Standardbibliotheken, dann Drittanbieterbibliotheken und schließlich lokale anwendungs-/bibliotheksspezifische Importe in separaten Gruppen zu ordnen.

- *Kommentare und Docstrings:* Kommentare und Docstrings (mehrzeilige Kommentare, die zur Dokumentation verwendet werden) sollten ganze Sätze und dabei klar und präzise sein sowie dazu beitragen, den Code zu erläutern.

Programmierkonventionen fördern einen einheitlichen Stil, der die Lesbarkeit des Codes verbessert. Durch konsistente Einrückungen, aussagekräftige Variablennamen und klare Strukturen wird die kognitive Last beim Lesen und Verstehen des Codes reduziert. Dies erleichtert nicht nur die eigene Arbeit, sondern auch die Wartung und Weiterentwicklung des Codes.

Automatisierte Code-Analysewerkzeuge

Um die Einhaltung von Programmierkonventionen sicherzustellen, können automatisierte Code-Analysewerkzeuge wie Flake8 oder Pylint verwendet werden. Diese Werkzeuge prüfen den Code auf Konformität mit den definierten Konventionen und geben Rückmeldungen, die Entwicklerinnen und Entwickler dabei unterstützen, den Code lesbarer und wartbarer zu gestalten.

Ein einfacher „Python-Computer“ mit Funktionsaufrufen

```

1 # Wiederholte Eingabe, bis Ganzzahl zwischen von und bis
2 # eingegeben wird, und gibt die eingegebene Zahl zurück
3 def zahl_eingeben(von, bis):
4     fertig = False
5     while not fertig:
6         zahl = int(input())
7         if zahl >= von and zahl <= bis:
8             fertig = True
9         else:
10            print('Bitte eine Zahl zwischen', von, 'und',
11                  bis, 'eingeben!')
12            return zahl
13
14 # Hauptprogramm: Dauer eines Termins berechnen
15 print('In welcher Stunde beginnt der Termin?')
16 von = zahl_eingeben(0, 23)
17 print('In welcher Stunde endet der Termin?')
18 bis = zahl_eingeben(von, 23)
19 print('Dauer des Termins in Stunden:', bis - von)

```

**Globaler Speicher
Hauptprogramm:**

Variablen-Name:	Wert:
von	8
bis	

**Lokaler Speicher
zahl_eingeben():**

Variablen-Name:	Wert:
von	0
bis	23
fertig	True
zahl	8
return:	8

Aktuelle Zeile:

16

Bedingung:

not fertig
False

Konsole:

```

>>>
In welcher Stunde beginnt der Termin?
8
In welcher Stunde endet der Termin?
9
Dauer des Termins in Stunden: 1

```

Abb. 4: Python-Code mit nachvollzogener Variablenbelegung, Bedingungsauwertung und Konsolenausgabe bis zur Zeile 16 inklusive Funktionsaufruf (STECHERT 2022b)

Dokumentation als integraler Bestandteil

Die Dokumentation ist ein wesentlicher Bestandteil von Programmierkonventionen. Kommentare im Code, die erklären, warum bestimmte Entscheidungen getroffen wurden oder wie bestimmte Teile des Codes funktionieren, tragen dazu bei, die kognitive Last zu reduzieren. Dies ist besonders wichtig, wenn andere Entwicklerinnen und Entwickler den Code verstehen oder weiterentwickeln müssen.

Generell gilt: Konventionen sind nicht für Python spezifisch, sondern können auch in anderen Programmiersprachen mit entsprechenden Anpassungen angewendet werden. Typischerweise gibt es sowohl programmiersprachenspezifische als auch firmenspezifische Programmierkonventionen. Hinzu kommt, dass Programmierkonventionen zum Teil durch die Entwicklungsumgebung und die Tool-Chain im Betrieb beeinflusst werden. Insbesondere in Teams sollten deshalb zu Beginn Absprachen getroffen werden, welche Programmierkonventionen eingesetzt werden. Der Google Python Style Guide legt beispielsweise im Gegensatz zu PEP 8 bei den oben genannten Import-Anweisungen einen stärkeren Fokus auf die Klarheit, um welches Modul aus welchem Package es sich handelt, um Kollisionen zu vermeiden.

Programmierkonventionen spielen zusammenfassend auch eine Rolle im Lernprozess. Sie sind nicht nur Richtlinien zur Verbesserung der Lesbarkeit und Wartbarkeit des Codes, sondern auch wichtige Werkzeuge, um Anfängerinnen und Anfängern die strukturierte Denkweise zu vermitteln, die für effektives Programmieren benötigt wird. Konventionen reduzieren die kognitive Last und erleichtern das Verständnis von Code.

AUSBLICK: WIE VERÄNDERT KÜNSTLICHE INTELLIGENZ DAS PROGRAMMIEREN UND DEN (PROGRAMMIER-)LERNPROZESS?

Dass ChatGPT programmieren kann, hat sich natürlich bereits zu den Auszubildenden (nicht nur) der gewerblich-technischen Berufe herumgesprochen. Github Copilot ist ein intelligenter Assistent, der Entwicklerinnen und Entwickler bei der Programmierung unterstützt, indem er kontextbezogen Codevorschläge macht und auch ganze Funktionen beisteuert. Und inzwischen hat auch Meta nachgelegt und stellt ein Modell zur Verfügung, das explizit auf die Codegenerierung trainiert ist: Code Llama. Code Llama ist dabei kein komplett neues Modell, sondern von Llama 2 abgeleitet. Beeindruckend ist, dass selbst das kleinste Code-Llama-Modell dem größten Llama-2-Modell bei Programmieraufgaben überlegen ist (vgl. WINKLER 2023). Perspektivisch müssen die

Auszubildenden lernen, GitHub Copilot und Co. in ihren Entwicklungsworkflows nach jeweiliger Eignung einzusetzen und lernen, deren Vorteile bei der Codegenerierung und Fehlerbehebung zu nutzen: Pair-Programming mit der KI.

Ähnlich argumentieren SEUFERT & SPIRGI. Sie stellen für die Berufsbildung die Frage, inwieweit in Zeiten von generativer KI weiterhin Programmierkenntnisse vermittelt werden sollten (vgl. 2023) und beantworten sie wie folgt:

„Zusammenfassend scheint uns die Frage, inwieweit weiterhin Programmierkenntnisse vermittelt werden sollten, da dies nun KI-Systeme effizienter übernehmen könnten, nicht gerechtfertigt. Aufgrund der vorgenommenen Analyse wird dafür argumentiert, dass vielmehr das Gegenteil der Fall ist. Berufslernende sind bereits heute an Prompting skills (im Sinne von low coding) heranzuführen, um diese auf künftige neue Zusammenarbeitsformen mit KI-Systemen vorzubereiten“ (ebd., S. 21).

Die Frage, ob Entwicklerinnen und Entwickler bald durch Modelle, die Anforderungen selbst umsetzen können, überflüssig werden, scheint also unbegründet. Es ist stattdessen anzunehmen, dass die Bedeutung von Expertinnen und Experten und Computational Thinking sogar zunimmt. Die von den Modellen produzierten Ergebnisse benötigen eine Überprüfung, wofür Fachkenntnisse unerlässlich sind. Im Sinne des DAGSTUHL-Dreiecks (DAGSTUHL-ERKLÄRUNG 2016) ist für die kritische Anwendung von KI auch der Blick auf die technischen Grundlagen sowie die gesellschaftlichen Auswirkungen notwendig:

„Entsprechende Maßnahmen zur Fort- oder Weiterbildung sollten nicht kurzfristiges Produktwissen zu spezifischen generativen Werkzeugen oder proprietären KI-basierten Lösungen priorisieren, sondern vielmehr langfristig anwendbare Kompetenzen in den Mittelpunkt stellen. Es geht darum, KI-basierte Informatiksysteme und Anwendungen als unterstützende Instanz in kollaborativen Systementwicklungsprozessen nutzen zu können und in der Lage zu sein abzuschätzen, welche Chancen, Risiken und Grenzen damit einhergehen“ (JASCHKE et. al 2023, S. 7).

Alle sollten sich bewusst sein, dass es grundsätzliche Unterschiede zwischen menschlicher und künstlicher Intelligenz gibt und dass man sich vor unzulässigen Anthropomorphisierungen hüten sollte. Die Rolle der Entwicklerinnen und Entwickler wird sich wahrscheinlich wandeln, weg von monotonen Aufgaben hin zu mehr Kreativität. Sprachmodelle könnten Routinearbeiten übernehmen, wodurch Programmierende sich verstärkt auf die kreativen Aspekte ihrer Tätigkeit konzentrieren können.

Und weiterhin gilt: Die Cognitive Load Theory bietet einen Rahmen, um die Lernprozesse zu verstehen und zu optimieren. Die Anwendung der Prinzipien der CLT auf das Programmierenlernen mit und ohne KI-Unterstützung in den gewerblich-technischen Berufen ermöglicht es, die Grundlagen des Programmierens zu meistern und sich gleichzeitig auf spezifische Anforderungen in verschiedenen Fachrichtungen vorzubereiten.

Literatur

- BADDELEY, A. D. (2000): The episodic buffer: A new component of working memory? In: Trends in Cognitive Sciences. 4(11), pp. 417–423, doi:10.1016/S1364-6613(00)01538-2.
- DAGSTUHL-ERKLÄRUNG (2016): Bildung in der digitalen vernetzten Welt. Eine gemeinsame Erklärung der Teilnehmerinnen und Teilnehmer des Seminars auf Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH. <https://dagstuhl.gi.de/dagstuhl-erklaerung> (letzter Zugriff: 10.04.2024)
- JASCHKE, S.; KLUSCH, M.; KRUPKA, D.; LOSCH, D.; MICHAELI, T.; OPEL, S.; SCHMID, U.; SCHWARZ, R.; SEEGERER, S.; STECHERT, P. (2023): Positionspapier der Gesellschaft für Informatik e. V. (GI): Künstliche Intelligenz in der Bildung. Gesellschaft für Informatik, Bonn.
- JENEWEIN, K. (2020): Zum Verhältnis der Didaktik der beruflichen Fachrichtung Elektrotechnik zur Didaktik der beruflichen Fachrichtung Informationstechnik/Informatik. In: Grimm, A. (Hrsg.): Didaktik der beruflichen Fachrichtung Informationstechnik/Informatik. Berlin: Peter Lang Verlag.
- KMK (2019): Ländergemeinsame inhaltliche Anforderungen für die Fachwissenschaften und Fachdidaktiken in der Lehrerbildung (Beschluss der Kultusministerkonferenz vom 16.10.2008 i. d. F. vom 16.05.2019).
- MASON, R.; COOPER, G. (2012): Why the bottom 10% just can't do it - Mental Effort Measures and Implication for Introductory Programming Courses: https://www.academia.edu/21401784/Why_the_bottom_10_just_cant_do_it_mental_effort_measures_and_implication_for_introductory_programming_courses?auto=citations&from=cover_page (letzter Zugriff: 10.04.2024)
- MEBIS-LERNPLATTFORM (o. J.): Lerntheoretische Grundlagen multimedialer Lernszenarien. <https://mebis.bycs.de/kategorien/mediendidaktik/themen-im-fokus-md/lernfoerderliche-gestaltung> (letzter Zugriff: 10.04.2024)
- MÜHLING, E. (2019): Didaktik der Programmierung. IQSH-Weiterbildung Informatik. WiSe 2019/20.
- PEP 8: <https://peps.python.org/pep-0008> (letzter Zugriff: 10.04.2024)
- PQR (2019): Pedagogy Quick Read #1: Cognitive Load Theory. <https://raspberrypi-education.s3-eu-west-1.amazonaws.com/Quick+Reads/Pedagogy+Quick+Read+1+-+Cognitive+Load+Theory.pdf> (letzter Zugriff: 10.04.2024)
- REY (2009): E-Learning. Theorien, Gestaltungsempfehlungen und Forschung. <http://www.elearning-psychologie.de/clt.html> (letzter Zugriff: 10.04.2024)
- SEUFERT, S.; SPIRGI, L. (2023): Programmieren im Zeitalter der generativen KI: eine transversale Kompetenz in der Berufsbildung? https://www.bwpat.de/spezial20/seufert_spirgi_spezial20.pdf (letzter Zugriff: 10.04.2024)
- SORVA, J. (2023): Misconceptions and the Beginner Programmer. In: SENTANCE, S.; BARENDSEN, E.; HOWARD, N. R. & SCHULTE, C. (Eds.). Computer Science Education: Perspectives on Teaching and Learning in School. 2nd Edition. Bloomsbury, S. 259-274.
- STECHERT, P. (2009): Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht. Dissertationsschrift. Fachbereich Elektrotechnik und Informatik. Universität Siegen: https://berufsinformatik.de/wp-content/uploads/2020/04/2009_stechert.pdf (letzter Zugriff: 10.04.2024)
- STECHERT, P. (2022): Cognitive Load Theory / Informatikdidaktik kurz gefasst, Teil 39. <https://youtu.be/tywFDzAAfjA> (letzter Zugriff: 10.04.2024)
- STECHERT, P. (2022b): Eine Notional Machine für Python: Funktionsaufrufe / Informatikdidaktik kurz gefasst, Teil 45. <https://youtu.be/7AOs3aAsLw4> (letzter Zugriff: 10.04.2024)
- STECHERT, P. (2023): How we teach computing. 12 Prinzipien. Part 1 / Informatikdidaktik kurz gefasst, Teil 48. <https://youtu.be/IWRGHdgntog> (letzter Zugriff: 10.04.2024)
- SWELLER, J.; MERRIËNBOER, J. J. G.; PAAS, F. (2019): Cognitive Architecture and Instructional Design: 20 Years Later. <https://link.springer.com/article/10.1007%2Fs10648-019-09465-5> (letzter Zugriff: 10.04.2024)
- WINKLER, C. (2023): Code Llama: Ein Llama lernt programmieren. Heise Developer. <https://www.heise.de/hintergrund/Code-Llama-Ein-Llama-lernt-programmieren-9537794.html> (letzter Zugriff: 10.04.2024)

Unterrichtsprojekt: Betriebsdatenvisualisierung mittels Node-RED und MQTT



JESSE SCHLÜTER



NIKOLAUS STEFFEN

Die Vernetzung und Fernabfrage von Produktionsanlagen (insb. auch bei Altbestand) ist eine praxisnahe und herausfordernde Möglichkeit, Fachkräfte der Informatik und Automatisierungstechnik in einem berufsübergreifenden Projekt zusammenzuführen. Innerhalb dieses praxisorientierten Artikels wird eine Unterrichtsidee skizziert, deren Ziel es ist, die Kommunikation zwischen einem industriellen Steuerungssystem und der IT mittels Node-RED zu realisieren. Das Projekt könnte sich z. B. im Unterricht mittels dreier Teilprojekte an Gruppen von Lernenden der Ausbildungsberufe

„Elektroniker/-in für Automatisierungstechnik“, „Fachinformatiker/-in Fachrichtung Anwendungsentwicklung“ und „Fachinformatiker/-in Fachrichtung Systemintegration“ richten. Inhalt des Projekts ist es, nach einem einfachen Start auf geringem bis mittlerem Anforderungsniveau in den berufsbezogenen Teilgruppen, die jeweiligen Gruppenergebnisse durch Kooperation zusammen zu führen, mit der Möglichkeit, die Gesamt- und Teilaufgaben im Komplexitätsgrad variieren zu können.

VORSTELLUNG DES ANLAGENZENARIOIS UND SKIZZIERUNG DER AUFGABEN DER EINZELNEN BERUFSBEREICHE

Von einer Sortieranlage (s. Abb. 1), die hier exemplarisch über eine ältere Siemens S7-300 gesteuert wird, sollen verschiedene Betriebsdaten mit der IT ausgetauscht werden, um diese zu visualisieren und zu beeinflussen. Dazu gehören u. a. die Motorparameter, wie Ist- und Solldrehzahl, sowie die Zyklus-zähler für die sortierten Metall- und Kunststoffteile.

Aufgaben für Elektroniker/-innen für Automatisierungstechnik (EAT)

Als Aufgabe der Auszubildenden der Automatisierungstechnik wird die Inbetriebnahme der Produktionsanlage und die Bereitstellung der Daten vorgesehen, mit späterem Datenaustausch zur zentralen IT. Je nach Zeitumfang lässt sich entscheiden, ob

sich auf die IT-Datenschnittstelle konzentriert wird, durch die Bereitstellung eines teilweise fertig vorprogrammierten Projekts, oder auf die Programmierung der Gesamtanlage.

Aufgaben für Fachinformatiker/-innen der Fachrichtung Anwendungsentwicklung

Die Aufgabe der Auszubildenden aus dem Bereich der Anwendungsentwicklung besteht darin, die Entwicklung einer Applikation vorzunehmen, die die Daten aus der Produktionsanlage visualisiert und Steuerbefehle an die Anlage zurückliefert.

Aufgaben für Fachinformatiker/-innen der Fachrichtung Systemintegration

Die zukünftigen IT-Administratorinnen und -Administratoren sollen die notwendige Infrastruktur zur Verfügung stellen, auf der die Kommunikation der entwickelten Applikation mit dem Produktionssystem realisiert wird.

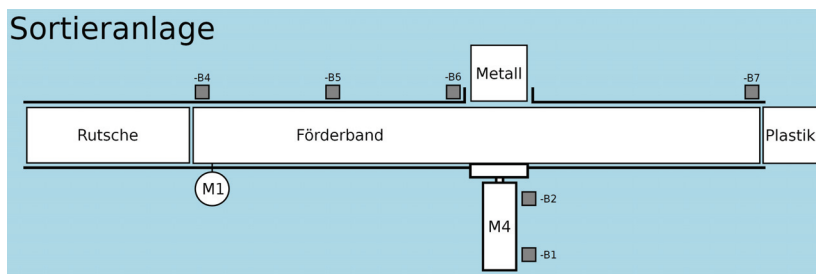


Abb. 1: Schematische Darstellung der Produktionsanlage

Berufsübergreifender Ansatz

Die Klärung aller Anforderungen und Aufgaben sollen die Lernenden der drei beteiligten Ausbildungsberufe möglichst selbst untereinander klären. Das Ergebnis könnte z. B. wie folgt aussehen:

- Auszubildende der Automatisierung

müssen sich mit denen der Anwendungsentwicklung verständigen, welche Daten sie zur Verfügung stellen und verarbeiten können. Die Beispielanlage ist ein einfaches Sortierband, für das Stückzahlen der sortierten Teile ausgegeben und die Magazingrößen beeinflusst werden können.

- Auszubildende der Anwendungsentwicklung können eine Webapplikation auf Basis von PHP und MySQL programmieren. Für die Auszubildenden der Systemintegration bedeutet das, einen Datenbankserver zur Verfügung zu stellen und die Daten von der Steuerung – die nicht auf SQL basiert – in die Datenbank zu transferieren und umgekehrt.
- Die Auszubildenden aus dem Bereich Systemintegration können sich ergänzend mit der Datensicherheit der Anlage auseinandersetzen. Hier sollte es um Netzsegmentierung und Paketfilterung gehen, aber auch um die Zugriffsrechte auf die Daten in der Datenbank und im Automatisierungssystem.

Skizzierung der praktischen Umsetzung des Projekts

Das letztendlich komplex gestaltbare Projekt bietet ein hohes Potential an Teilaufgaben und Anforderungsniveaus, deren genaue und vollumfängliche Dokumentation den Artikelumfang deutlich sprengen würde. Der Schwerpunkt dieses Artikels wird auf die Skizzierung des exemplarischen Weges zur Herstellung der Kommunikation zwischen „Maschinen-“ und „IT-Welt“ mittels Node-RED und MQTT gesetzt. Das Folgende versteht sich hierbei als „erste Annäherung“ an das Thema für fachlich Interessierte und weniger als eine „1:1-Ablaufdokumentation mit Laufgarantie“. Im folgenden Technologieschema (Abb. 2) sind der Systemumfang und die Ausführung dargestellt, dessen Umsetzung hier beschrieben wird. Am Ende des Beitrags sind relevante Links aufgeführt; u. a. ist auch der Downloadlink eines ZIP-Containers für das Projekt enthalten.

HARDWARE

Für die Umsetzung wird Hardware benötigt. Es ist versucht worden, Hardware auszuwählen, die an

Berufsschulen vorhanden sein sollte oder kostengünstig zu beschaffen ist.

Eingesetzte Hardware:

- Speicherprogrammierbare Steuerung (hier Siemens S7-300),
- Touchpanel (hier Siemens KTP700),
- Ethernet-Switch (hier Siemens XC208),
- Industrie-PC für die Kommunikation mit der SPS (hier Raspberry Pi),
- Server für die zentrale IT-Software (hier Raspberry Pi),
- Computer mit Programmierungsumgebung für die SPS (hier PC mit Siemens TIA-Framework).

Das Teuerste ist vermutlich das Produktionsanlagenmodell inklusive der speicherprogrammierbaren Steuerung (SPS). Da hier die Realisierung der Kommunikationsprozesse im Vordergrund steht und nicht z. B. die Fehlersuche an Realanlagen, wird die reale Produktionsanlage durch eine Softwaresimulation auf der eingesetzten SPS (Siemens S7-300) ersetzt und auf dem Touchpanel (Siemens KTP700) visualisiert. Alternativ zum Hardware-Touchpanel kann die Visualisierung auch mit einem PC umgesetzt werden.

Die Wahl der SPS kann relativ frei erfolgen. Wichtig ist nur, dass sie eine Kommunikationsschnittstelle zum Datenaustausch hat (bevorzugt Ethernet). Da einer der Autoren überwiegend mit Siemens-Produkten arbeitet, wird eine alte Siemens S7-300 verwendet. Das Modell ist in der Industrie noch weit verbreitet; die S7-314C sollte an vielen Berufsschulen bekannt und vorhanden sein. Des Weiteren ist diese Steuerung so alt, dass sie moderne und sichere

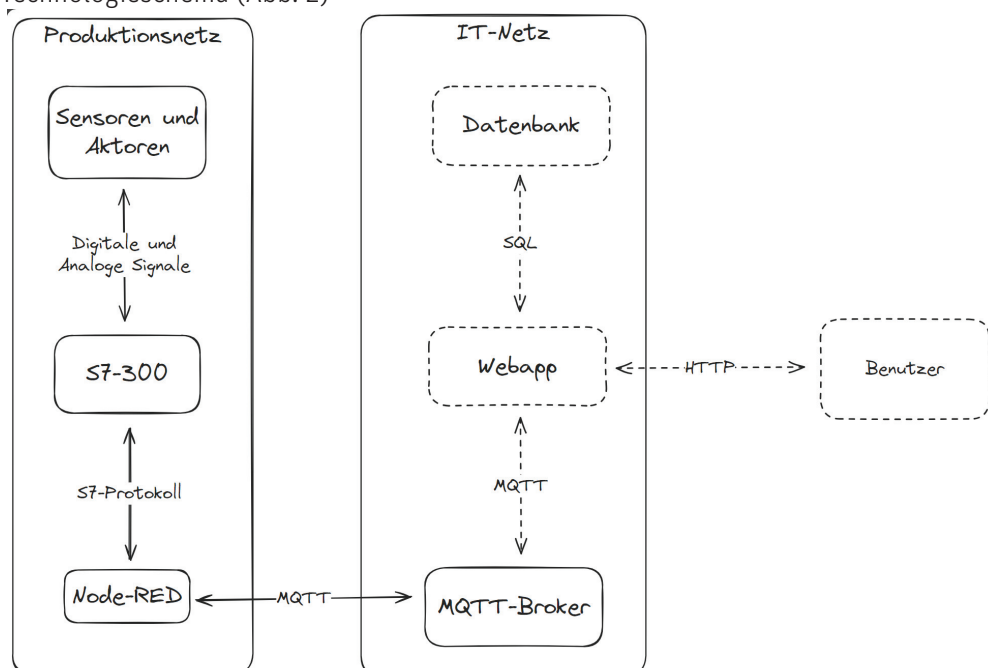


Abb. 2: Schematische Darstellung der Kommunikation

Kommunikations-Protokolle nicht von Haus aus unterstützt. Das bedeutet, dass die Kommunikation vor dem Verlassen des Anlagennetzes von der IT abgesichert werden muss.

Sowohl für den Industrie-PC auf der Automatisierungsseite als auch für den Server der übergeordneten IT wurde ein Raspberry Pi gewählt. Einerseits lässt sich dieser einfach programmieren und bedienen und andererseits sind im Internet viele Anleitungen und „Howtos“ für diese Plattform zu finden. Weiterhin ist er kostengünstig und an den Berufsschulen bereits relativ weit verbreitet (es muss nicht das neueste, größte und teuerste Modell sein).

Jede Arbeitsgruppe bekommt als Hardware eine SPS (Siemens S7-300), ein Touchpanel (Siemens KTP700) und einen Raspberry Pi. Für die Programmierung und Inbetriebnahme der Hardware steht ergänzend jeweils ein Computer mit der nötigen Software zur Verfügung. Weiterhin erhält jede Gruppe auch noch einen Ethernet-Switch. Hier wird der XC208 von Siemens eingesetzt, doch jeder andere Ethernet-kompatible Switch sollte es auch tun (Vorsicht ist bei Cisco-Geräten geboten, da deren Switches für ProfiNET eventuell korrekt konfiguriert werden müssen).

SOFTWARE

Als Software wird eingesetzt:

- TIA-Framework als Programmierungsumgebung für die S7-300,
- Mosquitto als zentraler MQTT-Broker der IT,
- Node-RED als Protokollwandler zwischen der S7-300 und dem MQTT-Broker.

Programmierungsumgebung

Als Software kommt für die Programmierung der Siemens S7-300 und des Siemens KTP700 das „TIA-Portal“ zum Einsatz. Wie oben schon erwähnt: Wer kein reales KTP700 hat, könnte dieses beim Einsatz des TIA-Portals auch simulieren.

Node-RED

Da die S7-300 eine sehr begrenzte Protokollauswahl anbietet, soll hier mit den Lernenden Node-RED

eingesetzt werden, um zwischen dem S7-Protokoll und Standard-IT-Protokollen zu übersetzen. Node-RED wurde ursprünglich von IBM für das Internet of Things entwickelt. 2016 gab IBM Node-RED als Open-Source-Software frei. Seitdem wird es von der Community stetig erweitert und unterstützt inzwischen die meisten bekannten Kommunikationsprotokolle. Node-RED ist daher ein gutes Mittel, um Daten zwischen der Steuerung (hier S7-Protokoll) und der IT auszutauschen.

Node-RED ist ein einfaches Entwicklungstool, bei dem direkt im Browser grafisch in sogenannten „Flows“ (ähnlich Flussdiagrammen, s. Abb. 3) programmiert wird. Die Flows setzen sich aus diversen Funktionsbausteinen (Nodes) zusammen. Die Nodes im Flow tauschen über die programmierten Verbindungen Nachrichten aus, die aus einem Betreff (Topic) und Inhalt (Payload) bestehen.

Die grafische Programmierung in Node-RED lässt sich leicht erlernen. Das System basiert auf JavaScript, genauer gesagt Node.js, und ist dadurch auch mit JavaScript programmierbar. Die Nachrichten in den Flows sind einfache Javascript-Objekte, die zwischen den Funktionen ausgetauscht werden.

MQTT

Mit Node-RED als universeller „Übersetzungsinstanz“ zwischen den verschiedensten Kommunikations-Protokollen sind in der Anwendungsentwicklung und Systemintegration kaum noch Grenzen gesetzt, sodass eine Vielzahl von Verbindungen und Lösungen möglich wäre. Um diesen Praxisbeitrag aber möglichst nachvollziehbar zu halten und auch die Automatisierungstechnik einbinden zu können, wird als Schnittstelle zur Informationstechnik das MQTT-Protokoll genommen. MQTT ist ein offenes Protokoll für den Austausch von Telemetriedaten zwischen Maschinen (Machine-to-Machine-Kommunikation; M2M). In diesem Fall ist eine „Maschine“ definiert als all das, was von MQTT unterstützt wird (somit hier nicht nur die „Produktionsmaschine“).

MQTT ist technisch ein sehr einfaches Protokoll und nutzt einen zentralen „Broker“, der Nachrichten ent-

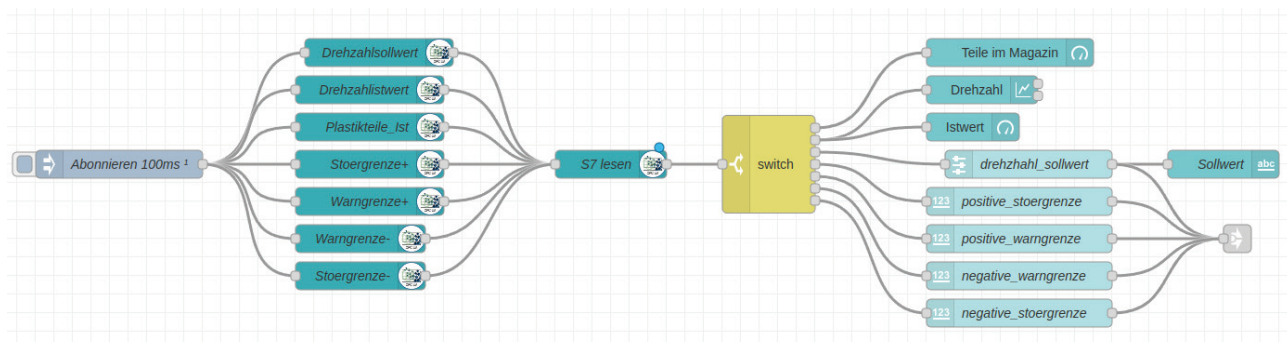


Abb. 3: Beispiel-Flow in Node-RED

Name	Datentyp	Offset	Startwert
Static			
Drehzahlsollwert	Real	0.0	14.0
Drehzahlwert	Real	4.0	0.0
SollwertFehler	Bool	8.0	false
Positive_Drehzahl	Struct	10.0	
Negative_Drehzahl	Struct	20.0	

Name	Datentyp	Offset	Startwert
Static			
AutomatikIstGestartet	Bool	0.0	false
Plastikteile_Ist	Int	2.0	0
Metallteile_Ist	Int	4.0	0

Abb. 4: Aufbau der Datenbausteine

gegennimmt und an „Subscriber“ (Abonnenten) weiterverteilt. Eine Maschine kann mittels MQTT eine Nachricht, bestehend aus einem Topic und einer Payload, an den Broker schicken. Andere Maschinen können beim Broker Topics abonnieren (Subscribe). Nachrichten, die an ein Topic gesendet werden, leitet der Broker automatisch an alle Abonnenten des selbigen Topics weiter. Zu beachten ist, dass Node-RED und MQTT dieselben Bezeichnungen für Topic und Payload benutzen, was aber nicht bedeutet, dass es sich hier um dieselben Inhalte handelt.

UMSETZUNG DES PROJEKTS

Die Anlage

Los geht es damit, die Steuerung und das Modell in Betrieb zu nehmen. Eine vorkonfigurierte Projektdatei für die S7-314C und das KTP700 ist im unten verlinkten ZIP-Container enthalten.

Wenn das bereitgestellte Projekt auf die SPS und das Touchpanel übertragen (und ggf. angepasst) wurde, kann über das KTP700 die Anlage bedient werden: Nothalt deaktivieren, Anlage einschalten, auf Automatik umschalten und Start betätigen, was zur Simulation des Teilesortierens führen sollte. Eine genaue Beschreibung der Sortieranlage und ergänzende Unterlagen zur Programmierung finden sich auf der unten verlinkten Siemens SCE-Site.

Von Interesse in diesem Anwendungsprojekt sind die Anlagendaten. Primär sind das die beiden Datenbausteine „Drehzahl_Motor“ und „Parameter“ (s. Abb. 4).

Hier stehen der „Drehzahlwert“, der „Drehzahlsollwert“ sowie die Anzahl der sortierten Teile zur Verfügung. Für den Zugriff auf die Daten über das „S7-Protokoll“ werden die „DB-Nummern“, die „Offsets“ der Variablen im DB und die „Datentypen“ der Variablen benötigt.

Raspberry Pi

Anleitungen zum Aufsetzen des Raspberry Pi (RPI) lassen sich im Internet finden. Damit der RPi später mit der Steuerung kommunizieren kann, ist dessen Netzwerkschnittstelle passend einzustellen (s. Abb. 5). Das neueste Raspberry Pi OS setzt inzwischen auf den „NetworkManager“ statt auf „dhcpcd“, deswegen sind viele bisherige Anleitungen veraltet. Mit dem Befehl „nmtui“ ist die Netzwerkschnittstelle relativ einfach anzupassen (alle hier aufgeführten Befehle müssen über die Konsole des

RPis ausgeführt werden). Für die Paketinstallation benötigt der RPi allerdings auch noch Internet (z. B. per WLAN). Später könnten die Auszubildenden der Systemintegration dafür sorgen, dass der RPi auch über die Netzwerkschnittstelle Internet erhält.

Mit den Befehlen „sudo apt update“ wird die Liste an verfügbaren Paketen aktualisiert, mit „sudo apt dist-upgrade“ alle verfügbaren Updates installiert und mit „sudo reboot“ der hier empfohlene Neustart durchgeführt.

Mosquitto

Wie oben bereits beschrieben, soll als zentrales IT-System der MQTT-Broker „Mosquitto“ zum Einsatz kommen. Zur Vereinfachung wird im Folgenden derselbe RPi für den MQTT-Broker und für Node-RED ein gesetzt.

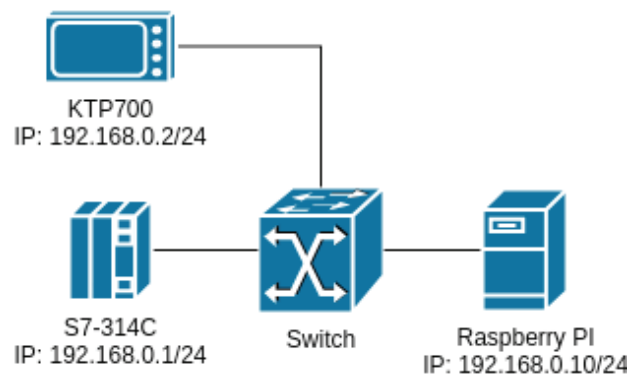


Abb. 5: Netzwerkschema

Es ist aber auch problemlos möglich, diese auf unterschiedlichen Systemen zu installieren und zu trennen, soweit, dass der MQTT-Broker auf einem zentralen RPi für alle Gruppen zur Verfügung steht.

Die Mosquitto-Installation erfolgt über den Befehl „sudo apt install mosquitto“ auf dem RPi. Eine Dokumentation zur Konfiguration des Dienstes ist auf der Mosquitto-Homepage zu finden (Link s. u.). Damit andere Systeme auf den Dienst zugreifen können, müssen ein paar wenige grundlegende Einstellungen vorgenommen werden. Hierfür wird auf dem RPi eine Datei „/etc/mosquitto/conf.d/example.conf“ mit folgendem Inhalt erstellt:

```
# Auf allen Schnittstellen arbeiten
listener 1883

# Keinen anonymen Login erlauben
allow_anonymous false

# Datei mit Benutzern und Passwörtern
password_file /etc/mosquitto/pwfile
```

Im nächsten Schritt sind die Benutzer- und Passwortdatei mit einem/einer ersten Nutzenden über den Befehl „sudo mosquitto_passwd -c /etc/mosquitto/pwfile benutzername“ zu erstellen. Der Parameter „-c“ (Create) ist wegzulassen, wenn weitere Nutzende hinzugefügt werden sollen.

Abschließend ist der MQTT-Broker über „sudo systemctl restart mosquitto“ neu zu starten. Danach ist der Dienst über die IP-Adresse des RPis, den Port 1883 und die vorher angelegten Benutzerdaten unverschlüsselt erreichbar. Dies ist eine rudimentäre Grundkonfiguration und ist natürlich nicht für den Einsatz außerhalb einer Laborumgebung gedacht. Hier wären die Auszubildenden der Systemintegration gefragt, den Dienst sicherer zu gestalten.

Zum Testen des MQTT-Brokers kann z. B. die Software „MQTT Explorer“ genutzt werden.

Node-RED

Als nächstes wird Node-RED installiert. Hierfür kann der relativ einfachen Anleitung auf der Node-RED-Website gefolgt werden. Es wird empfohlen Node-RED nach der Installation zunächst über den Konsolenbefehl „node-red“ zu starten und laufen zu lassen, bevor der Dienst in den Autostart kommt und im Hintergrund läuft. Wenn Node-RED im Vordergrund läuft, ist schneller ersichtlich, ob der Dienst korrekt arbeitet oder abgestürzt ist (inkl. der Ursache).

Sobald Node-RED gestartet ist, kann es über den Webbrowser auf dem Port 1880 auf dem RPi erreicht werden, z. B. http://192.168.0.10:1880. Über das Hauptmenü oben rechts und den Punkt „Palette verwalten“ lassen sich fehlende Funktionsbausteine

nachinstallieren. Für die S7-Kommunikation handelt es sich um das Paket „node-red-contrib-s7“. Anschließend sind die Funktionen (Nodes) „s7 in“ und „s7 out“ nutzbar, um Daten mit der SPS auszutauschen.

Nachdem die Node „s7 in“ in den Flow gezogen wurde, lassen sich deren Eigenschaften mit einem Doppelklick anpassen. Als erstes ist hier über die „Bleistift“-Funktion eine neue SPS anzulegen (s. Abb. 6). Hier ist neben der korrekten IP-Adresse auch die Angabe des „Racks“ und „Slots“ wichtig. Für die S7-300 sind das 0 und 2.

Abb. 6: Eigenschaften des S7-Endpunktes

Anschließend sind unter dem Register „Variablen“ noch die abzurufenden Daten in der SPS zu definieren. Hier kommen nun die Nummer der „Datenbausteine“, das „Offset“ und der „Datentyp“ ins Spiel. In das Feld „Adresse“ werden als erstes die Nummer des „Datenbausteins“, anschließend durch ein Komma getrennt der „Datentyp“ und das „Offset“ eingetragen (z. B. „DB2,REAL4“; s. Abb. 7). Der Name lässt sich hier frei wählen. Genauere Infos finden sich in der verlinkten Hilfe für die S7-Nodes.

Abb. 7: Variablenkonfiguration

weiter auf Seite 67

Vier neue Umsetzungshilfen für die umwelttechnischen Berufe erschienen

Neue Berufsbezeichnungen, aktualisierte Ausbildungsinhalte und neue Prüfungsanforderungen: Zum 1. August 2024 treten die modernisierten Ausbildungsordnungen für Umwelttechnologinnen und Umwelttechnologen für Wasserversorgung, Abwasserbewirtschaftung, Kreislauf- und Abfallwirtschaft sowie Rohrleitungsnetze und Industrieanlagen in Kraft. Die Umsetzungshilfen der Reihe „Ausbildung gestalten“ unterstützen das Ausbildungspersonal und erläutern die Neuerungen. Quelle: Newsletter BBNE info 02/2024

Bewerbungsstart des Nationalen Preises – Bildung für nachhaltige Entwicklung

Die Bewerbungsphase für den „Nationalen Preis – Bildung für nachhaltige Entwicklung“ (BNE) 2025 hat begonnen! Mit dem Preis würdigen das Bundesministerium für Bildung und Forschung und die Deutsche UNESCO-Kommission herausragende Initiativen, die Menschen durch BNE zu zukunftsfähigem Denken und Handeln befähigen. Angesprochen sind neben Unternehmen oder Einrichtungen der Berufsbildung auch Kitas, Schulen, Hochschulen, Vereine, Netzwerke, Kommunen oder Einzelpersonen. Bewerbungen können ab sofort bis zum 15. September 2024 über die BNE-Akteursprofile online eingereicht werden. Quelle: Newsletter BBNE info 02/2024

BIBB startet Portal für berufliche Orientierung – zynd.de fördert Selbstreflexion und Entscheidungsfähigkeit

Der Übergang von der Schule in den Beruf ist ein Prozess, den es zu begleiten gilt. Bei der Berufsorientierung sollen sich Jugendliche einerseits ihrer Interessen und Kompetenzen bewusst werden. Andererseits müssen sie diese mit den sich ständig ändernden Anforderungen der Arbeitswelt abgleichen und ihre Entscheidungen entsprechend abwägen.

INTRO

Die Integration von Geflüchteten in den Arbeitsmarkt ist eine wichtige Herausforderung für unsere Gesellschaft. Berufsbildung spielt dabei (auch) eine entscheidende Rolle, um ihnen den Einstieg in den Beruf zu erleichtern und ihre Chancen auf eine erfolgreiche Teilhabe zu erhöhen. Viele Geflüchtete bringen bereits Qualifikationen und Berufserfahrungen aus ihren Herkunftsländern mit. Dennoch sind oft Anpassungen und Nachqualifizierungen erforderlich, um den Anforderungen des deutschen Arbeitsmarktes gerecht zu werden. Sprachkurse sind der erste Schritt, um die Kommunikation im Berufsalltag zu ermöglichen. Darüber hinaus bieten Bildungsträger und Unternehmen spezielle Ausbildungsprogramme an, die auf die Bedürfnisse von Geflüchteten zugeschnitten sind. Neben fachlichen Inhalten vermitteln diese Programme auch interkulturelle Kompetenzen und Kenntnisse über die Arbeitswelt in Deutschland. Praktika und Mentorenprogramme erleichtern den Einstieg und fördern den Austausch mit erfahrenen Fachkräften. Durch Berufsbildung erhalten Geflüchtete die Chance, ihre Potenziale zu entfalten und einen wertvollen Beitrag zur Wirtschaft und Gesellschaft zu leisten. Gleichzeitig profitieren Unternehmen von den Fähigkeiten und der Motivation dieser Zielgruppe. Das BIBB bietet zum Thema Berufsbildung und Geflüchtete unter <https://www.bibb.de/de/35066.php> eine ergiebige Auswahl an Publikationen sowie Ergebnissen und Erfahrungen aus Forschung und Praxisprojekten an.

Michael Sander

Vor diesem Hintergrund versteht sich das neue Portal „zynd“ des Bundesinstituts für Berufsbildung (BIBB) sowohl als Angebot für Jugendliche als auch und vor allem für pädagogische Fachkräfte – denn deren Anliegen ist es, junge Menschen im Prozess der beruflichen Orientierung erfolgreich zu unterstützen. Bei der Eröffnung des „Somers der Berufsausbildung“ hat die Bundesministerin für Bildung und Forschung, Bettina Stark-Watzinger, heute offiziell den Online-Gang von „zynd“ eingeläutet. Das Portal war im Auftrag des Bundesministeriums für Bildung und Forschung (BMBF) entwickelt worden. „zynd“? Das steht für den Startschuss in eine wichtige Phase, den Anstoß, das Aktivwerden. Es geht darum, auf der Suche nach einem beruflichen Weg irgendwann für etwas zu „brennen“; es geht um das Feuer, das im Zweifel zu entfachen ist. Und der Name ist auch Programm für das Angebot dahinter, das junge Menschen auf verschiedenen Ebenen „entflammen“ lassen soll – bei der Ansprache, der Gestaltung sowie der technischen Umsetzung zielgrup-

pengerechter Medieninhalte. Bei der Fachkräftesicherung kommt qualitätsgesicherten, digitalen Angeboten zur beruflichen Orientierung eine zentrale Bedeutung zu. Mit dem Portal „zynd“ ist es gelungen, diese in den gesamten Prozess der Übergangsgestaltung einzubetten. Das BIBB unterstützt so die Arbeit der vielen Fachkräfte, Coaches und Beratenden, die junge Menschen in den unterschiedlichen Bildungskontexten engagiert begleiten. Im Zentrum stehen bei „zynd“ die „Playlets“ – spielerische Tools zum Selbstlernen, die sich zwischen Games und klassischen Lernangeboten verorten lassen. Die frei zugänglichen, in ein Gamification-Konzept eingebetteten „Playlets“ sollen in erster Linie zur Selbstreflexion anregen sowie entsprechende Handlungen und Entscheidungen fördern. Das Angebot reicht von interaktiven 360°-Berufsfeldpanoramen über die Reflexionsmodule „Blick in die Zukunft“ und „Selbst- und Fremdeinschätzung“ bis zu kleineren Lernmodulen wie „Who's the Fachkraft? – Das Quiz“.

Was und wann

10.10.2024 – 11.10.2024: Lern- und Arbeitsprozesse für die Transformation gewerblich-technischer Facharbeit M3 – Mittelstand macht Mixed-Learning, Siegen, gtw, Details: <https://www.tvd-edu.com/gtw-2024.html>

17.03.2025 – 19.03.2025: Hochschultage Berufliche Bildung 2025. Nachhaltig – Digital – Chancengerecht. Zukunftsszenarien von Arbeit, Bildung und Beruf. 23. Hochschultage Berufliche Bildung Universität Paderborn, Details: <https://www.uni-paderborn.de/htbb2025>

Das Portal schließt eine Lücke. Denn eine digitale Begleitung am Übergang Schule-Beruf, die diesem reflexiven Ansatz folgt, findet bisher kaum statt. Im geschlossenen Bereich von „zynd“ können die Fachkräfte mit den Jugendlichen Kontakt halten, etwa in Form eines Austausches über die Erkenntnisse aus den gespielten Modulen. Die Fachkräf-

te erhalten als registrierte Mitglieder nicht nur Zusatzinformationen und begleitende Materialien zu den einzelnen Angeboten, sondern außerdem Zugang zu diversen Funktionen pädagogischer Arbeit. Beispiele dafür sind Dateiablage, Videokonferenz und Kalender. Über Gruppenräume und sogenannte „Coaching Zones“ können sie mit den

Jugendlichen kommunizieren und diese individuell begleiten. Die registrierten Jugendlichen selbst können über ihr Dashboard nachhalten, an welcher Stelle ihres persönlichen Entwicklungsprozesses sie sich gerade befinden. Weitere Informationen unter www.zynd.de.
Quelle: BIBB-Pressemitteilung 19/2024

PROTOKOLL DER ORDENTLICHEN 7. MITGLIEDERVERSAMMLUNG DER BUNDESARBEITSGEMEINSCHAFTEN FÜR BERUFSBILDUNG IN DEN FACHRICHTUNGEN ELEKTROTECHNIK, INFORMATIONSTECHNIK, METALLTECHNIK UND FAHRZEUGTECHNIK E.V. IM RAHMEN DER FACHTAGUNG 2024 AM 08.03.2024 IN MÜHLHAUSEN/THÜRINGEN

Ort: Deutsches Bratwurstmuseum, Am Stadtwald 60, 99974 Mühlhausen

Beginn: 18:25 Uhr, Ende: 19:35 Uhr

Anwesende: 31, siehe Anwesenheitsliste

Protokoll Brigitte Schweckendieck

Tagesordnung

1. Formalia
2. Wahl des Protokollführers
3. Grundsätze der Tätigkeit und Bericht des Vorstandes
4. Bericht des Schatzmeisters, Bericht der Kassenprüfer
5. Entlastung des Vorstandes
6. Neuwahl des Vorstandes, Bestellung besonderer Vertreter, Wahl der Beiräte gemäß §§ 6 und 7 und Berufung der Landesvertreter gemäß §8
7. Entwicklung und Zukunft der BAG Elektro-, Informations-, Metall und Fahrzeugtechnik e.V.
8. Verschiedenes

PROTOKOLL

TOP 1 Formalia

Der Vorsitzende der BAG Elektro-, Informations-, Metall und Fahrzeugtechnik e.V., Uli Neustock, begrüßt die Anwesenden, weist auf die fristgerecht erfolgte Einladung, eröffnet die Mitgliederversammlung, stellt die Beschlussfähigkeit fest und weist auf den heutigen Weltfrauentag hin. Er dankt ausdrücklich allen für ihr Engagement, den Vortragenden und mit der Organisation Betrauten, die dies alles ehrenamtlich erfüllen.

Die Tagesordnung wird wie vorgelegt angenommen.

Das Protokoll der letzten Sitzung (2022) wird ohne Änderungen einstimmig angenommen. Hinweis: Die Protokolle sind auch jeweils auf der Homepage der BAG ElektroMetall zu finden.

TOP 2 Wahl des Protokollführers

Die Mitgliederversammlung wählt einstimmig bei einer Enthaltung Brigitte Schweckendieck zur Protokollführung der anstehenden Sitzung.

TOP 3 Grundsätze der Tätigkeit und Bericht des Vorstandes

Herr Neustock informiert die Mitgliederversammlung über die Aktivitäten des Vorstandes. Einzelheiten und detaillierte Informationen können beim Vorstand der BAG nachgefragt werden.

- Herr Neustock gibt zunächst einen Überblick über die Entwicklung der Mitgliederzahlen (381). Danach hat sich die Zahl der Mitglieder auch in den letzten Jahren stabilisiert, die BAG ist in fast allen Bundesländern vertreten.
- Herr Vollmer veranschaulicht in einer Grafik die regional ungleiche Verteilung der Mitglieder. Hier lässt sich weiterhin ein recht starkes Gefälle zwischen dem Nordwesten und dem Südosten Deutschlands ausmachen. Er motiviert die Anwesenden, insbesondere in den nicht so stark vertretenen Regionen um Mitglieder zu werben.
- Die 2020 geplante Fachtagung an der BS 13 in Hamburg-Wilhelmsburg musste coronabedingt abgesagt werden. Die daraufhin im Folgejahr vorbereitete musste aufgrund der Pandemie ebenfalls abgesagt werden. Stattdessen wurde kurzfristig eine Online-Tagung „Hybrides Lernen – ein Zukunftskonzept für die gewerblich-technische Berufsbildung“ organisiert, die mit 190 Teilnehmenden sehr gut wahrgenommen wurde. Endgültig durchgeführt werden konnte sie dann im September 2022. Am 20.-22.03.2023 beteiligte sich die BAG an den Hochschultagen Berufliche Bildung in Bamberg.
- Es sind zwei Tagungsbände geplant, ein gemeinsamer für die Jahre 2023/24 – die Finanzierung ist gesichert.

- Zum 40-jährigen Bestehen der BAG ist eine Sonderausgabe der Zeitschrift lernen&lehren herausgegeben worden. Diese Sonderausgabe, die auch gleichzeitig eine Arbeitsdokumentation der BAGen darstellt, hat Kosten von über 10.600,- € verursacht.
- Lernen & lehren erscheint nach wie vor viermal jährlich in Papierform, ein knappes Drittel der Mitglieder hat sich für eine Zustellung als Onlineformat (per pdf) entschieden.
- AK VT: Der Sprecher des Arbeitskreises Versorgungstechnik (Harald Strating) berichtet, dass die Tagungen gut besucht und gelaufen seien, jedoch sei die Abfolge HT/FT sehr kurz gewesen. Er begrüßt, dass es nun wieder zum regulären Zwei-Jahres-Rhythmus kommt.
- AK FT Fahrzeugtechnik (Matthias Becker) musste kurzfristig seine Teilnahme absagen, es gibt einen spürbaren Mitgliedermangel.
- BAK FST: Wolfgang Hill hat aus persönlichen Gründen kurzfristig abgesagt, hat jedoch einen Überblick über die Aktivitäten des BAK zugesichert. Es wird nach wie vor versucht, die Strukturen bezüglich der Anrechnungen und Durchlässigkeiten zwischen Fachschulausbildung und Hochschulstudium zu bedarfsgerecht zu modernisieren.
- AK PT: Der Arbeitskreis Produktionstechnik lag längere Zeit brach, Andreas Lindner möchte ihn gern wiederaufleben lassen und sich als Sprecher hierfür zur Verfügung stellen.

TOP 4: Bericht des Schatzmeisters, Bericht der Kassenprüfer

In Abwesenheit von Herrn Sander als Schatzmeister der BAG berichtet Herr Neustock über die Finanzlage der BAG, sie sei ordnungsgemäß wie immer. Der Kassenstand am 07.03.2024 betrug 18.084,23 € (der Beitragseinzug 2024 steht noch aus) und der mittlere jährliche Überschuss 868€.

Thomas Vollmer und Stefan Nagel haben den Bericht nach eigenen Angaben gründlich und gewissenhaft geprüft und kommen zu dem Ergebnis, dass er sehr korrekt geführt wurde und sachlich sowie rechnerisch stimmig ist. Grüße und Anerkennung an den Schatzmeister für seine gute Arbeit.

TOP 5: Entlastung des Vorstandes

Georg Spöttl stellt den Antrag, den Schatzmeister und den Vorstand zu entlasten. Der Vorstand und der Schatzmeister werden einstimmig entlastet.

TOP 6: Neuwahl des Vorstandes, Bestellung besonderer Vertreter, Wahl der Beiräte gemäß §§ 6 und 7 und Berufung der Landesvertreter gemäß §8

Herr Vollmer erklärt sich bereit die Wahl des Vorstandes zu moderieren. Martin Hartmann gibt sein Amt ab, es wird ihm für seine bisherige Arbeit gedankt.

Wahl des Vorstandes

Funktion	Name*	Ergebnis
Erster Vorsitzender	Ulrich Neustock	30 Ja, 0 Nein, 1 Enthaltung
Erster Stellvertreter und Sprecher (ET/IT)	Sören Schütt-Seyed	30 Ja, 0 Nein, 1 Enthaltung
Erster Stellvertreter und Sprecher (MT/FT)	Torben Karges	30 Ja, 0 Nein, 1 Enthaltung
Weitere Vorstandsmitglieder		
Stellvertr. Sprecher (ET/IT)	Wilko Reichwein	30 Ja, 0 Nein, 1 Enthaltung
Stellvertr. Sprecher (MT/FT)	Tim Richter-Honsbrok	30 Ja, 0 Nein, 1 Enthaltung
Schatzmeister	Michael Sander	31 Ja, 0 Nein, 0 Enthaltung
Erster Kassenprüfer	Thomas Vollmer	31 Ja, 0 Nein, 0 Enthaltung
Zweiter Kassenprüfer	Stefan Nagel	31 Ja, 0 Nein, 0 Enthaltung

*Personendaten siehe Anlage A1

Ulrich Neustock, Sören Schütt-Seyed, Torben Karges und alle weiteren Gewählten nehmen die Wahl an.

Ferner wurden die besonderen Vertreter der einzelnen Arbeitskreise, die Beiräte für I&I sowie die Landesvertreter gewählt bzw. bestätigt.

Bestätigung der besonderen Vertreter

Funktion	Name	Ergebnis
Bundesarbeitskreis Fachschule für Technik (BAK FST)	Wolfgang Hill	31 Ja, 0 Nein, 0 Enthaltung
Arbeitskreis Fahrzeugtechnik (AK KFZ)	Matthias Becker	30 Ja, 0 Nein, 1 Enthaltung
Arbeitskreis Versorgungstechnik (AK VT)	Harald Strating	30 Ja, 0 Nein, 1 Enthaltung
Arbeitskreis Produktionstechnik (AK PT)	Andreas Lindner	30 Ja, 0 Nein, 1 Enthaltung

PROTOKOLL MITGLIEDERVERSAMMLUNG

Bestätigung und Wahl der Beiräte

Funktion	Name	Ergebnis
Ständiger Beirat f. die Hrsg. v. Literatur	Wilko Reichwein	30 Ja, 0 Nein, 1 Enthaltung
1. Ständiger Beirat für Tagungsmanagement	Ulrich Schwenger	30 Ja, 0 Nein, 1 Enthaltung
2. Ständiger Beirat für Tagungsmanagement	Sören Schütt-Sayed	30 Ja, 0 Nein, 1 Enthaltung

Alle Beiräte nehmen die Wahl an. Thomas Vollmer wird für seine Beiratsarbeit (Herausgabe von Literatur) gedankt.

Wahl der Landesvertreter gem. §8

Die Berufung der Landesvertreterin und der Landesvertreter erfolgte als Listenwahl.

Bundesland	Namen	Ergebnis
Baden-Württemberg	Lars Windelband	31 Ja, 0 Nein, 0 Enthaltung
Bayern	Peter Hoffmann	
Berlin/Brandenburg	Carolin Lohse	
Bremen	Olaf Herms	
Hamburg	Wilko Reichwein	
Hessen	Uli Neustock	
Mecklenburg-Vorpommern	Christine Richter	
Niedersachsen	Matthias Becker	
Nordrhein-Westfalen	Thomas Wessler	
Rheinland-Pfalz	N.N.	
Saarland	N.N.	
Sachsen	Dirk Wohlrabe	
Sachsen-Anhalt	Frank Wengemuth	
Schleswig-Holstein	Maik Jepsen	
Thüringen	Matthias Grywatsch	

Alle Landesvertreter nehmen die Wahl an.

Der Vorstand wird beauftragt, Landesvertreter für Saarland und Rheinland-Pfalz zu finden und für andere Bundesländer, wenn in der Zeit bis zur nächsten Mitgliederversammlung ein Wechsel erforderlich sein sollte.

TOP 7: Entwicklung und Zukunft der BAG Elektro-, Informations-, Metall und Fahrzeugtechnik

Es sollen mehr Mitglieder geworben werden, darüber wird diskutiert. Vorschlag von G. Spöttl: eine Werbestrategie entwickeln. Aktion an den Studienseminaren, ggf. mittels des Sonderheftes? Tim Richter-Honsbrok weist auf die diesjährige Young Teacher-Aktion hin, während der es gute Ideen gab. Ist ein Facelifting erforderlich? Wo soll/will die BAG hin? – Es soll eine Arbeitsgruppe gebildet werden, hierzu wird Uli Neustock alle interessierten Mitglieder einladen (per E-Mail).

Hinweis darauf, dass bei Neumitgliedschaft für Studierende und Referendare im ersten Jahr der Mitgliedschaft kein Beitrag gezahlt werden muss.

TOP 8: Verschiedenes

Lars Windelband bemängelt die verspätete Zustellung der Werbeunterlagen an die Landesvertretungen für die Fachtagung und bittet darum, diese mindestens zwei, besser drei Monate vorher zur Verfügung zu stellen.

Herr Neustock bedankt sich für die Teilnahme und schließt die Mitgliederversammlung.

Sitzungsende: 19:35 Uhr

F.d.R.d.P.: Brigitte Schweckendieck
Mühlhausen, 08.03.2024

IMPRESSUM

Bundesarbeitsgemeinschaften für Berufsbildung in den Fachrichtungen Elektro-, Informations-, Metall- und Fahrzeugtechnik (e. V.):
BAG ElektroMetall e. V.
c/o ITB – Institut Technik und Bildung
Am Fallturm 1
28359 Bremen
04 21/2 18-66 301
kontakt@bag-elektrometall.de

Redaktion	Layout	Gestaltung
Michael Sander	Brigitte Schweckendieck	Winnie Mahrin

Nachdem die SPS-Konfiguration in Node-RED durchgeführt wurde, kann die „s7 in“-Node die einzelne Variable „Drehzahlwert“ abrufen. Um jede Sekunde den aktuellen Wert zu erhalten muss „Sende nur bei Wertänderung“ deaktiviert werden.

Abschließend muss der ausgelesene Wert noch an beliebiger Stelle visualisiert werden. Zu einfachen Testzwecken eignet sich hierfür die Funktion „Debug“ (s. Abb. 9). Nach einem Klick auf „Deploy“ sollte auf der rechten Seite unter dem „Käfer“-Reiter die aktuelle Drehzahl sekundlich aktualisiert ersichtlich sein (s. Abb. 8).

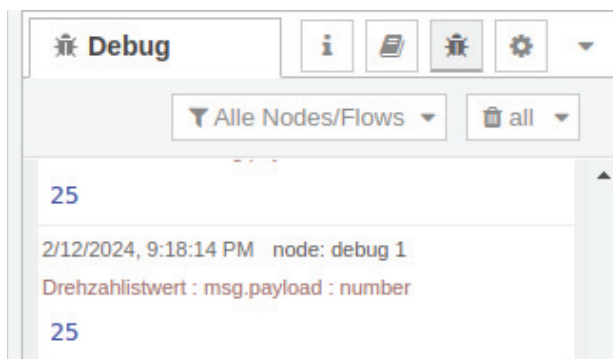


Abb. 8: Anzeige der Drehzahl

Nachdem die Verbindung zwischen Node-RED und der S7-300 getestet wurde, kann nun der Wert an den MQTT-Broker gesendet werden. Dafür ist die „s7 in“-Node mit einer „mqtt out“-Node zu verbinden. (s. Abb. 9).

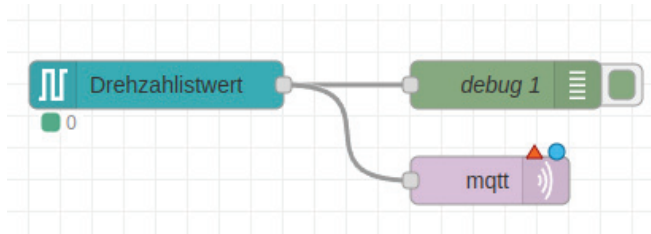


Abb. 9: Übermittlung des Drehzahlwerts an den MQTT-Broker

In den Einstellungen der „mqtt out“-Node muss als erstes der Server (die IP-Adresse und Port des MQTT-Brokers) hinterlegt und unter „Security“ der oben vergebene Benutzername und dessen Passwort ergänzt werden. Wenn in den Einstellungen der „mqtt out“-Node kein Topic angegeben ist, so wird das Topic der Node-RED-Nachricht automatisch auch als MQTT-Topic genutzt. Nach erneutem „Deploy“ des Flows können die gesendeten Werte über den MQTT Explorer eingesehen werden.

Das Senden von Daten an die SPS ist anschließend über eine „mqtt in“-Node in Verbindung mit einer „s7 out“-Node relativ einfach umsetzbar. Dabei kann der MQTT Explorer genutzt werden, um eine Nachricht mit dem passenden Topic auf dem MQTT-Broker zu

veröffentlichen. Ein importierbarer Beispiel-Flow für Node-RED ist im bereitgestellten ZIP-Container zu finden.

ALTERNATIVEN FÜR KOMMUNIKATION UND HARDWARE

Das obige Beispiel arbeitet mit einer S7-300 und dem S7-Protokoll von Siemens. Andere Steuerungssystemhersteller bieten eventuell andere Protokolle an. So können für Node-RED z. B. auch Funktionen für die Kommunikation über Modbus (TCP und seriell) oder OPC-UA installiert werden. Auf der Node-RED Webseite ist die Flows-Bibliothek bzgl. der verfügbaren Nodes leicht durchsuchbar.

MQTT muss nicht zwingend als Protokoll zur Kommunikation mit der IT genutzt werden. Node-RED unterstützt auch verschiedenste SQL-Verbindungen und Web-APIs. Wer ganz auf ein zentrales System verzichten möchte, könnte auch über Node-RED ein Dashboard erstellen.

Ein Raspberry Pi würde in einer industriellen Realanlage kaum eingesetzt. Stattdessen werden von den Steuerungstechnikherstellern passende Industrie-PCs angeboten (auch als „Ausbildungspakete“), die auch in der Schule Einsatz finden könnten.

IDEEN ZUR WEITEREN AUSGESTALTUNG UND VERTIEFUNG

Zur Ausgestaltung und Vertiefung, z. B. bei Binnendifferenzierung oder Vertiefung in den Fachrichtungen, lässt sich das Projekt nach Belieben erweitern. Einige Ideen dazu sind auch in der Abb. 10 dargestellt:

- Für die Aufzeichnung und Verarbeitung von Zeit-Serien-Daten eignet sich z. B. die Software „Grafite“.
- Mit „Grafana“ gibt es ein einfaches Tool, um diese Daten in Dashboards anzuzeigen. So könnten z. B. auch die Auszubildenden der Systemintegration oder die der Automatisierung relativ einfach ein Dashboard für die „Overall Equipment Effectiveness“ (OEE) erstellen.
- Auch im Bereich Netzwerktechnik ist noch einiges machbar: In der Realität wäre das Anlagenetz vom IT-Netz durch eine Firewall zu trennen. Beispielsweise ist OPNSense eine quelloffene Firewall die auf einem Großteil der Hardware mit amd64-Architektur lauffähig ist und professionell eingesetzt wird.
- Im Bereich der Netzwerksicherheit wäre eine Mischung der Systeme umsetzbar; z. B. eine industrielle Firewall an der Produktionsanlage, die sich

über VPN mit der zentralen Firewall der IT koppeln ließe.

- Für Elektronikerinnen und Elektroniker für Automatisierungstechnik gibt es bei jedem Hersteller eine Auswahl an 24-Volt-Geräten, die auch unter „raueren“ Bedingungen Stand halten sollte.
- Vor allem kann das Thema Sicherheit in der Kommunikation generell mit allen Fachrichtungen diskutiert und praxisnah verändert werden, z. B. in dem MQTT durch SSL/TLS abgesichert wird oder z.B. auf OPC-UA gewechselt wird.

Node-RED hat sich in diversen Lehr-/Lernsituationen als einfaches Mittel bewiesen, um sich der Vernetzung von industriellen Anlagen anzunähern. Vor allem für Teilnehmende, die keinen Informatikhintergrund haben, ist es leicht zu erlernen und vielseitig einsetzbar. Gleichmaßen bietet es für Teilnehmende, die über keinen Automatisierungshintergrund verfügen, eine einfache Möglichkeit, die SPS zu beeinflussen, ohne selbst programmieren können zu müssen. Die „schönsten“ Lehr-/Lernsituationen ergaben sich mit heterogenen Gruppen, in denen sowohl IT-, als auch Automatisierungswissen vorhanden war.

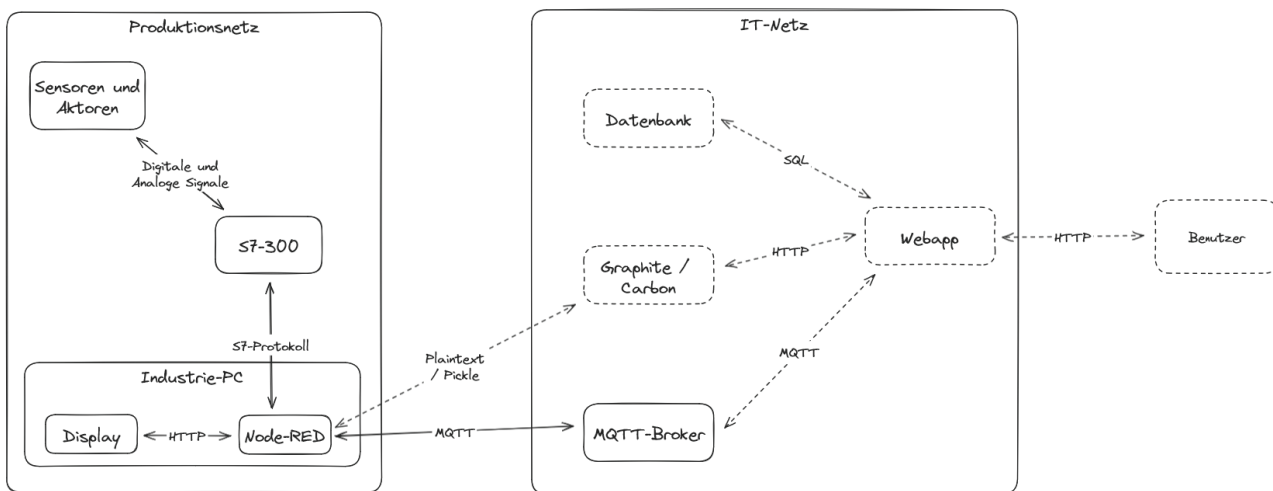


Abb. 10: Erweitertes Kommunikationsschema

Weitere Ideen zum Einsatz von Node-RED im Rahmen von Lernsituationen hat auch Stefan Sayk dokumentiert, deren Beschreibung im ZIP-Container enthalten sind. Stefan Sayk ist Fachberater für berufsbildende Schulen für den Beratungsbereich Industrie 4.0 am Regionalen Landesamt für Schule und Bildung in Osnabrück. In einer seiner beiden dokumentierten Lernsituationen wird z. B. eine KI zur Qualitätssicherung von Werkstücken eingesetzt. Die Bilderkennung geschieht über eine Standard-Webcam, deren Bildauswertungsalgorithmus über Googles Machine-Learning erstellt wurde und mittels Node-RED die Verbindung zur SPS herstellt. Weiteres Material zu den Lernsituationen von Stefan Sayk sind auf dem RLSB-OS-Moodle-Server verfügbar (s. Linkliste).

ZUSAMMENFASSUNG

Dieser Beitrag soll keine „1:1-Umsetzungsanleitung“ sein. Die Ausführungen ermöglichen vielmehr eine Annäherung und einen Zugang zu den Themen „Betriebsdatenvisualisierung an Altanlagen“ sowie „Kommunikation zwischen Automatisierungs- und IT-Welt“, in einem Feld, das sich in Breite und Tiefe noch sehr weit ausbauen lässt.

Der Einsatz von MQTT bietet einen verständlichen und schnellen Einstieg in Kommunikationsprotokolle. Es fällt vielen Teilnehmenden leichter damit zu starten, als beispielsweise mit den umfangreichen Möglichkeiten, die OPC-UA bietet. Unabhängig von den Protokollen eröffnet sich innerhalb dieser Thematik immer die Möglichkeit, auf das wichtigste Thema bei der digitalen Kommunikation einzugehen, die „Security“. Viele industrielle Automatisierungsprotokolle (wie z. B. das verwendete S7-Protokoll oder auch Modbus) wurden nicht in Hinblick auf Kommunikationssicherheit entwickelt, was sie von heutigen (modernen) IT-Protokollen unterscheidet.

Fragen zum hier vorgestellten Projekt können gern per E-Mail an die Autoren gestellt werden.

Integration von OT und IT - PLCnext Technology



GÖTZ JÄCKEL

Die aktuelle Entwicklung der industriellen Automatisierung (Industrie 4.0) ist gekennzeichnet durch die Integration von operativer Technologie (OT) und Informationstechnologie (IT), die Vernetzung allgemein und im Sinne von Industrial Internet of Things (IIoT), die künstliche Intelligenz (KI) und vieles mehr. Die PLCnext Technology erleichtert die Projektierung entsprechender Maschinen und Anlagen, indem sie die ganze Bandbreite von Industrie 4.0-Technologien durch Einsatz eines einzigen Gerätes unterstützt. Dieser Artikel stellt die PLCnext Technology vor.

KONVERGENZ VON OPERATIVER TECHNOLOGIE UND INFORMATIONSTECHNOLOGIE

Traditionell waren Informationstechnologie (IT) und operative Technologie (OT) zwei unabhängige Welten in der produzierenden Industrie. Es wurde getrennt zwischen Bürotechnik und Betriebstechnik. Zur IT der Bürotechnik gehören als im Bereich der Hardware Computer und andere Endgeräte, Server, Netzwerk-Infrastruktur usw. Die dortige Software umfasst die Verarbeitung kommerzieller Daten, die Verwaltung, das Personalwesen, die interne und externe Kommunikation. Unter OT fallen die Technologien mit Werkzeugen, Geräten, speicherprogrammierbaren Steuerungen (SPS) und Maschinen, die für den eigentlichen Produktionsprozess benötigt werden; SPS-Anwendungen und Visualisierungsprogramme oder Feldbus-Konfiguratoren sind die häufigsten Softwarekomponenten.

In beiden Bereichen kommen Computer und Elektronik zum Einsatz, die aber vor Industrie 4.0 kaum IT/OT-übergreifend verwendet wurden. So war es undenkbar, den PC zur Maschinensteuerung zu verwenden, denn die erste Assoziation mit dem PC war die Tastenkombination [ALT] + [STRG] + [ENTF] zum Neustart nach dem Absturz des Rechners. Damit war

der PC unbrauchbar für die Steuerung von Maschinen und Anlagen.

Umgekehrt reagierten die Programmierer „höherer“ Programmiersprachen auf die grafischen Entwicklungswerkzeuge der Speicherprogrammierbaren Steuerungen (SPS), den „Rechnern“ zur Maschinensteuerung, mit Unverständnis. Abgesehen davon, dass eine SPS keinen Bildschirm, keine Tastatur und keine Maus zur Verfügung stellte, war die Performance dieser Geräte „unterirdisch“ und ihre Entwicklungswerkzeuge für die Lösung von Aufgaben der IT unbrauchbar (wie z. B. Vernetzung, integrierte Datenbanken).

DER PC ALS SPEICHERPROGRAMMIERBARE STEUERUNG

Das Zusammenwachsen der IT- und OT-Welt begann, als die Steuerungshersteller ihre proprietären Programmiergeräte durch Software auf Computern ersetzen. Die lief anfänglich unter MS-DOS, später unter Windows auf Standard-PCs. Bei Siemens war das STEP 5/7, bei Phoenix Contact MULTIPROG oder PC WORX sowie bei 3S CODESYS. Damit gab es die erste Verbindung zwischen PC und SPS, wobei der PC lediglich für die Projektierung verantwortlich war. Der nächste Schritt war die Entwicklung von PC-Software, sog. Visualisierungen oder SCADA-Systemen

(Supervisory Control and Data Acquisition) zum Bedienen und Beobachten (B&B) sowie zum Management von Maschinen- und Produktionsdaten.

Eine große Herausforderung war es in beiden Fällen – der Programmierung der SPS-Programme und der Installation von B&B bzw. SCADA –, die Kommunikation zwischen PC und den SPS mit ihren unterschiedlichen Prozessoren, Geräte-Schnittstellen, Betriebssystemen und Protokollen herzustellen. Im Bereich der Programmierung kam erschwerend hinzu, dass es für die Maschinensteuerungen keine Programmierstandards gab, wie beispielsweise für die Programmiersprachen „C“, „Pascal“ oder „Basic“ für PCs. Erst mit der Normierung in der IEC 61131 „Speicherprogrammierbare Steuerungen“ wurde für die Vereinheitlichung der SPS-Programmierung gesorgt.

Trotzdem wurden allenthalben Forderungen der Industrie nach einer Integration der SPS in den PC laut: „Der PC ist so leistungsfähig und teuer, der ist mit der B&B-Software nicht ausgelastet, also soll der auch die Maschinensteuerung übernehmen, sprich die SPS überflüssig machen.“

Und in der Tat entwickelten Steuerungshersteller Systeme, bei denen eine sogenannte Soft-SPS auf einem Industrie-PC unter Windows NT die Maschinensteuerung übernahmen. Letztere wird traditionell in den SPS-Programmiersprachen nach IEC 61131 programmiert.

PROZESSDATEN: VOM „KLASSISCHEN“ FELDBUS ZUM ETHERNET

Spätestens im Zusammenhang mit der Verwendung eines PC zur Maschinen- und Anlagensteuerung geraten Netzwerke oder Vernetzung allgemein – und damit ein wesentlicher Aspekt von Industrie 4.0 – ins Blickfeld. Gemeint ist der Zugriff auf die Prozessdaten, der auf dem Höhepunkt von Industrie 3.0 mit Hilfe von Netzwerken organisiert wurde, um die parallele Verdrahtung von Ein- und Ausgängen durch Feldbusse zu ersetzen: Profibus, Interbus, Modbus, Can, DeviceNet usw. Sie arbeiten i. d. R. nach dem Master-Slave-Prinzip, wobei der Master als Geräteamschaltung in die Steuerung oder eben auch in den PC integriert werden konnte.

Parallel zu diesen industriellen Netzwerken wurde die Vernetzung von PCs im Büro- und Konsumenten-umfeld entwickelt. Die Verkabelung der Bustopologie 10BASE-T, vorher 10BASE2, mündete ab den 1990er-Jahren in die Spezifizierung von Hard- und Software im Ethernet. Ethernet hat sich als Standard durchgesetzt und inzwischen verfügt nahezu jede CPU über einen Ethernet-Controller.

Klassische Feldbusse werden angesichts dessen mehr und mehr durch auf (industriellem) Ethernet basierende Feldbusse ersetzt. Dies sind u. a. Profinet, EtherCAT oder Modbus TCP. Durch ihren Einsatz werden Gerätekosten und je nach Architektur Verkabelungsaufwand reduziert.

VON DER KONVERGENZ VON OT UND IT ZUM IIoT

An dieser Stelle sind wir mit der Kommunikationsarchitektur physisch exakt bei der Verbindung von OT und IT: Es ist nun möglich, dasselbe Kabel¹ zu verwenden für den Zugriff auf die Prozessdaten, für die Verbindung zwischen der Steuerungsarchitektur auf der einen Seite und den erforderlichen Tools für die Programmierung, die Konfiguration, die Inbetriebnahme, die Bedienung und Instandhaltung (Service, Reparatur) auf der anderen Seite.

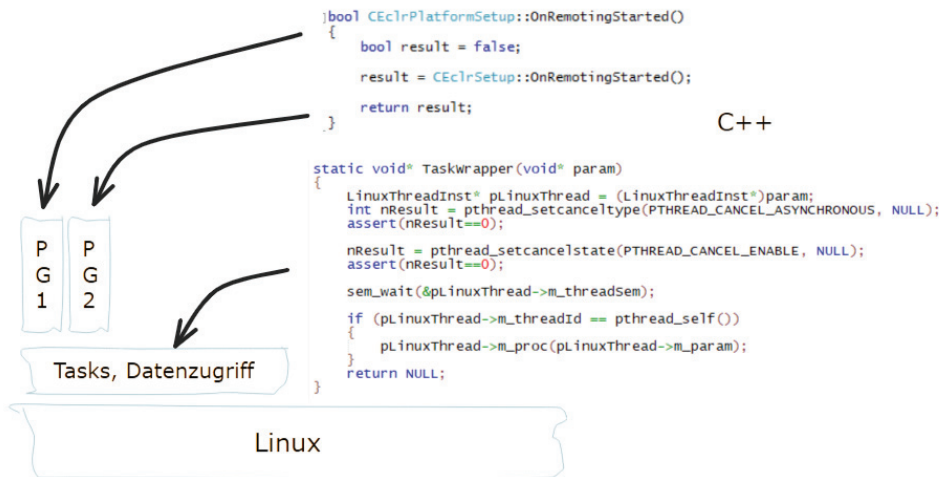
Betrachten wir nun den Datenaustausch für B&B / SCADA, die Kommunikation mit dem Management wie ERP (Enterprise Resource Planning), MES (Manufacturing Execution System) oder Predictive-Maintenance, sind wir spätestens beim Datenzugriff in die Cloud und beim Einsatz dort angebotener Services bei der Kommunikation via Internet: Informationstechnologien (Ethernet, Internet) werden für operative Technologien verwandt.

Vornehmlich im Consumer-Bereich sind diese Informationstechnologien Grundlage für das Internet der Dinge (IoT – Internet of Things), nämlich die Vernetzung von Haushaltsgeräten, Kameras, medizinischen Geräten, Smartphones u. v. m. Betrachten wir diese Form der digitalen Transformation aus dem Blickwinkel industrieller Produktion, dann sehen wir das „Thing“, Typ Roboter, das sich selbst ein „Thing“, Typ Schweißzange, anmontiert. Das funktioniert ohne Interaktion mit Menschen; wir sehen das Industrial Internet of Things (IIoT).

Die Integration von OT und IT erfordert neben der Physik die Integration auf der Ebene der Software. Hier besteht die wesentliche Herausforderung darin, die OT- und IT-Programmierwelten zu integrieren.²

VERBINDUNG DER IT UND OT IN DEN PLCNEXT-STEUERUNGEN VON PHOENIX CONTACT

Für die Verbindung der IT und OT wurde bei Phoenix Contact eine neue Steuerungstechnologie entwickelt: Das Ziel der „PLCnext Technology“ ist die angesprochene Konvergenz von OT und IT seitens der Programmierung/Projektierung abzubilden und zu realisieren. Dafür wurden die Möglichkeiten und Grenzen der Betriebssysteme klassischer SPS dem Betriebssystem Linux gegenübergestellt und diese miteinander verbunden; es handelt sich hierbei um eine neue Technologie, die SPS und Informati-



Tasks und Daten müssen mit Linux-Werkzeugen programmiert werden.

Abb. 3: Architektur und Code-Management unter Linux

- Open Source & App Integration,
- Einsatz von Echtzeit-HLL-Programmen mit Zugriff auf die OS API (Operating System; Application Program Interface),
- Zukunftssicherheit durch modulare Erweiterbarkeit,
- integrierte Echtzeitfähigkeit,
- integrierte Verbindung zur Cloud,
- integrierte Security.

Funktionsumfang der PLCnext AXCF 2152-Controller

Die PLCnext Technologie mit ihren Steuerungen zeichnet das Folgende aus:

- Zukunftssicherheit, Erweiterbarkeit:

Die offene Architektur der Technologie erlaubt es, zukünftige Protokolle, Sprachsysteme, Treiber usw. in die vorhandene Umgebung einzubauen.

- Zielgruppen:

Dank der Konversion von OT und IT in demselben Gerät kann sie für die Arbeit von den unterschiedlichsten Zielgruppen

pen verwendet werden; sei es in der Projekt- oder Firmwareentwicklung, SPS-Programmierung, SCADA-Verantwortlichkeit, Netzwerktechnik, Inbetriebnahme, Instandhaltung oder Maschinenbedienung.

Projektierung und Programmierung

- Das zentrale Tool für die Projektierung und SPS-Programmierung (IEC61131) ist der PLCnext Engineer. Dieser steht in der Basisversion kostenlos zur Verfügung. Projektierung umfasst die

Code-Erzeugung, die Integration der HLL-Module, die Konfiguration des Prozessdaten-Zugriffs, des ESM, des GDS, des Netzwerks, der Cloud und einer Datenbank, das Design der Webvisualisierung, die Projektverwaltung, die Verbindung zur und das Management der Steuerung für Inbetriebnahme und Service.

- Außer den grafischen Sprachen FBD / Kontaktplan und dem ST normiert die IEC 61131-3 auch die Schrittkette (oder Ablaufsprache oder SFC – Sequential Function Chart). Ebenso wie die Sprachen entspricht das Daten- und Projektmanagement des PLCnext Engineer genau dieser Norm.
- Code-Management der Hochsprachen (C#/C++, Matlab): Wie schon erwähnt, schließt das Task-

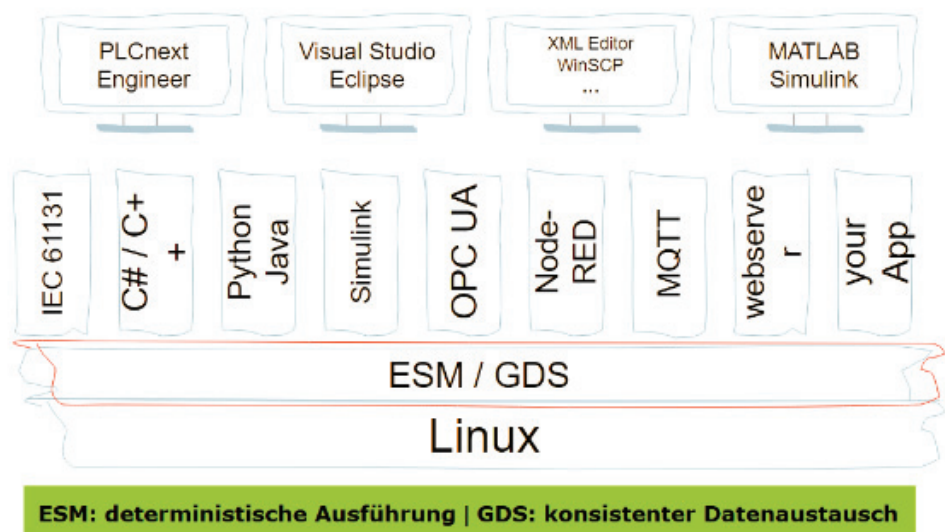


Abb. 4: Architektur und Code-Management unter PLCnext Technology

Management das Timing der Hochsprachenmodule ein (ESM). Performance und Determinismus der Module erfordern direkt ausführbaren Maschinencode statt interpretativer Abarbeitung. Aus diesem Grund wird der Binärcode vor dem Projekt-Download vom PLCnext Engineer kompiliert.

- Das Web Based Management (WBM) erlaubt den Zugriff auf die komplette Verwaltung des Controllers von der Aktualisierung der Firmware über die Zugriffsberechtigung und die Profinet-Konfiguration bis hin zur Installation von Komponenten aus dem PLCnext Store.
- Das Gesamtsystem wird Industriezweig-übergreifend eingesetzt: Prozessindustrie, Maschinenbau, Energiegewinnung, Nahrungsmittelindustrie, Automobilindustrie, Gebäudeautomatisierung u.v.m.

Kommunikation, Vernetzung

- IP-Kommunikation:

Für die Vernetzung, sei es von mehreren AXC F 2152-Controllern oder mit anderen Geräten, ist eine Bibliothek im PLCnext Engineer integriert, die auch für die Programmierung im klassischen SPS-Programm (FBD, ST) die erforderlichen Routinen zur Verfügung stellt.

- Cloud:

PLCnext-Steuerungen können prinzipiell mit jeder Cloud verbunden werden, wobei die „Proficloud“ (Cloud von Phoenix Contact) direkt unterstützt wird. Mit Apps aus dem PLCnext Store ist aber die Integration jeder anderen Cloud möglich.

- REST (REpresentational State Transfer):

PLCnext-Technologie stellt einen REST-basierten Ansatz für den Datenaustausch über HTTP mit der Firmware der PLCnext-Steuerung zur Verfügung. Er verwendet die Daten des Global Data Space.

FAZIT

Die PLCnext Technology als – je nach Perspektive – Betriebs- oder Laufzeitsystem diverser Hardware-Plattformen unterschiedlicher Leistungsklassen realisiert zuvorderst Steuerungstechnologie im Sinne von Industrie 4.0 für den industriellen Einsatz. Sie unterstützt die Anforderungen an Stabilität, Performance, Determinismus, Datenkonsistenz und Programmierbarkeit klassischer SPS und steht gleichzeitig für die Konversion von OT und IT. Die Auswirkungen auf die Qualifizierung und die Arbeit von Fachkräften sind für integrierende, domänenübergreifende Systeme dieser Art derzeit noch nicht untersucht.

Anmerkungen

- 1) Wenn wir schließlich unterstellen, dass wir uns nicht um die Schichten 1 und 2 des OSI-Modells kümmern müssen, weil das geräteseitig automatisch erfolgt, dann schließt dasselbe „Kabel“ Wi Fi Verbindungen ein.
- 2) Besonders erfahrene Protagonisten dieser Welten, die an Konvergenzprojekten arbeiten sollen, müssen motiviert werden, beide Perspektiven (OT und IT) einzunehmen und zu erkennen, wie sie ihre jeweilige Expertise einbringen und integrieren können.

Literatur

IEC 61131: Speicherprogrammierbare Steuerungen

IEC 61131-3: Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen

Fundamentale Kompetenzen in der industriellen Steuerungstechnik – Versuch einer didaktischen Strukturierung der Computer-Bedientechniken im Maschinenbau und angrenzender Bereiche



ANDREAS LINDNER

Obwohl die Nutzung von Software in verschiedensten Formen und in verschiedensten Berufen benötigt wird, gibt es derzeit noch wenig Hinweise darauf, wie deren Anwendung konzeptionell erlernt werden kann. Häufig dominieren „Klickanleitungen“, die keine wesentliche strukturelle Kompetenzentwicklung für die Softwarenutzung ermöglichen und sehr schnell veralten. Aufbauend auf dem „Konzept der fundamentalen Ideen“ von BRUNER (1977) werden in diesem Beitrag „fundamentale Kompetenzen“ für die Vermittlung von Computer-Bedientechniken im Maschinenbau entwickelt, um hersteller- und softwareunabhängig aus- und weiterzubilden.

AUSGANGSLAGE

Die zunehmend disruptiven Änderungen in den Konstruktions-, Fertigungs- und Steuerungstechnologien werden subsummiert unter „Industrie 4.0“. Beispielsweise sei hier der um sich greifende Einsatz von kollaborativen Robotern (Cobots), fahrerlosen Transportsystemen und der additiven Fertigungsverfahren genannt.

Demgegenüber steht ein eklatanter Mangel an didaktischen Reaktionen auf diese Entwicklung. Ein Großteil der informatischen Bildungskonzepte, auch bei professionellen Buchverlagen sowie einer Vielzahl von Videoanleitungen, z. B. auf YouTube, beschränkt sich bisher auf reine „Klickanleitungen“. Diese sind häufig bereits mit dem nächsten Update, bei dem sich Icons geändert haben oder die Anordnung der Untermenüs umgestaltet wurde, für Lernende überholt. SCHUBERT und SCHWILL (2011) verweisen darauf, dass die Didaktik der Informatik, als ein hierfür zuständiger Bereich, eine noch sehr junge Fachdisziplin sei, beziehen sich hierbei aber insbesondere auf allgemeinbildende Schulen (ebd., S. 21 f.). Für den Bereich der beruflichen Bildung schätzen sie die Entwicklung als noch rudimentärer ein.

Verstärkt wird die Problematik durch eine traditionell verankerte Trennung der Lehrkräfte, die sich mit klassischer Automatisierungstechnik (erweitert um Roboteranwendungen), typischer Fertigung mit CNC-Maschinen, Konstruktion mit CAD-Systemen und klassischen Office-Anwendungen im berufsschulischen Unterricht beschäftigen.

GEMEINSAME FUNKTIONALITÄTEN

Softwareanwendungen sind letztlich Informatiksysteme, die zur Lösung eines Anwendungsproblems programmiert wurden (SCHUBERT/SCHWILL 2011, S. 5). Daher ist zu vermuten, dass es gemeinsame „Funktionsweisen“ aller Anwendungen gibt. Dies lässt sich auch aus der Erfahrung heraus untermauern. Anwendende wissen mit der Zeit ziemlich genau, welches „Feature“ bei der Lösung eines Problems hilft. Dessen Existenz steht außer Frage, der produktspezifische Name und seine Platzierung in den Menüs ist jedoch unklar. So wird zum Beispiel die Erzeugung eines geraden Körpers aus einer zweidimensionalen Skizze bei „Solidworks 2022“ als „Feature“ > „linear ausgetragener Aufsatz“ bezeichnet, während die gleiche Aufgabe unter „Inventor 2017“ als „Extrusion“ im Reiter „3D-Modell“ zu finden ist. Dennoch ist die Suche anhand der Funktion und Erfahrung erleichtert. Es ist daher für das Erlernen der Bedienung weniger wichtig, mit welchen „Klickfolgen“ ein Prozess des Informatiksystems ausgelöst wird. Wichtig wäre, mögliche Vorgehensweisen zur Problemlösung zu kennen und ggf. ein „geeignetes“ Informatiksystem hierfür auszuwählen. Ähnlich argumentieren SCHUBERT und SCHWILL, wenn sie fordern, dass „im Mittelpunkt [...] vielmehr die Vermittlung von Wirkprinzipien [steht], die anhand von Systemen oder Produkten studiert werden und nicht umgekehrt“ (ebd., S. 32). Sie nennen diese Kompetenz „nichtspezifischer Transfer“ (vgl. ebd., S. 60). Dieser soll sich auf langfristige Effekte beziehen, damit grundlegende Prinzipien und Denkweisen erlernt werden (vgl. ebd., S. 60). In Anlehnung an BRUNER (1977) nennen sie diese „Fundamentale Ideen der Informatik“.

In der Aus- und Weiterbildung sollte oberstes Ziel sein, grundlegende Konzepte der verschiedenen Steuerungen exemplarisch darzustellen, um den Lernenden die Möglichkeit zur Anpassung zu geben. So können sie mit Hilfe zielgerichteter Suche über Fachbegriffe in Foren, Handbüchern oder Videotutorials die passende steuerungsspezifische „Schreibweise“ für eine Anweisung und deren Lokalisierung in den Menüstrukturen finden.

KONZEPT DER FUNDAMENTALEN IDEEN NACH SCHUBERT UND SCHWILL

BRUNER (1977) war einer der ersten, der für die Mathematik ein Konzept fundamentaler Ideen präsentierte. Dabei legte er bereits Wert darauf, insbesondere die Struktur eines (akademischen) Fachs, also die Vorgehensweise bei der Problemlösung, zu betonen. Dies greifen SCHUBERT und SCHWILL (2011) auf und adaptieren das Konzept für die Informatik allgemeinbildender Schulen. Dabei definieren sie ein „Denk-, Handlungs-, Beschreibungs- oder Erklärungsschema ...“ (ebd., S. 64) als fundamentale Idee, wenn dieses die folgenden fünf Kriterien erfüllt:

1. „in verschiedenen Gebieten des Bereichs vielfältig anwendbar oder erkennbar“ sein („Horizontalkriterium“)
2. „auf jedem intellektuellen Niveau aufgezeigt und vermittelt werden können (Vertikalkriterium)“,
3. „zur Annäherung an eine gewisse idealisierte Zielvorstellung“ dienen, „die jedoch faktisch möglicherweise unerreichbar ist (Zielkriterium)“,
4. „in der historischen Entwicklung des Bereichs deutlich wahrnehmbar“ sein und „längerfristig relevant“ bleiben („Zeitkriterium“),
5. „einen Bezug zu Sprache und Denken des Alltags und der Lebenswelt“ besitzen und für das Verständnis des Faches notwendig sein („Sinnkriterium“). (SCHUBERT/SCHWILL (2011), S. 64 f.)
6. HARTMANN, NÄF und REICHERT (2006, S. 32) erweitern diese Definition der fundamentalen Idee noch um das „Repräsentationskriterium [so dass sie] sich auf verschiedenen kognitiven Repräsentationsstufen (enaktiv, ikonisch, symbolisch) darstellen lässt.“

SCHUBERT und SCHWILL weisen darauf hin, dass „in Berufsschulen und in der betrieblichen Fort- und Weiterbildung ... der spezifische Transfer [dominiert]. Die vermittelten Fertigkeiten werden überwiegend nicht in einer Weise behandelt, die bei den Lernenden Fundamentale Ideen ausbildet“ (2011, S. 60). Dennoch scheint im Umgang mit Computern und Steuerungen gerade diese Denk- und Handlungswei-

se notwendig, sie soll daher im Folgenden adaptiert werden. Im Sinne der Kompetenzorientierung wird im Folgenden von den „Fundamentalen Kompetenzen“ der Steuerungstechnik gesprochen, an die jedoch ähnliche Anforderungen gestellt werden wie an die fundamentalen Ideen der Informatik. Dabei wird die Definition der KULTUSMINISTERKONFERENZ zugrunde gelegt, wonach Fachkompetenz „...die Bereitschaft und Befähigung [bezeichnet], auf der Grundlage fachlichen Wissens und Könnens Aufgaben und Probleme zielorientiert, sachgerecht, methodengeleitet und selbstständig ... lösen und das Ergebnis ... beurteilen [zu können]“ (2007, S. 11).

DEFINITION FUNDAMENTALER KOMPETENZEN

Im Folgenden soll – in Anlehnung an SCHUBERT und SCHWILL – eine „Fundamentale Kompetenz“ (FK) vorliegen, wenn

- a) für Lernende Anwendungsfälle für die entsprechende Kompetenz ersichtlich sind (Anwendungskriterium).
Damit rückt im Wesentlichen das Zielkriterium an die erste Stelle, wird jedoch um die Verständlichkeit für die Teilnehmenden ergänzt.
- b) die entsprechende Kompetenz in verschiedenen Bereichen der industriellen Steuerungstechnik zu finden ist und zur Anwendung gebracht wird. (Horizontalkriterium)
Mit dieser Forderung wird der Eingangsidee Rechnung getragen, dass eine (einheitliche) Benennung einer Lösungsidee im Unterricht unabhängig von der Art der Steuerung (Pneumatik, Tabellenkalkulation, CAM, SPS,...) die Bildung von Analogien und damit den anwendungsübergreifenden Transfer ermöglicht und die Einarbeitung in neue Techniken und Steuerungen erleichtert.
- c) die Kompetenz auf vielen verschiedenen Entwicklungsstufen der Lernenden und in verschiedenen Arten der didaktischen Reduktion erkennbar ist (Vertikalkriterium).
Damit wird einerseits die Idee greifbarer, indem sie in verschiedenen Reduktionsstufen dargestellt werden kann, ermöglicht aber auch ein Vorgehen nach BRUNERS Spiralprinzip (1977, S. 33 ff).
- d) die entsprechende Kompetenz einen „Bezug zu Sprache und Denken des Alltags und der Lebenswelt besitzt. (Sinnkriterium)“. Diese Formulierung wird unmittelbar und wörtlich von SCHUBERT und SCHWILL übernommen (2011, S. 65), da sie in ihrer Präzision eindeutig ist und auch auf den Begriff der Kompetenz ohne weiteres anwendbar.

Weil davon auszugehen ist, dass es eine gewisse Zeit benötigt, bis sich eine Kompetenz in Sprache

und Denken des Alltags etabliert und Eingang in das Verständnis eines Fachs findet, wird hier auf das Zeitkriterium verzichtet. Damit wird bewusst das Tor aufgestoßen für zukünftige nützliche Vorgehensweisen, die zwar „neu“, aber aus dem Alltag intuitiv verständlich erscheinen.

Auf den ersten Blick scheinen das Anwendungs- und das Sinnkriterium sehr ähnlich zu sein, sich teilweise sogar zu überschneiden. Bei näherer Betrachtung kann aber wahrgenommen werden, dass das Anwendungskriterium auf den Blickwinkel der Lernenden fokussiert, während das Sinnkriterium von einem übergeordneten, didaktischen, fachwissenschaftlichen, möglicherweise sogar allgemeingültig objektiven Standpunkt aus zu beurteilen ist. Es ist sozusagen der Blick des (all-?)wissenden Lehrpersonals.

DENTIFIKATION FUNDAMENTALER KOMPETENZEN DER STEUERUNGSTECHNIK

Um fundamentale Kompetenzen (FK) der Steuerungstechnik zu finden, wurden zunächst die bisher in der Informatik beschriebenen fundamentalen Ideen, bei denen der Nachweis bereits geführt ist, gesichtet und auf ihre Umsetzbarkeit als Kompetenz geprüft. Danach wurden der gewerblich-technische Wertschöpfungsprozess und der zugehörige Datenfluss analysiert. Den einzelnen Prozessschritten wurden typische Technologien und Tätigkeiten zugeordnet. In einem hier nicht dargestellten Prozess wurden die einzelnen Kompetenzen ausgewählter Lehrplanrichtlinien den Prozessschritten zugeordnet. Dabei konnte festgestellt werden, dass sich Häufungen in den Kompetenzen ergeben, die als Kandidaten für fundamentale Kompetenzen in Frage kommen. Es ergaben sich auch Häufungen, die so allgemeinen Charakter haben, dass das Konzept um allgemeinberufliche FKs erweitert wurde. In einem Prozess des Reversierens wird versucht, weitere Kandidaten durch die Suche nach ähnlichen Vorgehensweisen in den verschiedenen Technologien (dem Horizontalkriterium entsprechend) zu identifizieren. Auch diese Ideen müssen dann auf die Erfüllung aller Kriterien hin untersucht werden.

Während der Analyse konnte festgestellt werden, dass es offensichtlich Kompetenzen gibt, die fundamental sind, aber weitere fundamentale Kompetenzen in der Durchführung benötigen und andere, die – weil sie keine weitere Kompetenz benötigen – fundamentaler sind. Diese sollen hier als basale Kompetenzen bezeichnet werden. Alle basalen Kompetenzen sind fundamental, aber offenbar sind nicht alle Kompetenzen basal. Damit ist eine Entscheidungshilfe angeboten, in welcher zeitlichen Abfolge die Einführung erfolgen kann. Didaktische Prinzipien wie induktives vs. deduktives sowie handlungsorientier-

tes vs. fachsystematisches Vorgehen müssen jedoch berücksichtigt werden.

Mit zunehmendem Umfang und Komplexität der Tätigkeiten, die dem Kern nach mit Steuerungstechnik verbunden sind, kommen neue Kompetenzen hinzu, die derzeit noch nicht ersichtlich sind. Es wird aber davon ausgegangen, dass die Identifikation einer Reihe von FK einen deutlichen Fortschritt für Lernende und Lehrende darstellen wird, so dass sich die Suche nach einer überschaubaren Anzahl von FK bereits auszahlen wird.

Durch die vorgenannte Methodik wurden spezifisch steuerungstechnische, allgemeinberufliche und personale FK gefunden, die im Folgenden noch genauer erläutert werden sollen:

Steuerungstechnische Kompetenzen

- Kompetenz zum Algorithmisieren von Abläufen
- Kompetenz zum Treffen von Entscheidungen (basal)
- Kompetenz zur Beschreibung von Lage und Richtungen im Raum (basal)
- Kompetenz zur Übertragung von Stoff, Informationen und Energie zwischen technischen Systemen (basal)
- Kompetenz zur Wiederholung von Prozessen

Allgemeine berufliche Kompetenzen

- Kompetenz zur Dokumentation von Tätigkeiten (basal)
- Kompetenz der Nichteindeutigkeit von Lösungen (basal)

Personale Kompetenz

- Kompetenz zum kreativen Umgang mit Fehlern (basal)

Da insbesondere die Anwendung des Vertikalkriteriums strittig ist, wird eine weitere Kompetenz nicht in den Kanon der fundamentalen Kompetenzen aufgenommen, gleichwohl aber als wichtig erachtet und den fundamentalen Kompetenzen beigeordnet, die Kompetenz zur Recherche von Lösungsbausteinen.

BESCHREIBUNG DER FUNDAMENTALEN KOMPETENZEN

Die Kompetenz zum Algorithmisieren von Abläufen

Die Lernenden beschreiben technische Abläufe und Prozesse in einer Art, die eine fehlerfreie, widerspruchslöse Abarbeitung ohne Rückfragen durch andere Mitarbeitende oder technische Systeme zulässt. Dazu nutzen sie gängige, in der jeweiligen Fachdisziplin gebräuchliche, Darstellungsformen.

Die Idee des Algorithmisierens fällt sowohl in der Mathematik (vgl. STEEN, 2010) als auch in der Informatik (vgl. SCHWILL 1993, Onlineversion S. 17 f.) auf. Unter der Idee des Algorithmisierens wird die Kompetenz verstanden, eine „Klasse von Abläufen“ so zu beschreiben, dass sie fehlerfrei und ohne Rückfrage abgearbeitet werden können. Mit Blick auf die Inhalte des allgemeinbildenden Unterrichts ist die Erfüllung des Vertikal- und des Sinnkriteriums, vor allem durch Lernziele im Fach Deutsch, die auf die plausible Erstellung von Sachtexten und Beschreibungen zielen, offensichtlich.

Die Erstellung eines Plans steht im gewerblich-technischen Bereich, auch als Grundlage für ein Angebot im kaufmännischen Sinn, im Mittelpunkt der Tätigkeiten. Dabei gibt es Arbeitspläne und Prüfpläne, aber auch Ablaufpläne wie z. B. Funktionsdiagramme nach GRAFCET in Pneumatik, Hydraulik und SPS, GANTT-Diagramme im Projektmanagement und schließlich Darstellungen wie Flussdiagramme, Blockdiagramme und Struktogramme, die u. a. zur Planung und Beschreibung von Programmabläufen Verwendung finden. Auch die Darstellung von Baugruppen in CAD-Systemen in Baumstrukturen kann als „Arbeitsplan“ betrachtet werden (siehe Abb. 1). Da die Auszubildenden und Mitarbeiter je nach Gewerk diese Darstellungen nicht nur lesen, sondern auch erstellen müssen, ist nicht nur die Erfüllung des Anwendungskriteriums in diesem Zusammenhang nachgewiesen, sondern gleichzeitig auch die des Horizontalkriteriums.

Die basale Kompetenz zum Treffen von Entscheidungen

Die Lernenden verknüpfen Zustände oder Daten mit Hilfe von Vergleichen, Booleschen Operatoren und „Wenn-dann-sonst“ – Bedingungen zu Aussagen und führen damit Entscheidungen herbei.

Da ein Algorithmus sinnvollerweise nicht ohne diese Kompetenz auskommen wird, ist sie eine basale Kompetenz.

Sowohl beim Einhalten von Toleranzen als auch bei Sortiervorgängen werden offensichtlich die Vergleiche benötigt, um Entscheidungen zu begründen. Damit existieren starke Anwendungen für dieses Prinzip. Dies nutzt man explizit bei Druckbegrenzungsventilen (Pneumatik, Hydraulik), in Tabellenkalkulationen und CAD-Systemen (ein Zylinder ist die Menge aller Punkte, deren Abstand zur Achse höchstens – „kleiner oder gleich“ – den Radius beträgt). Ein Filamentdrucker startet den Druckvorgang erst, wenn die Temperatur in der Düse und auf der Druckplatte einen vorgegebenen Wert erreicht hat („gleich“). Damit gilt auch das Horizontalkriterium.

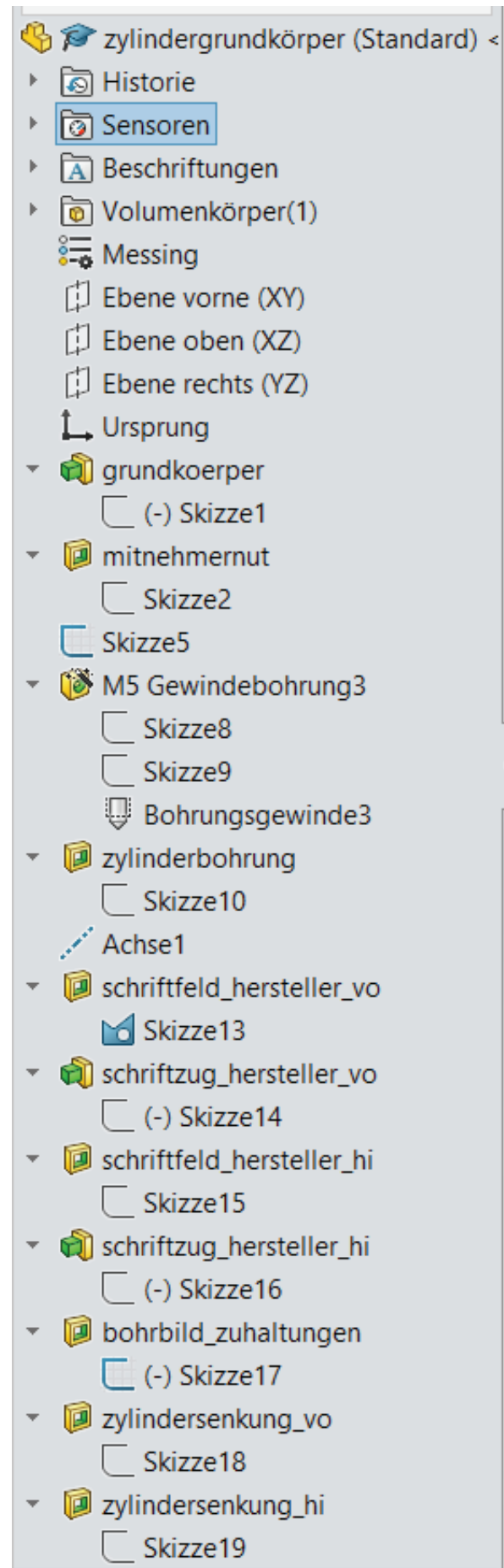


Abb. 1: Featurebaum unter SolidWorks als Beispiel eines Arbeitsplans

Ähnliches gilt für die Booleschen Verknüpfungen „und“, „oder“ und „nicht“. Sie sind offensichtlich im beruflichen Alltag zur Beschreibung von Zusammenhängen notwendig („Der Motor läuft und das Getriebe ist eingekuppelt“). Sie finden sich in verschiedensten Fachgebieten in großer Zahl wieder und werden dabei in unterschiedlichen „Darstellungsformen“ beschrieben, z. B. als Reihenschaltung („Und“) und Parallelschaltung („Oder“), entsprechende Befehle in der Tabellenkalkulation, Verknüpfung als Aufsatz („oder“) bzw. als Schnitt („und nicht“) in CAD-Systemen, die reine Aneinanderreihung von Befehlen in der PAL-Programmiersprache. Somit sind das Anwendungs- und das Horizontalkriterium nachgewiesen.

Gleiches gilt für die „Wenn... dann... sonst“-Bedingung. Sie ist nicht nur Startbedingung (Wenn die Maschine eingeschaltet ist...), sondern auch Auswahlkriterium in den unterschiedlichsten Berufs- und Lebenssituationen (wenn der Stahl gehärtet ist, verwende..., sonst...) und hat damit offensichtliche Anwendungen. Es existieren in allen Programmiersprachen und Tabellenkalkulationen Funktionen der Art „Wenn dann sonst“ oder „If then else“. Sie sind bei der Reihen- oder Parallelschaltung von Bedienelementen in der elektrischen, pneumatischen und hydraulischen Steuerung unerlässlich, werden bei der Programmierung mit Hilfe von Parametern in der CNC-Technik verwendet und bilden die Grundlage von Wiederholungen (Mustern, Bohrbildern) in CAD. Letztlich kommen sie auch bei der Entscheidung zum Einsatz, ob bei additiver Fertigung Supportmaterial zum Einsatz kommen soll oder nicht. Deutlich wird an diesem Beispiel, dass eine Entscheidung ohne den booleschen Datentyp („wahr“/„falsch“ oder „1“/„0“) nicht denkbar ist. Außerdem existiert noch ein weiterer Zustand, der als „ohne Sinn“ bezeichnet werden kann und zu vermeiden ist, da er wahlweise zu undefinierten Zuständen, unerwarteten Systemstillständen oder unklaren Fehlermeldungen (Excel: „#WERT“) führt. Der Nachweis des Vertikal- und des Sinnkriterium ist evident.

Die basale Kompetenz zur Beschreibung von Lage und Richtungen im Raum

Die Lernenden beschreiben die Lage von Objekten und deren Bewegungen im Raum. Dazu verwenden sie kartesische Koordinaten und Polarkoordinaten mit absolutem oder inkrementalem Ursprung und benennen diese. Die Lage einer Ebene können sie auch als Richtung senkrecht zu ihr interpretieren. Sie sind sich bewusst, dass manchmal nur die Beschreibung „Am Sensor“ oder „zwischen zwei Sensoren“ Verwendung finden kann.

Zum Nachweis des Horizontal- und des Anwendungskriteriums sei auf die Verwendung von kartesischen und Polarkoordinatensystemen in absoluter und inkrementaler Form in CAD- und CNC-Systemen und bei Robotern verwiesen, für die Beschreibung einer Ebene durch den Normalenvektor auf die Konstruktionsweise „normal auf“ und die Verwendung der „Rechten-Hand-Regel“. Das Vertikal- und das Sinnkriterium sind offensichtlich erfüllt.

Die basale Kompetenz zur Übertragung von Stoff, Informationen und Energie zwischen technischen Systemen

Die Lernenden ermöglichen die Übertragung von Stoff, Information und Energie von einem System zum anderen, sofern dies technisch vorgesehen ist. Dazu informieren sie sich vor Beginn der Arbeit über die Möglichkeiten der Bereitstellung am abgebenden und der Annahme am aufnehmenden Gerät und suchen – sofern möglich – nach geeigneten Adaptionsmöglichkeiten und nach Möglichkeiten, die jeweiligen Ein- und Ausgänge anzusprechen. Bei der Fehlersuche beziehen sie die Möglichkeit fehlender Kommunikationsmöglichkeit oder „Formatierungsfehler“ aktiv in ihre Überlegungen ein.

Diese Formulierung ist bewusst sehr weit gefasst. Sie umfasst letztlich die Spannweite von mechanischen Stecksystemen über angepasste Energie bis hin zu Dateiformaten und den Eigenschaften eines Datenformats in der EDV (Horizontalkriterium). Da aber der angestrebte Zweck (Informations- und Energieströme zwischen abgebender Stelle und aufnehmender Stelle anzupassen) bei all diesen Anwendungen gleich ist (Sinn- und Anwendungskriterium), wird diese allgemeine Fassung verwendet. Viele dieser Tatbestände sind offensichtlich auch bei Kindern bereits verfügbar, somit ist auch das Vertikalkriterium vorhanden.

Die Kompetenz zur Wiederholung von Prozessen

Die Lernenden entwickeln im Rahmen des Algorithmisierens Klassen von ähnlichen Abläufen, die auf Grund ihrer Häufigkeit dazu geeignet sind, sie abzubilden. Dazu nutzen sie als nützliche Sequenzen Wiederholungen, Schleifen und Unterprogramme (oder Zyklen) sowie Sprungbefehle. Sie übergeben die notwendigen Informationen, insbesondere die Abbruchbedingung, in geeigneter Art und Weise an die entsprechende Sequenz. Dazu suchen sie aktiv nach bereits vorgefertigten Sequenzen. Ansonsten sind sie in der Lage, solche Sequenzen auch unter Verwendung von Parametern selbst zu erstellen. Dazu zählt insbesondere das Konzept der Zählvariablen.

Das Horizontal- und Anwendungskriterium ist offensichtlich erfüllt, da Wiederholungen, Schleifen und Unterprogrammtechnik in allen Computersystemen gängig sind. Da Wiederholungen in nahezu allen Entwicklungsstufen angewendet werden können, ist auch das Vertikalkriterium erfüllt; der Nachweis des Sinnkriteriums ist evident.

Bei den folgenden Kompetenzen sind das Anwendungs-, Horizontal- und Sinnkriterium bereits aus der Veranlagung als sogenannte Schlüsselkompetenzen nachgewiesen. Das Vertikalkriterium ist leicht zu bestimmen.

Basale Kompetenz der Nichteindeutigkeit von Lösungen

Die Lernenden sind sich bewusst, dass es keine alleingültige Musterlösung für ein Problem gibt. Sie sind neugierig auf andere Lösungswege und untersuchen mehrere Lösungen auf ihre grundlegenden Lösungsideen sowie die jeweiligen Vor- und Nachteile. Diese Erkenntnisse nutzen sie für die Optimierung und zukünftige Problemlösungen.

Basale Kompetenz zur Dokumentation von Tätigkeiten

Die Lernenden erstellen ihre Waren und Dienstleistungen im Rahmen einer vollständigen Handlung. Dazu gehören die Phasen der Information, Planung, Durchführung und Kontrolle. In diesem Rahmen werden die Tätigkeiten dokumentiert. Dazu zählen Dokumentationen zur Rechnungsstellung und solche, die im Rahmen der Inbetriebnahme, Verwendung und Instandhaltung die Arbeit erleichtern sowie Grundlage für Gewährleistungsansprüche aller Art bilden.

Basale Kompetenz zum kreativen Umgang mit Fehlern

Die Lernenden sind sich bewusst, dass Fehler im Arbeitsprozess nicht zu vermeiden sind. Daher wenden sie geeignete, häufig prozessspezifische Kontrollmechanismen an, um Fehler frühzeitig zu finden und zu beheben. Sie begreifen jeden entdeckten Fehler primär als Chance zur Weiterentwicklung des Prozesses und dokumentieren Ursachen und Wirkungszusammenhänge für ihr Team.

Kompetenz zur Recherche von Lösungsbausteinen

Programmsequenzen werden immer häufiger aus Datenbanken und Foren zu Gesamtprogrammen zusammengefügt. Dies gewinnt mit modernen KI-Systemen zunehmend an Bedeutung, da solche Sequenzen oder auch Applets in Windeseile für beschränkte Aufgaben in den gängigen Programmiersprachen ohne Syntaxfehler zu erzeugen sind. Dieses Vorgehen macht sicherlich aus betriebswirtschaftlicher Sicht Sinn. Da

es in vielerlei Bereichen üblich ist, gilt das Horizontalkriterium. Ob es sich bei dieser Kompetenz jedoch um ein Bildungsziel handelt, wird hier zur Diskussion gestellt. „Abschreiben“ ist in den ersten Bildungsjahren verpönt und verboten. Viele Bildungsideale halten an der Beherrschung von Faktenwissen fest. Auch mit der Orientierung an Kompetenzen ist die Frage, wie viel Faktenwissen beherrscht werden soll, um trotzdem „gebildet“ zu sein, noch nicht entschieden. Daher soll hier zur Diskussion gestellt werden, ob das Vertikalkriterium als erfüllt angesehen werden kann und ob in der Folge die Kompetenz, Programmcode in Sequenzen aufzufinden und zu kombinieren, als fundamental betrachtet werden darf. Dennoch scheint es sinnvoll, diesen Weg „von Zeit zu Zeit“ im Unterricht zu thematisieren und als legale, nützliche und wirtschaftliche Lösungsstrategie darzustellen. Daher gehört diese Kompetenz in diesen Kanon, auch wenn sie in diesem Text nicht als fundamental bezeichnet werden soll.

MÖGLICHKEITEN UNTERRICHTLICHER UMSETZUNG

Eindeutigkeit der Begriffe

Damit Lernende die Chance haben, die jeweiligen Kompetenzen in verschiedenen Lernfeldern und Technologien zu transferieren, müssen Schlüsselwörter, die auf ihren Einsatz hinweisen, von allen beteiligten Lehrkräften in gleicher Weise bezeichnet werden. An solchen Stellen bedarf es eines gemeinsamen „Wordings“. Im günstigsten Fall geschieht dies einheitlich im deutschen Sprachraum, es wäre aber schon ein großer Schritt, wenn sich dieses Wording innerhalb eines Kollegiums ausbilden würde.

Sichtbarmachung der Begriffe

Damit die Anwendungsfälle der fundamentalen Kompetenzen von den Schülerinnen und Schülern (wieder-)erkannt werden, ist es hilfreich, wenn nicht sogar notwendig, ihre Verwendung sichtbar zu machen und reflektieren zu lassen sowie die Verknüpfungen zwischen den einheitlichen Formulierungen für die Kompetenzen und den jeweiligen domänenspezifischen Bezeichnungen herzustellen. Somit wird dem repetitiven Charakter dieser Kompetenzen und damit dem Horizontalkriterium Rechnung getragen. Es werden Verknüpfungen geschaffen zwischen den bisher isoliert unterrichteten „Anwendungen“ und dem gezielten Suchen nach steuerungstechnischen Lösungen für diese „Begriffe“ und Kompetenzen.

Hierzu kann in einfacher Art und Weise ein Poster im Fachraum dienen, auf dem die Erklärungen zu der jeweiligen Anwendung nach ihrer Erarbeitung festgehalten werden und somit präsent bleiben.

Da der Advance Organizer insgesamt gut geeignet ist, einen Überblick über die zu erwerbenden Fachkompetenzen zu vermitteln, ist er auch ein probates Mittel, die jeweiligen (schuleinheitlichen) Begriffe mit einfließen zu lassen. Hier wird die Darstellung in einer Mind-Map-ähnlichen grafischen Form bevorzugt, um an den entsprechenden Schlüsselstellen die zu verwendenden Begriffe einzufügen (siehe Abb. 2, graue Raute).

In situativen Fachgesprächen oder im Rahmen der Präsentation der Arbeitsergebnisse ergibt sich die Chance, die entsprechenden Begrifflichkeiten einzufordern.

Ein ähnlicher Effekt ergibt sich, wenn die Lernenden ihre Tätigkeiten reflektieren. Ob dies in der klassischen Form eines Tätigkeitsnachweises in Berichtsform oder tabellarisch als Lerntagebuch erfolgt, spielt keine Rolle. Wichtig ist, dass die Verwendung

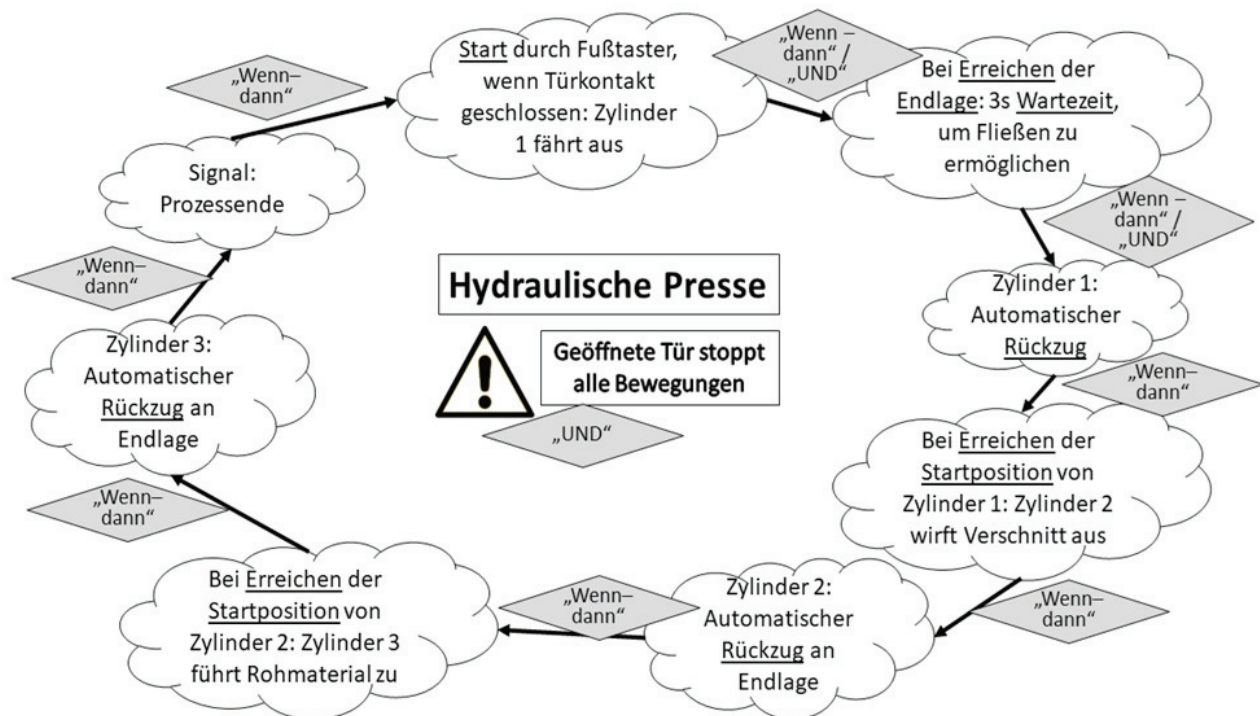


Abb. 2: Advance Organizer einer steuerungstechnischen Aufgabe

Die UND-Funktion:

- Reihenschaltung zweier Schalter
- Pneumatik:** Paralleler Anschluss zweier Ventile an einem Zweidruckventil
- „Und“ in **Tabellenkalkulationen:** UND(Bedingung1; Bedingung2; ...)

Englische Variante: AND(Argument1, Argument2, ...)

Achtung: Semikolon / Komma beachten!
- „Und“ in Sprungbefehlen in **Heidenhain iTNC 530 – CNC-Steuerung:** geschachtelte Abfrage nach Bedingung 1 (größer, gleich, kleiner,...) und dann in einem Unterprogramm die Abfrage nach der zweiten Bedingung mit Verweis in weiteres Unterprogramm (bei Nichterfüllung Rücksprung in die Zeile, die der Bedingung folgte):


```

IF Q1 GT.. GOTO LBL1
...
M30

LBL1
IF Q2 GT.. GOTO LBL2
LBL0
            
```

Abb. 3: Beispiel für Wiki-Einträge zur UND-Funktion

der standardisierten Begriffe explizit gefordert und ggf. bei Beratungsgesprächen oder Kontrollen auch ange-mahnt wird.

Während die bisher beschriebenen Methoden insbesondere dazu geeignet waren, den Prozess zu strukturieren und die Suche nach geeigneten Lösungen einzuleiten, bietet sich gerade ein Wiki an, die Begriffe „zum Leben zu erwecken“. Wie bereits beschrieben, ist es heutz-

tage durchaus üblich, auf bereits erstellte Bausteine zur Lösung komplexer Probleme zurückzugreifen. Daher ist es nützlich, wenn im Laufe der Zeit für die verschiedenen Begriffe eine Sammlung von Lösungen – unterteilt nach den Technologien – entsteht, in der bewährte Lösungswege beschrieben werden. Wichtig ist, dass einem Begriff mit der Zeit entsprechende „Realisierungswege“ zugeordnet werden (siehe Abb. 3).

FAZIT

Die Anzahl der auf Steuerungstechnik beruhenden Systeme wird immer vielfältiger. Dennoch ist es gelungen, auf Basis von Bruners Fundamentalen Ideen neun Kompetenzen zu identifizieren, die in sehr vielen dieser Anwendungen – in unterschiedlichen Darstellungen – immer wieder auftauchen. Für die Transferfähigkeit der Lernenden wäre es daher eine Hilfe, wenn diesen Kompetenzen einheitliche Schlüsselbegriffe zugeordnet werden, die – zumindest innerhalb einer Schule – von allen Lehrenden unabhängig von deren Fachrichtung zur Beschreibung einer Aufgabe verwendet werden. Dadurch wird die Suche nach Lösungsansätzen in einer neuen Steuerungstechnik erheblich vereinfacht.

Literatur

- BRUNER, J. S. (1977): The process of education. Cambridge, Mass: Harvard University Press.
- DÖRGE, C. (2012): Informatische Schlüsselkompetenzen. Konzepte der Informationstechnologie im Sinne einer informatischen Allgemeinbildung; Online verfügbar unter: <http://oops.uni-oldenburg.de/1426/1/doiinf12.pdf>, zuletzt geprüft am 10.05.24.
- HARTMANN, W.; NÄF, M.; REICHERT, R. (2007): Informatikunterricht planen und durchführen. 1. Korr. Nachdruck. Berlin, Heidelberg: Springer.
- JOCHMANN, W. (2022): Future Skills im Maschinen- und Anlagenbau. Eine Analyse entlang des Produktlebenszyklus. Unter Mitarbeit von Kienbaum. Hg. v. VDMA. Köln/Frankfurt am Main. Online verfügbar unter; https://media.kienbaum.com/wp-content/uploads/sites/13/2022/05/VDMA_Kienbaum_Studie.pdf, zuletzt geprüft am 02.08.2022.
- KULTUSMINISTERKONFERENZ (2007): Handreichung für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe. Bonn (2007).
- SCHRATL, D. (2018): Fundamentale Ideen der Informatik im Kontext der Technischen Informatik. Diplomarbeit in der Studienrichtung Lehramtsstudium Informatik. Unter Mitarbeit von Betreuer: Ao. Univ.-Prof. Mag. Dr. KARL FUCHS; Online verfügbar unter <https://ulb-dok.uibk.ac.at/download/pdf/2600986.pdf>; zuletzt geprüft am 10.05.2024.
- SCHUBERT, S.; SCHWILL, A. (2011): Didaktik der Informatik. Der Begriff der fundamentalen Ideen in der Pädagogik. 2. Aufl. Heidelberg: Spektrum Akademischer Verlag.
- SCHWILL, A. (1993): Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 25 (1), S. 20–31. Online verfügbar unter: <http://www.informatikdidaktik.de/forschung/schriften/zdm.pdf>, zuletzt geprüft am 12.04.2024.
- SCHWILL, A. (1994): Fundamentale Ideen: Eine Studie zur Methodologie der Informatik. Online verfügbar unter <http://www.informatikdidaktik.de/Forschung/Schriften/Methodologie1994.pdf>, zuletzt geprüft am 20.06.2023.
- SCHWILL, A. (2001): Fundamentale Ideen der Informatik aus kognitionspsychologischer Sicht. Online verfügbar unter: <http://www.informatikdidaktik.de/Forschung/Schriften/Oldenburg.pdf>, zuletzt geprüft am 12.06.2023.
- STEEN, L. A. (Hg.) (2010): On the shoulders of giants. New approaches to numeracy. National Research Council. Washington D.C: National Academy Press.

Liebe Leserinnen und Leser,

die Zeitschrift „lernen & lehren“ möchte sehr gern vor allem den Fachleuten an den Lernorten die Möglichkeit einräumen, die vielfältigen Erfahrungen gut funktionierender Ausbildungs- und Unterrichtspraxis in Beiträgen der Zeitschrift zu veröffentlichen. Daher möchten wir Sie ermuntern, sich mit der Schriftleitung in Verbindung zu setzen. Wir streben wie bisher an, pro Heft zwei vom Themenschwerpunkt unabhängige Beiträge zu veröffentlichen.

Wenn Sie Interesse haben, an einem Themenschwerpunkt mitzuwirken, setzen Sie sich bitte rechtzeitig mit uns in Verbindung, da die Herstellung der Zeitschrift einen langen zeitlichen Vorlauf benötigt.

Wir freuen uns auf Ihre Rückmeldung!

Herausgeber und Schriftleitung

Hinweis auf Lern-/Lehrmaterial zum Themenschwerpunkt: Grundkurs Robotik am OSZ TIEM Berlin Spandau

AXEL GRIMM

NIKOLAUS STEFFEN

Ende Februar 2024 besuchten Nikolaus Steffen und Axel Grimm das Oberstufenzentrum TIEM (Technische Informatik, Industrieelektronik, Energiemanagement) in Berlin-Spandau. Im Rahmen der Gespräche wurde das Labor für „Mobile Autonome Roboter“ besichtigt und erkundet. Der gerade stattfindende Unterricht konnte problemlos durch uns besucht werden, da die Lernenden sehr zielgerichtet und eigenständig handlungsorientiert in Einzel- und Partnerarbeit an den Robotern lernten. Das in der gymnasialen Oberstufe verankerte Grundkursangebot „Mobile Autonome Roboter“ erweitert den technischen Zweig durch ein aktuelles Bildungsangebot, das mit vier Unterrichtsstunden pro Woche zweisprachig über alle drei Jahre den Schülerinnen und Schülern angeboten wird.

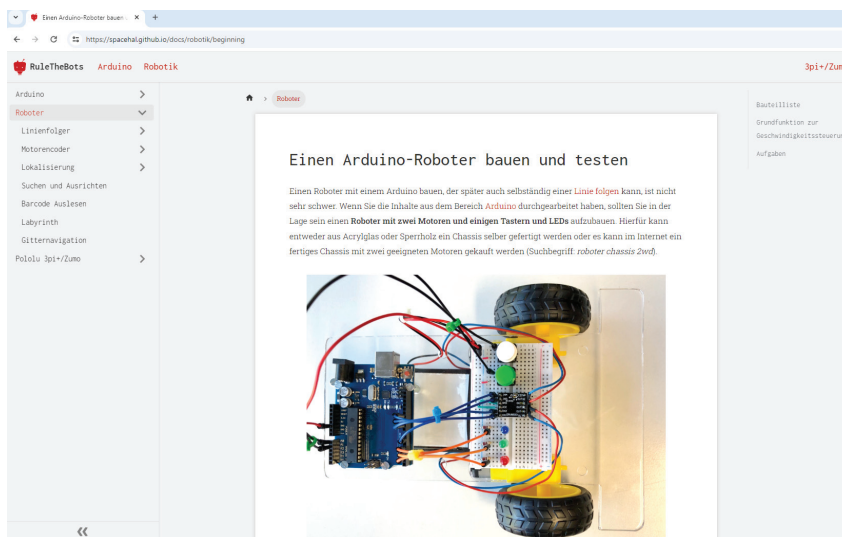


Abb. 1: Screenshot des Lernangebots zur Robotik

Nicht nur die begeisterten Lernenden, die beiden hochmotivierten Lehrkräfte und die sehr gute technische Ausstattung ermunterten zur hiesigen Berichterstattung; es war insbesondere die Art der Wissensdarbietung auf der Seite <https://spacehal.github.io/>, die uns dazu motivierte, diesen Hinweis hier publik zu machen.

Auf der o. g. Seite stehen den Lernenden – und nun auch interessierten Kolleginnen und Kollegen – zwei unabhängige, doch miteinander verbindbare Lernpfade zu Arduino- und Robotikprojekten als Open Educational Ressource (OER) zur Verfügung. Das hat Vorbildcharakter und soll zum Nachmachen ermuntern. Diese github-Ressource ist das extern begleitende Lehr-/Lernmaterial von schulspezifischen Kursangeboten des OSZ TIEM auf dem Berliner Landes-Moodle „Lernraum Berlin“.

Am praktischen Modell wird auf einfachem Weg am Grundprinzip eines Kantenfolgers (Abb. 2)

algorithmisches Denken veranschaulicht, umgesetzt und codiert. Für eine einfache Motorregelung zum Folgen einer Linie werden hier für den Roboter, bzw. den Messwert, drei verschiedene Fälle unterschieden:

Fall A: Der Sensor ist im Wesentlichen neben der Linie und detektiert viel IR-Licht (z. B. Messwert: 900). Der Roboter sollte mehr nach links zur Linie fahren.

Fall B: Der Sensor ist hauptsächlich über der Linie und detektiert wenig IR-Licht (z. B. Messwert: 180). Der Roboter sollte mehr nach rechts zur Linienkante fahren.

Fall C: Der Sensor befindet sich zur Hälfte über der Linie und ist somit ungefähr über der Kante (z. B. Messwert: 550). Der Roboter kann einfach geradeaus fahren.

Das Beispiel rechts verdeutlicht, wie ein Grundverständnis für eine einfache Umsetzung von autonomem Fahren entwickelt werden kann. Durch das Erkennen und die Formulierung der algorithmischen Strukturen zur Steuerung des Roboters sowie die weitere Planung und Verwendung von strukturierten

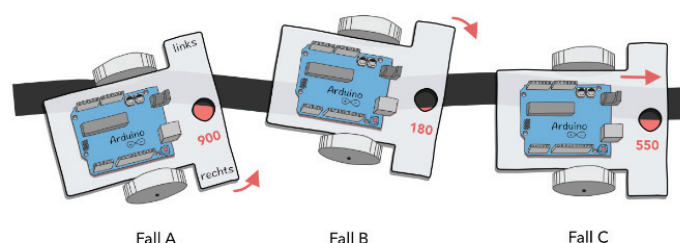


Abb. 2: Prinzip des Kantenfolgers

algorithmischen Sequenzen zur Lösung von Problemen (hier dann Fall C: das Fahrzeug kann weiter geradeaus fahren) wird algorithmisches Denken gefördert.

Wir möchten uns bei Stefan Warm und Detlev Herres recht herzlich für die vielen Eindrücke und den Einblick in deren Unterrichtspraxis bedanken. Wir hoffen darauf, dass OER-Ideen wie diese viele Nachahmende finden.

Ein Algorithmus zur Steuerung des Roboters (ohne Fall C) könnte folgendermaßen aussehen:

```
// Pseudocode für einen 2-Stufen-Kantenfolger
wertKante = 600 // Wert muss angepasst werden
wiederhole dauerhaft
    wertItr = messenItrSensor()
    wenn (wertItr > wertKante) // Fall A
        fahrenLinks()
    sonst wenn (wertItr <= wertKante) // Fall B
        fahrenRechts()
```

Usability in der Anwendungsentwicklung



CHRISTOPH DREEßEN



AXEL GRIMM

Programmieren ist in der Berufsarbeit nicht nur „Codieren“. Es werden Konventionen, Erfolgsfaktoren und Vorgaben berücksichtigt. Die Themen Usability und Barrierefreiheit sind aus verschiedenen Perspektiven heraus für das Programmieren sehr bedeutend und sollten daher bei der Softwareentwicklung in beruflichen Lernumgebungen mitgedacht werden. Durch deren Berücksichtigung lassen sich nicht nur nachhaltige wirtschaftliche Erfolge erzielen; gute Usability und Barrierefreiheit gestalten gesellschaftliche Verantwortung und sind damit Bildungsinhalt.

USABILITY - EIN BILDUNGSTHEMA

In der Lebens- und Arbeitswelt nahezu aller Personen nimmt der alltägliche Kontakt mit Computersystemen stetig zu. Der Ticketautomat am Bahnhof, der Online-Shop eines großen Versandhauses, das Online-Banking oder der tägliche Umgang mit verschiedenster Software im Berufsleben verdeutlichen die Mensch-Maschine-Interaktion, die sich manifestiert durch die Nutzung einer Benutzeroberfläche von Computersystemen. Somit wird auch beinahe jeder Mensch bereits Situationen erlebt haben, in welcher Probleme bei der Bedienung dieser Benutzeroberflächen auftraten: die Verwaltungssoftware bei der Arbeit gibt eine Fehlermeldung aus, die unverständlich ist und nicht erklärt, was falsch gemacht wurde, das passende Bahnticket am Ticketautomaten kann nicht intuitiv ausgewählt

werden oder beim Anmelden auf einer neuen Website werden die Bedingungen für das zu wählende Passwort erst angezeigt, nachdem alle Eingaben getätigt wurden und das Passwort nicht den Regeln entsprach.

Diese Probleme sind zurückzuführen auf eine schlechte Usability bzw. Benutzerfreundlichkeit der Benutzeroberflächen. Der Begriff Usability umfasst Aspekte wie die Anwenderfreundlichkeit, eine einfache und intuitive Bedienung sowie eine Fehlertoleranz (vgl. Florin 2015, S. 16) und ist ein Maß für die Qualität einer Software (vgl. SARODNICK & BRAU 2011, S. 19).

Für Anwendende ist eine benutzerfreundliche Software entscheidend, da sie eine intuitive Bedienung ermöglicht und somit die Effizienz und Zufriedenheit bei der Nutzung steigert. Usability ist daher aus mindestens drei Perspektiven von Bedeutung:

- Eine gut gestaltete Benutzeroberfläche und klare Funktionalitäten erleichtern es den Anwenderinnen und Anwendern, ihre Aufgaben schnell und ohne Probleme zu erledigen.
- Für Programmiererinnen und Programmierer ist Usability wichtig, da eine gut durchdachte Benutzeroberfläche und eine klare Strukturierung der Software die Entwicklung und Wartung erleichtern.
- Für Hersteller von Anwendungssoftware ist Usability ein entscheidender Wettbewerbsfaktor, da eine benutzerfreundliche Software dazu beitragen kann, Kundinnen und Kunden zu binden und neue zu gewinnen.

In der beruflichen Bildung werden in ganz unterschiedlichen Bildungsgängen informatische Inhalte unterrichtet und Programmieren gelernt. Im Grunde ließen sich Beispiele von der Berufsvorbereitung bis zur Fachschule hier anführen. Im Gegensatz zum Informatikunterricht an allgemeinbildenden Schulen sind die Ziele berufsbildender Bildungsgänge auch auf eine (spätere) Bewältigung von beruflichen Arbeitsaufgaben hin angelegt. Das Thema Usability kann daher als Bildungsinhalt angesehen werden, da es zur „nachhaltigen Mitgestaltung der Arbeitswelt und der Gesellschaft in sozialer, ökonomischer, ökologischer und individueller Verantwortung“ (KMK 2021, S. 14) beiträgt.

BEGRIFFSDEUTUNGEN ZUR USABILITY

In der deutschen Literatur zur Usability wird in der Regel der englische Fachbegriff verwendet. Eine einfache Übersetzung des Begriffes Usability ist aufgrund seiner komplexen Bedeutung in der Facharbeit nur bedingt möglich. Im Deutschen kann Usability am ehesten als „Gebrauchstauglichkeit“ gedeutet werden (vgl. SARODNICK & BRAU 2011, S. 19). Usability vereint jedoch auch mehrere Aspekte anderer deutschsprachiger Begrifflichkeiten wie „Anwenderfreundlichkeit“, „Einfache/Intuitive Bedienung“ oder „Fehlertoleranz“ (vgl. FLORIN 2015, S. 16).

Eine der frühesten Definitionen von Usability lieferte im Jahr 1984 EASON, der Usability als den Kontrast zwischen der Nützlichkeit eines Systems, der Nutzungsbereitschaft und Nutzungsfähigkeit der Anwendenden beschreibt (vgl. Eason nach Sarodnick & Brau 2011, S. 19). Die Nützlichkeit steht an dieser Stelle für den funktionalen Umfang einer Anwendung und die ihr immanenten technischen Möglichkeiten, bestimmte Aufgaben zu erfüllen. Die Nutzungsfähigkeit steht für die Fähigkeit der Anwendenden, die gegebenen technischen Möglichkeiten in der Praxis nutzen bzw. bedienen zu können. Die Nutzungsbereitschaft steht für die Akzeptanz der Anwendenden gegenüber der Software und dem Willen, das System zu nutzen. Eine Software kann also einen hohen Umfang an technischen Möglichkeiten bieten, wenn diese jedoch von den Nutzerinnen und Nutzern nicht bedient werden kann, sinkt die Akzeptanz für das Produkt und damit die Lust darauf, dieses auch zu tun. Die Usability einer Software ist

hierbei kein absoluter Zustand, sondern vielmehr ein Maß für die Qualität eines Produktes, die einer Varianz zwischen einer hohen und einer niedrigen Usability unterliegt (vgl. FLORIN 2015, S. 16). Usability sollte daher auch immer im zeitlichen Kontext der Entstehung eines Produktes betrachtet werden.

Eine weitere Beschreibung von Usability liefert NIELSEN. Er beschreibt Usability als die Frage danach, wie gut Anwenderinnen und Anwender die Funktionalität einer Software nutzen können (vgl. NIELSEN 1994, S. 25). Usability ist daher kein eindimensionaler Begriff, sondern umfasst eine Vielzahl an Attributen. Die von NIELSEN beschriebenen Kernattribute bieten einen guten Überblick darüber, welche Faktoren Einfluss auf eine gute bzw. schlechte Usability nehmen. Als Kernattribute der Usability macht Nielsen die Learnability (Erlernbarkeit), Efficiency (Effizienz), Memorability (Einprägsamkeit), Errors (Fehler) und Satisfaction (Befriedigung) aus (ebd., S. 26).

Unter Learnability wird verstanden, dass die Bedienung eines Systems einfach zu erlernen sein sollte, sodass Nutzerinnen und Nutzer schnell und intuitiv damit beginnen können, Aufgaben innerhalb eines Systems zu erledigen. Mit Efficiency wird verdeutlicht, dass das System effizient nutzbar sein sollte, sodass, nachdem Nutzerinnen und Nutzer die Bedienung des Systems erlernt haben, ein hohes Maß an Produktivität möglich ist. Memorability meint, dass die Bedienung des Systems leicht zu merken sein sollte. Nutzende, die das System nur gelegentlich nutzen, sollen auch nach einer gewissen Zeit der Nichtbenutzung des Systems wieder mit diesem schnell arbeiten können, ohne dass sie die Bedienung neu erlernen müssen. Ein System sollte eine niedrige Fehlerrate (Errors) besitzen, sodass Nutzende bei der Bedienung wenig Fehler begehen. Wenn Nutzerinnen und Nutzer Fehler begehen, so muss es ihnen auf einfache Weise möglich sein, die Fehler rückgängig zu machen. Systembedingte Laufzeitfehler dürfen nicht auftreten. Unter Satisfaction wird verstanden, dass ein System angenehm zu benutzen sein sollte, sodass Personen bei dessen Nutzung eine subjektive Zufriedenheit durch die Nutzung des Systems erfahren (vgl. Nielsen 1994).

Der Begriff Usability ist auch ein Gegenstandsbereich der Normung. Eine dieser Normen zur Usability ist die im Jahr 2018 veröffentlichte Norm „DIN EN ISO 9241-11:2018-11: Ergonomie der Mensch-System-Interaktion“. Hier lassen sich sowohl Ansätze von EASON als auch die Kernaspekte der Usability nach NIELSEN wiedererkennen. In dieser Norm wird Usability beschrieben als ein „Ausmaß, in dem ein System, ein Produkt oder eine Dienstleistung durch bestimmte [Benutzerinnen und] Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellen zu erreichen“ (DIN EN ISO 9241-11:201811).

ZUR USABILITY ALS HERAUSFORDERUNG DER SOFTWAREENTWICKLUNG

Problemfelder

Welche Ziele User Interfaces (UI)-Entwicklerinnen und -Entwickler bezüglich der Usability eines Produktes verfolgen, erscheint nach den bisherigen Ausführungen zum Maß der Qualität einer Software trivial; Ziel der Entwicklung eines Softwareproduktes sollte stets das Schaffen einer Benutzeroberfläche sein, die eine sehr gute Usability besitzt. Zur Erzielung eines guten Ergebnisses reicht es jedoch nicht aus, basierend auf eigenen Erfahrungen mit schlechter Usability bestimmte Dinge zu vermeiden (vgl. FLORIN 2015, S. 16). Die reine Erfahrung in der Bedienung von Systemen ist nicht ausreichend, um Kenntnisse über eine gute Benutzeroberfläche zu haben (vgl. MYERS 1993, S. 3). Damit Softwareentwicklerinnen und -entwickler eine Software mit einer guten Usability programmieren können, ist es notwendig, im Entwicklungsprozess gezielt auf das Thema Usability zu achten und bestimmte Prinzipien des Usability Engineerings anzuwenden.

User und deren Aufgaben kennen

Die Analyse der Absichten der Anwendenden eines Produktes stellt den ersten Schritt in Usability-Prozessen dar, da die unterschiedlichen Charakteristika der User, sowie die Vielfalt der von diesen zu erledigenden Aufgaben, zwei der bedeutendsten Faktoren sind, die einen Einfluss auf die Usability einer Software nehmen (vgl. NIELSEN 1994, S. 73). Die Anforderungen an die Usability sind auch abhängig vom Einsatzgebiet der verschiedenen technischen Systeme. Je nach Einsatzgebiet einer Software, können die spezifischen Anforderungen an ein Produkt variieren, so kann in bestimmten Bereichen ein ansprechendes Design wichtig sein, in anderen Bereichen sind jedoch die Zuverlässigkeit oder Mobilität des Systems von höherer Bedeutung (vgl. SARODNICK & BRAU 2011, S. 29).

Die Zielgruppe einer Software kann je nach Nutzungskontext eines Produktes ebenfalls stark variieren. Das hat Auswirkungen darauf, wie Benutzeroberflächen gestaltet werden sollten. In bestimmten Fällen beschränkt sich die Zielgruppe auf Angestellte einer bestimmten Abteilung innerhalb eines Unternehmens und fällt somit eher klein aus (vgl. NIELSEN 1994, S. 74). Ein Produkt kann sich jedoch auch an eine größere Zielgruppe, möglicherweise auch an die gesamte Bevölkerung oder zumindest eine große Teilgruppe dieser richten (ebd.). Unter anderem sind der Bildungsgrad, das Alter, die Arbeitserfahrung und die Erfahrung mit Computersystemen der Zielgruppe Faktoren, die Einfluss darauf nehmen, wie komplex eine Benutzeroberfläche gestaltet werden darf (vgl. ebd.). Für Programmierende ist es notwendig, sich beim Entwerfen von Nutzeroberflächen ihrer Zielgruppe bewusst zu sein. Einer der häufig genannten Leitsprüche zu diesem Thema ist „You are not the user“ (dt. „Du bist nicht die Nutzerin/der Nutzer“).

Programmierende sind Spezialistinnen und Spezialisten in ihrem Bereich und können sich somit nicht ohne weiteres in die Lage eines Anwendenden aus unterschiedlichsten Zielgruppen hineinversetzen, die ja dann dieser Technologie zumeist zum ersten Mal gegenüberstehen (vgl. RICHTER & FLÜCKIGER 2016, S. 3). Zudem sind diese Programmiererinnen und Programmierer oftmals Laien in Bezug auf das Anwendungsfeld einer Software und daher nicht umfassend mit der Terminologie und den Abläufen im Anwendungsgebiet vertraut (vgl. ebd.).

Einbezug in den Entwicklungsprozess

Bei der Entwicklung einer neuen Software ist die Systementwicklung einer der wesentlichen Bestandteile. Die Systementwicklung bezeichnet die technische Umsetzung der zu realisierenden Software. Für die Systementwicklung wurden verschiedene Prozesse modelliert; ebenfalls wurden Prozesse im Bereich des Usability Engineerings entwickelt (vgl. SARODNICK & BRAU 2011, S. 103). Diese parallel entwickelten Prozesse bieten nach Meinung von SARODNICK und BRAU ein gewisses Konfliktpotenzial. Prozesse zur Sicherung einer guten Usability und die Prozesse bezüglich der technischen Entwicklung bedingen einander und sollten nicht als zwei voneinander getrennte Prozesse wahrgenommen werden (ebd. 2011, S. 103).

Ergebnisse aus der Usability können bereits die Funktionalität der Software maßgeblich beeinflussen und neue technische Anforderungen darstellen, die wiederum in der Systementwicklung aufgegriffen werden müssen (vgl. SARODNICK & BRAU 2011, S. 103). Im Idealfall sollten daher die Prozesse des Usability-Engineerings und des Software-Engineerings als gemeinsamer Prozess in der Entwicklung angesehen werden. Es ist somit notwendig, dass zur Erzielung einer guten Usability, Maßnahmen zur Sicherung dieser implementiert und Aspekte der Benutzeroberfläche bezüglich ihrer Usability bereits früh in der Entwicklung berücksichtigt werden.

Als Organisationsstruktur innerhalb der Softwareentwicklung setzen sich agile Prozesse durch. In agilen oder auch inkrementellen Entwicklungen werden einzelne Komponenten des Produktes sukzessiv entworfen, entwickelt, getestet und abschließend in das Gesamtsystem integriert (vgl. SARODNICK & BRAU, S. 105). Kurze Entwicklungsphasen, durch welche bereits früh in der Entwicklung testbare Teilprodukte entstehen, ermöglichen es, schnelles Feedback zum Produkt zu erhalten (vgl. RICHTER & FLÜCKIGER 2016, S. 23). Durch die kurzen Iterationen in agilen Entwicklungsprozessen findet durch jede dieser Iterationen ein Entwicklungs- und Lernprozess statt (vgl. ebd.). Durch das Feedback innerhalb der Zyklen und der Reflexion der Ergebnisse kann ein agiles Entwicklungsteam mehr über das zu entwickelnde Produkt lernen und ein besseres Problemverständnis für das Produkt entwickeln (vgl. ebd.).

Eines der aktuell erfolgreichsten Modelle im Bereich der agilen Entwicklung ist Scrum. In einem Scrum-

Prozess (siehe Abb. 1) werden Einträge der zu erledigenden Arbeit im sogenannten Backlog festgehalten. Die priorisierten Einträge aus diesem Backlog werden dann in sogenannten Produktinkrementen zusammengeführt und innerhalb eines Sprints abgearbeitet. Innerhalb des Sprints finden tägliche Meetings statt, die dem Feedback dienen. Jede Iteration stellt hierbei einen Feedbackzyklus dar.

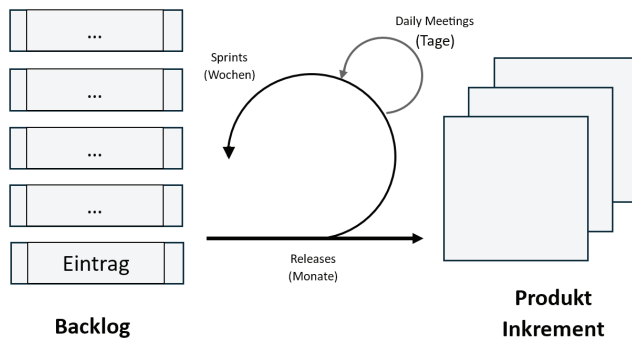


Abb. 1: Scrum Prozess (nach Richter & Flückiger 2016, S. 24)

Die Prinzipien der agilen Entwicklung zeichnen sich dadurch aus, dass frühes Feedback bereit durch eine echte Nutzung des Systems erzielt werden kann, da stets Komponenten der Software funktional implementiert werden und somit testbar sind. Diese iterative Herangehensweise ermöglicht eine stetige Überprüfung mit Kundinnen und Kunden sowie Nutzerinnen und Nutzern (vgl. RICHTER & FLÜCKIGER 2016, S. 27). Somit können derartige iterative Prozesse für eine Erzielung einer guten Usability genutzt werden.

Fehlende Expertise in Unternehmen

Es ist zu beobachten, dass insbesondere größere Unternehmen vermehrt einen Fokus auf die Einführung von Usability-Engineering-Methoden legen. In einer Studie von GULLIKSEN et al. konnte festgestellt werden, dass 70 Prozent der befragten Usability-Expertinnen und -Experten in Unternehmen beschäftigt waren, die mindestens 50 Mitarbeitende beschäftigen (GULLIKSEN et al. nach BRUUN 2010, S. 82). In einer weiteren Studie von ROSENBAUM wurde festgestellt, dass die meisten Usability-Expertinnen und -Experten in Unternehmen mit einer Größe von 250 Mitarbeitenden und mehr beschäftigt sind (ROSENBAUM nach BRUUN 2010, S. 82).

BRUUN schlägt in seiner Arbeit vor, dass Fort- und Weiterbildung – von Nicht-Usability-Expertinnen und -Experten im Bereich der Beschäftigten in der Softwareentwicklung – im Umgang mit Usability-Engineering-Methoden potenziell die Herausforderungen in der Entwicklung reduzieren können. In einer Studie von METZKER und OFFERGELD aus dem Jahr 2001 konnte festgestellt werden, dass die Fortbildung von Software-Entwicklerinnen und -Entwicklern in grundlegenden Usability-Engineering-Methoden dazu führte, dass diese im Anschluss Softwarekomponenten mit einer besseren Usability als zuvor entwickelten (METZKER & OFFERGELD nach BRUUN 2010, S. 82).

Zum Wert von Usability in der Softwareentwicklung

Eine systematische Verbesserung der Usability eines Produktes kann entscheidend dazu beitragen, dass die Bindung von Kundinnen und Kunden an einen Anbieter erhöht wird (vgl. GRÜNWIED 2017, S. 57). Eine gute Usability kann sich darüber hinaus positiv auf die Akquise weiterer Kundinnen und Kunden auswirken, da eine gute Usability die Nutzenden dazu anregt, das genutzte Produkt weiterzuempfehlen (vgl. ebd., S. 58). Erhöhte Entwicklungskosten durch zusätzliche Usability-Engineering-Maßnahmen können so langfristig positive finanzielle Effekte bewirken. Eine mangelnde Usability kann sich hingegen messbar negativ auf die Betriebskosten eines Unternehmens, bspw. eines Onlinehandels, auswirken. Wenn die Kundinnen und Kunden durch eine schlechte Usability häufiger in Situationen kommen, in denen sie Unterstützung bei der Bedienung des Systems benötigen, werden Kosten durch ein erhöhtes Aufkommen an Anrufen in Call-Centern des Unternehmens oder Anfragen per Mail im Kundencenter anfallen (vgl. EBERHARD-YOM 2010, S. 12).

Abseits der positiven Auswirkungen einer guten Usability aus wirtschaftlicher Perspektive beschreiben NIELSEN und LORANGER weitere Potenziale guter Usability für die Gesellschaft. So kann älteren Personen und Menschen mit Beeinträchtigungen durch die Verbesserung der Usability von Systemen eine einfachere Teilhabe in der digitalen Welt sowie ein einfacherer Zugang zu Informationen und Dienstleistung ermöglicht werden (vgl. NIELSEN & LORANGER 2006, S. XXII). In der Bewertung von NIELSEN und LORANGER wird insbesondere der Aspekt der Barrierefreiheit als Teil der Usability positiv gewertet.

BARRIEREFREIHEIT UND USABILITY

Barrierefreiheit kann als ein Teilgebiet der Usability angesehen werden. Usability bezieht sich auf die Gesamtheit der Benutzererfahrung, einschließlich der Benutzerfreundlichkeit, Effizienz und Zufriedenheit, während Barrierefreiheit spezifisch darauf abzielt, die Zugänglichkeit für Menschen mit besonderen Anforderungen sicherzustellen. Barrierefreiheit wird somit zu einem wichtigen Bestandteil der Usability, da eine Software nur dann als wirklich benutzerfreundlich betrachtet werden kann, wenn sie für alle zugänglich ist. Wenn eine Software für bestimmte Benutzergruppen aufgrund von Barrieren nicht zugänglich ist, beeinträchtigt dies die Gesamterfahrung und somit auch die Usability.

Bisher waren Bundesbehörden, Landesbehörden und Kommunalverwaltungen verpflichtet, sicherzustellen, dass ihre digitalen Dienste barrierefrei und für Menschen mit besonderen Anforderungen zugänglich sind. Das Ziel des Barrierefreiheitsstärkungsgesetzes (BFSG) ist es, die Teilhabe aller Bürgerinnen und Bürger an den digitalen Diensten der öffentlichen Verwaltung zu gewährleisten und die Diskriminierung von Menschen mit Behinderungen zu verhindern. Durch das neue Bar-

rierefreiheitsstärkungsgesetz wird das Thema Barrierefreiheit weiter an Bedeutung in der Facharbeit der Softwareentwicklung gewinnen. Das BFSG wurde im Juli 2021 beschlossen, setzt den European Accessibility Act (EAA) im nationalen Recht der Bundesrepublik um und tritt am 25. Juni 2025 in Kraft (vgl. Informationstechnikzentrum Bund). Im Unterschied zu vorherigen Verordnungen zur Barrierefreiheit, welche lediglich öffentliche Einrichtungen zur barrierefreien Gestaltung ihrer Websites verpflichteten, werden diese Verpflichtungen jetzt auch auf private Wirtschaftsakteure erweitert.

Das BFSG unterscheidet zwischen Produkten und Dienstleistungen. Hierbei umfasst die Kategorie der Produkte Hardwaresysteme sowie deren Betriebssysteme, Selbstbedienungsterminals sowie Endgeräte für Verbraucherinnen und Verbraucher, die für den Zugang zu Telekommunikationsdiensten oder den Zugang zu audiovisuellen Mediendiensten verwendet werden (vgl. Informationstechnikzentrum Bund). Unter die Kategorie der Dienstleistungen fallen Telekommunikationsdienste, Bankdienstleistungen, Bestandteile von Personenbeförderungsdiensten und deren Apps sowie elektronische Ticketsysteme, E-Books und für diese bestimmte Software sowie Dienstleistungen im elektronischen Geschäftsverkehr (vgl. ebd.).

USABILITY TESTS

Usability wurde hier als eine messbare Eigenschaft einer Software eingeführt. Ein Bereich des Usability-Engineerings, das sogenannte Usability-Testing, befasst sich mit eben jener Messung der Usability von Softwareprodukten. Durch Usability Tests kann geprüft werden, wie Anwendende tatsächlich mit einer Applikation umgehen. Diese Tests stellen somit eine der wichtigsten Methoden im Entwicklungsprozess dar (vgl. JACOBSEN & MEYER 2022, S. 209). Bei Usability-Tests kann zwischen formativen und summativen Evaluationen differenziert werden. Die formative Evaluation stellt eine Analyse zur Verbesserung der Qualität eines Systems dar und die summative Evaluation fungiert als eine zusammenfassende Prüfung zur Qualitätskontrolle (vgl. RICHTER & FLÜCKIGER 2016, 103 f.; NIELSEN 1994, S. 170).

Zur Evaluierung der Usability existieren verschiedene Verfahren. Moderierte Tests finden mit Nutzenden in Laboren statt; unmoderierte Tests können auch online durchgeführt werden (vgl. TULLIS & ALBERT 2013, S. 52 ff.). Für die klassischen Usability Tests, bei denen Nutzende bei der Nutzung eines Produktes beobachtet werden, müssen zunächst Aufgaben erstellt werden, die von Testpersonen mit der zu prüfenden Applikation durchgeführt werden sollen (vgl. RICHTER & FLÜCKIGER 2016, S. 104). Wichtig hierbei ist eine Erarbeitung von Aufgaben, die für die Nutzenden realistisch und relevant sind.

Neben qualitativen Daten, wie dem Feedback der Testnutzenden, können im Usability-Testing auch quantitative Daten, sogenannte Usability Metriken, gesammelt werden (vgl. JACOBSEN & MEYER 2022, S. 217). Beispiele

für Metriken sind die Task-Completion Rate (TCR, dt. Anteil erfolgreich abgeschlossener Aufgaben), Time-On-Task (TOT, dt. Zeit pro Aufgabe) und die durchschnittliche subjektive Zufriedenheit der Nutzenden (vgl. ebd.).

Spielen Zeit und Kosten einen limitierenden Faktor in der Entwicklung eines Produktes, so können sogenannte Guerilla-Tests eine Alternative zu konventionellen Usability-Tests sein. Guerilla-Tests ignorieren bestimmte Regeln, die für andere Usability-Tests gelten, z. B., dass die Testnutzenden aus der Zielgruppe der später Nutzenden des Produktes akquiriert werden müssen (vgl. JACOBSEN & MEYER 2022, S. 241). Im Rahmen von Guerilla-Tests können Kolleginnen und Kollegen, Freunde oder die Familie als Testpersonen eingesetzt werden (ebd., S. 243 f.). Guerilla-Tests können somit schneller durchgeführt werden, da Testprobandinnen und -probanden innerhalb des Unternehmens oder der Entwicklung akquiriert werden können.

USABILITY IM UNTERRICHT

Für Lernende in Bildungsgängen und Ausbildungsberufen mit informatisch-informationstechnischen Bezügen sind die Themen Usability und Barrierefreiheit dann von hervorzuhebender Bedeutung, wenn Kompetenzen erlangt werden sollen, die es ermöglichen, technische Systeme mitzugestalten. Erst durch eine Sensibilisierung für dieses Themenfeld und dessen Durchdringung kann im Sinne des Bildungsauftrages Verantwortung übernommen werden. Sich dieser Verantwortung bewusst zu sein, kann als ein Ziel unterrichtlichen Handelns ausgemacht werden.

Schlechte Erfahrungen mit der Usability haben vermutlich alle Lernenden bei der Verwendung von Softwareprodukten oder technischen Geräten bereits gemacht. Aufbauend darauf können die Lernenden im Unterricht über negative Erfahrungen in der Bedienung von elektronischen Geräten berichten. Hierdurch wird ein Arbeits- und Lebensweltbezug für die Lernenden geschaffen. Software oder Geräte können dann exemplarisch im Unterricht analysiert und Usability-Probleme thematisiert werden. Durch einen theoriegestützten Maßnahmenkatalog (bspw. angelehnt an NIELSEN 1994), der Verbesserungsvorschläge für das User-Interface des untersuchten Produktes enthält, entstehen eine erste Vertiefung und ein Anknüpfungspunkt für Ideen zur Weiterentwicklung des Produktes im Unterricht.

Standardisierte Usability Tests sind aufgrund ihrer Komplexität eher ungeeignet für den Einsatz im Unterricht; die Verwendung von Guerilla-Tests, also nicht-standardisierten Tests, bietet sich für die unterrichtliche Einbindung eher an. Lernende sollten in ihren berufsschulischen Projekten zur Anwendungsentwicklung dazu angehalten werden, die entwickelten Benutzeroberflächen von ihren Mitlernenden testen zu lassen. Wenn eine Aufteilung in Projektgruppen stattfindet, so sollten auch gruppenübergreifend Testpersonen einbezo-

gen werden. Testpersonen aus anderen Projektgruppen haben eine größere Distanz zum entwickelten Produkt und können so andere Einblicke ermöglichen.

Der konsequente Einbezug der Thematiken Usability und Barrierefreiheit beim Programmierenlernen hat einen bedeutenden Bildungswert, der bisher sowohl in der realen Facharbeit als auch in der berufsschulischen Praxis noch nicht denjenigen Stellenwert erlangt hat, der nötig wäre, um zukünftige Mitgestaltende der Lebens- und Arbeitswelt bezüglich ihrer Verantwortung dafür zu sensibilisieren.

Literatur

BRUUN, A. (2010): Training software developers in usability engineering. In: NordiCHI '10: 6th Nordic Conference on Human-Computer Interaction. New York, pp. 82–91. <https://doi.org/10.1145/1868914.1868928>

DIN EN ISO 9241-11:2018-11, Ergonomie der Mensch-System-Interaktion- Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte.

EBERHARD-YOM, M. (2010): Usability als Erfolgsfaktor. + Grundregeln + User Centered Design + Umsetzung. Berlin: Cornelsen.

FLORIN, A. (2015): User - Interface - Design. Usability in Web- und Software Projekten. Norderstedt: Books on Demand.

GRÜNWIED, G. (2017): Usability von Produkten und Anleitungen im digitalen Zeitalter. Handbuch für Entwickler,

IT-Spezialisten und Technische Redakteure. Berlin: Publicis.

JACOBSEN, J.; MEYER, L. (2022): Praxisbuch Usability und UX. Was alle wissen sollten, die Websites und Apps entwickeln. 3. Aufl. Bonn: Rheinwerk Verlag.

KMK (2021): Handreichung für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe. Berlin.

MYERS, B. A. (1993): Why are Human-Computer Interfaces Difficult to Design and Implement? Pittsburgh: Carnegie Mellon University, Computer Science Department.

NIELSEN, J. (1994): Usability engineering. San Francisco: Morgan Kaufmann.

NIELSEN, J.; LORANGER, H. (2006): Web Usability. München: Addison-Wesley Verlag.

RICHTER, M.; FLÜCKIGER, M. (2016): Usability und UX kompakt. Produkte für Menschen. 4. Aufl. Berlin, Heidelberg: Springer Vieweg.

RUBIN, J.; CHISNELL, D. (2008): Handbook of usability testing. How to plan, design, and conduct effective tests. 2nd Ed. Indianapolis: John Wiley & Sons.

SARODNICK, F.; BRAU, H. (2011): Methoden der Usability Evaluation. 2. Aufl. Bern: Hogrefe AG.

TULLIS, T.; ALBERT, B. (2013): Measuring the User Experience. Collecting, Analyzing, and Presenting Usability Metrics. 2nd Ed. Amsterdam: Morgan Kaufmann.

Verzeichnis der Autorenschaft

DREEßEN, CHRISTOPH

Absolvent des Studienganges „Master of Vocational Education“ am BIAT der Europa-Universität Flensburg in der beruflichen Fachrichtung Informationstechnik, christoph.dreessen@googlemail.com

GRIMM, AXEL

Prof. Dr., Hochschullehrer, Berufsbildungsinstitut Arbeit und Technik, Europa-Universität Flensburg, axel.grimm@biat.uni-flensburg.de

JACOBS, SVEN

Universität Siegen, Didaktik der Informatik, sven.jacobs@uni-siegen.de

JÄCKEL, GÖTZ

Trainer für Automatisierungssoftware, Goetz.Jaeckel@GJ-Didaktik.de

JASCHKE, STEFFEN

Dr., Akademischer Rat, Universität Siegen, Didaktik der Informatik, steffen.jaschke@uni-siegen.de

HARDEBUSCH, TIMO

Universität Siegen, Didaktik der Informatik, timo.hardebusch@student.uni-siegen.de

ANDREAS LINDNER

StD. Dipl. - Berufspäd. Univ., Fachbetreuer der Fachstufe Industriemechanik, Städt. Berufsschule für Fertigungstechnik München, andreas.lindner@bsz-deroy.muenchen.musin.de

SCHLÜTER, JESSE

Trainer für Automatisierungstechnik bei Michael Dziallas Engineering, jesse.schlueter@mde-online.de

STECHERT, PEER

Dr., Landesfachberater und Studienleiter für Informatik am IQSH, Informatik-Herausgeber des MNU-Journal und langjähriger IHK-Prüfer im Ausbildungsberuf IT-Fachinformatiker/in, peer.stechert@iqsh.de

STEFFEN, NIKOLAUS

Dr., OStR, Referent im Landesprogramm „Zukunft Schule im digitalen Zeitalter“ in Schleswig-Holstein für gewerblich-technische Berufsbildung, Berufsbildungsinstitut Arbeit und Technik, Europa-Universität Flensburg, nsteffen@uni-flensburg.de

Impressum

„lernen & lehren“ erscheint in Zusammenarbeit mit den Bundesarbeitsgemeinschaften für Berufsbildung in den Fachrichtungen Elektro-, Informations-, Metall- und Fahrzeugtechnik e. V.

www.lernenundlehren.de

Herausgeber

Axel Grimm (Flensburg), Volkmar Herkner (Flensburg), Georg Spöttl (Bremen), Michael Tärre (Hannover)

Beirat

Matthias Becker (Hannover), Thomas Berben (Hamburg), Ralph Dreher (Siegen), Peter Hoffmann (Lauingen), Claudia Kalisch (Rostock), Andreas Lindner (München), Tamara Riehle (Rostock), Reiner Schlausch (Flensburg), Friedhelm Schütte (Berlin), Ulrich Schwenger (Heidelberg), Nikolaus Steffen (Flensburg), Thomas Vollmer (Kassel), Lars Windelband (Karlsruhe), Sören Schütt-Sayed (Hamburg)

Heftbetreuende: Nikolaus Steffen/Axel Grimm

Titelbild: BAG-Archiv

Schriftleitung (V. i. S. d. P.) lernen & lehren

Prof. Dr. Torben Karges, Europa-Universität Flensburg, Berufsbildungsinstitut Arbeit und Technik, Auf dem Campus 1, 24943 Flensburg, torben.karges@uni-flensburg.de

OStR Dr. Tim Richter-Honsbrok, Leibniz Universität Hannover, Institut für Berufswissenschaften der Metalltechnik, Appelstraße 9, 30167 Hannover, richter@ibm.uni-hannover.de

Alle schriftlichen Beiträge und Leserbriefe bitte an eine der obenstehenden Adressen senden. Manuskripte gelten erst nach Bestätigung der Schriftleitung als angenommen. Namentlich gezeichnete Beiträge stellen nicht unbedingt die Meinung der Herausgeber dar. Theorie-Beiträge des Schwerpunktes werden einem Review-Verfahren ausgesetzt. Unverlangt eingesandte Rezensionsexemplare werden nicht zurückgesandt.

Layout/Gestaltung

Brigitte Schweckendieck/Winnie Mahrin

Verlag, Vertrieb und Gesamtherstellung

Roco Druck GmbH, Neuer Weg 48a, 38302 Wolfenbüttel, Telefon: (0 53 31) 97 01-0

Als Mitglied einer BAG wenden Sie sich bei Vertriebsfragen (z. B. Adressänderungen) bitte stets an die Geschäftsstelle, alle anderen wenden sich bitte direkt an den Verlag.

Geschäftsstelle der BAG Elektro-, Informations-, Metall- und Fahrzeugtechnik
c/o ITB – Institut Technik und Bildung der Universität Bremen, Am Fallturm 1 – 28359 Bremen
kontakt@bag-elektrometall.de

ISSN 0940-7340

ADRESSAUFKLEBER

BAG

WWW.BAG-ELEKTROMETALL.DE
KONTAKT@BAG-ELEKTROMETALL.DE