

Was jeder über Informatik lernen sollte – Eine Analyse von Hochschulkursen für Studierende anderer Fachrichtungen

Stefan Seegerer und Ralf Romeike

Friedrich-Alexander-Universität

Didaktik der Informatik

Martensstr. 3

91058 Erlangen

stefan.seegerer@fau.de

ralf.romeike@fau.de

Abstract: Um für ein Leben in der digitalen Gesellschaft vorbereitet zu sein, braucht jeder heute in verschiedenen Situationen umfangreiche informatische Grundlagen. Die Bedeutung von Informatik nimmt nicht nur in immer mehr Bereichen unseres täglichen Lebens zu, sondern auch in immer mehr Ausbildungsrichtungen. Um junge Menschen auf ihr zukünftiges Leben und/oder ihre zukünftige berufliche Tätigkeit vorzubereiten, bieten verschiedene Hochschulen Informatikmodule für Studierende anderer Fachrichtungen an. Die Materialien jener Kurse bilden einen umfangreichen Datenpool, um die für Studierende anderer Fächer bedeutenden Aspekte der Informatik mithilfe eines empirischen Ansatzes zu identifizieren. Im Folgenden werden 70 Module zu informatischer Bildung für Studierende anderer Fachrichtungen analysiert. Die Materialien – Publikationen, Syllabi und Studentafeln – werden zunächst mit einer qualitativen Inhaltsanalyse nach Mayring untersucht und anschließend quantitativ ausgewertet. Basierend auf der Analyse werden Ziele, zentrale Themen und Typen eingesetzter Werkzeuge identifiziert.

Keywords: Informatik für alle, Hochschulkurse, Andere Fachrichtungen, Inhaltsanalyse.

1 Einleitung

Um für ein Leben in der digitalen Gesellschaft vorbereitet zu sein, braucht *jeder* heute in verschiedenen Situationen umfangreiche informatische Grundlagen. In dieser Hinsicht wird informatische Bildung als wichtiger Bestandteil der Allgemeinbildung betrachtet, der bisher vor allem im Schulkontext diskutiert und ausdifferenziert (z. B. [Gel16]) wurde. Auch an Universitäten steigt das Interesse an informatischer Bildung für Studierende aller Fachrichtungen. Mit steigender Nachfrage kommen unweigerlich Fragen über die Gestaltung entsprechender Kursangebote auf (vgl. [Ba10]). Ein „Informatikcurriculum für alle“ müsste eine Basis enthalten, mit der jeder Studierende vertraut sein sollte. Bei der Gestaltung eines solchen Curriculums stellt sich u. a. die Frage, welche Ziele mit einem solchen Bildungsangebot verfolgt werden sollen und welche Themen dabei zentral sind. Diese Frage ist eng verknüpft damit, was *jeder* gebildete Mensch über Informatik wissen sollte. Dies kann zum einen normativ durch Experten festgelegt und argumentativ untermauert werden, andererseits aber auch durch eine empirische Erhebung der Aspekte, die bereits existierende Angebote als wichtig erachten, ermittelt werden.

Viele Hochschulen bieten Informatikveranstaltungen für Studierende anderer Fachrichtungen an. Die Materialien jener Kurse bilden einen umfangreichen Datenpool für einen entsprechenden empirischen Ansatz, der die für Studierende anderer Fächer bedeutenden Aspekte der Informatik identifizieren soll. Aus diesem Grund analysieren wir eine Auswahl solcher Kurse, die zwischen 2001 und 2018 angeboten wurden. Diese Datenbasis erlaubt es, die Gemeinsamkeiten verschiedener Ansätze informatischer Bildung zu erfassen. Im Mittelpunkt der Untersuchung stehen formulierte Ziele, thematisierte Inhalte und die Typen verwendeter Programmierwerkzeuge. Abschnitt 2 beschreibt zunächst Hintergründe und Argumentationslinien im Kontext „Informatik für alle“. Daran anknüpfend werden in den Kapiteln 3 und 4 Forschungsfragen, Methodik und die zugrundeliegende Datenbasis dargestellt. In den Kapiteln 5 und 6 werden die Ergebnisse der Untersuchung präsentiert und diskutiert.

2 Hintergrund

Immer mehr Akteure betrachten Informatikkompetenz als zentralen Bestandteil der Allgemeinbildung. Es sind nicht mehr nur Vertreter des Fachs, die dessen Wichtigkeit betonen, sondern auch das Wirtschaftsministerium [Bu16] oder Fachverbände anderer Disziplinen wie der Chemie [BV18] sprechen sich für „Informatik für alle“ aus.

In öffentlichen Diskussionen über die Bedeutung informatischer Allgemeinbildung wird häufig auf die Anforderungen zukünftiger Arbeit verwiesen. Aber im Zusammenhang, warum Informatik für alle wichtig ist, werden noch weitere Argumente aufgeführt. Vogel et al. [VSC17] oder Döbeli Honegger [Dö16] stellen verschiedene Argumente für informatische Bildung heraus. Döbeli Honegger nennt dabei beispielsweise das Konzeptwissenargument, mit Informatik ließen sich digitale Werkzeuge besser nutzen, oder das Welterklärungs- bzw. Mündigkeitsargument, Informatik helfe die technische Welt zu verstehen bzw. mitzugestalten. Häufig wird auch problemlösendes Denken oder Computational Thinking als Argument für informatische Bildung genannt. Letzteres ist ein Begriff der, von Wing [Wi06] popularisiert, die Art und Weise charakterisiert, wie ein Informatiker Probleme löst. Ziel informatischer Bildung in diesem Kontext ist es also vor allem Denkweisen und Problemlösestrategien zu vermitteln. Dazu zählen das Zerlegen von Problemen in kleinere Teilprobleme (Dekomposition) oder das Weglassen von Details (Abstraktion). Solche informatischen Problemlösefähigkeiten bzw. Denkweisen werden an vielen Stellen als fundamental für die Allgemeinbildung betrachtet (z. B. [Di11]). Der Einfluss Wings auf Universitäts- und Collegenkurse zeigt sich beispielsweise im Framework *Advanced Placement Computer Science Principles*, das, erstmals in 2010 veröffentlicht, die Inhalte und Ziele von Informatikkursen auf College Niveau an Highschools umreißt. Das Framework besteht aus sieben großen Ideen der Informatik und sechs Computational-Thinking-Praktiken.

Die den Argumenten zugrundeliegenden Sichtweisen können Zieldimensionen informatischer Allgemeinbildung an Hochschulen darstellen. Sie geben allerdings nur bedingt Aufschluss über mögliche Themen, die in solchen Kursen behandelt werden sollten. Mit den „Great Principles of Computing“ [De03], den „Fundamentalen Ideen der Informatik“ [Sc93] oder auch den „Big Ideas of K-12 Computing“ [BTY18] existieren Kataloge, die wichtige und zeitlose Aspekte der Informatik erfassen. Während Denning und Schwills Ansätze vor allem aus dem Fach heraus motiviert sind, bezieht sich der Ansatz von Bell konkret auf Schulbildung und nennt zehn wesentlichen Ideen,

die hinter Informatik in der Schule stehen und damit auch wichtige Themen für die Schulinformatik darstellen. Die Themen des Schulunterrichts wurden von Hubwieser et al. bereits mithilfe eines empirischen Ansatzes untersucht [Hu15]. Sie verglichen 2015 den Informatikunterricht in zwölf verschiedenen Ländern unter anderem hinsichtlich der Ziele, Inhalte sowie verwendeten Programmiersprachen und -werkzeuge. Ein zentrales Ergebnis der Studie waren 19 Kategorien, die die Inhalte der Schulcurricula auf Basis von acht der untersuchten Länder abbilden. Diese Studie wurde auch zur Orientierung für die hier erfolgende Analyse herangezogen.

Auch Kurse für Studierende anderer Fachrichtungen beschäftigen sich häufig mit Programmierung bzw. deren Grundlagen. Besonders in der Informatik spielen daher Werkzeuge eine bedeutende Rolle beim Lernen. Die in der Studie von Hubwieser et al. identifizierten Programmiersprachen und -werkzeuge wurden in die Kategorien didaktische Umgebung mit eigener Programmiersprache, didaktische Umgebung basierend auf einer anderen Programmiersprache und professionell genutzte Sprachen eingeordnet. Eine andere Herangehensweise Programmierwerkzeuge zu klassifizieren, stellt der Ansatz von Kelleher und Pausch dar [KP05]. Deren Kategorisierung von Programmierumgebungen für Einsteiger unterscheidet Teaching und Empowering Systems und kann im Rahmen von Bildungsangeboten dazu verwendet werden, mögliche Intentionen, die mit genutzten Werkzeugen einhergehen, zu erfassen.

3 Forschungsfragen

Die Konzeption von Informatikkursen für Studierende anderer Fachrichtungen erfordert eine bewusste Auswahl der zu erreichenden Ziele, zu behandelnden Themen oder zu verwendenden Werkzeugen. Bisher besteht keine Einigkeit über eine ausdifferenzierte Basis mit der jeder vertraut sein sollte. Daher soll in einem ersten Schritt ein empirischer Ansatz helfen, die Grundlagen der Informatik für alle Studierenden zu bestimmen. Diese sollten sich in entsprechenden Hochschulkursen für diese Zielgruppe widerspiegeln. Eine Analyse dieser Kurse kann demnach helfen, wesentliche Aspekte zu identifizieren. Dabei können im Kurs behandelte Themen nicht losgelöst betrachtet werden, da diese möglicherweise direkt von den angestrebten Zielen beeinflusst werden. Gleichzeitig beziehen sich informatische Bildungsangebote oft stark auf Programmierwerkzeuge bzw. -sprachen. Im weiteren Verlauf werden daher folgende Fragestellungen untersucht:

- Welche Ziele setzen sich Informatikkurse für Studierende anderer Fachrichtungen?
- Welche Themen der Informatik sind in Informatikkursen für Studierende anderer Fachrichtungen zentral?
- Welche Typen von Programmiersystemen werden in Informatikkursen für Studierende anderer Fachrichtungen eingesetzt?

4 Vorgehen

Das Vorgehen orientiert sich zunächst an den Schritten der qualitativen Inhaltsanalyse nach Mayring [Ma00]. Ziel dieser Analyse ist es, die zentralen Aspekte des untersuchten Materials kontextabhängig vollständig abzubilden. Zentral ist dabei die Auswahl einer geeigneten Datenbasis, die möglichst repräsentativ sein sollte. In dieser Untersuchung wurden daher 70 nationale wie internationale Kurse mit verschiedenen Ansätzen und unterschiedlichen Zielgruppen berücksichtigt. Grundlegend für die Ansätze Schwills und Dennings ist die Annahme, dass informatische Bildung sich auf zeitbeständige Aspekte fokussieren sollte. Daher wurden Dokumente von 2001 bis 2018 berücksichtigt. Kurse, die ausschließlich das Erlernen einer bestimmten Programmiersprache oder den Umgang mit bestimmten Anwendungen zum Ziel hatten, ohne sich auf die zugrundeliegenden Konzepte zu beziehen, wurden nicht in die Datenbasis übernommen. Kriterien waren außerdem eine Berücksichtigung in Studienplänen für Studierende anderer Fachrichtungen und ausreichend verfügbare Informationen in Form konkret benannter Ziele und Themen. Die Datenbasis umfasst sowohl Kurse, die rein für bestimmte Studierendengruppen, wie angehende Lehrerinnen und Lehrer, Historikerinnen und Historiker oder Biologinnen und Biologen konzipiert sind, als auch solche für alle Studierenden der jeweiligen Bildungseinrichtung.

Eine anfängliche Auswahl erfolgte mithilfe verfügbarer Publikationen in der ACM Digital Library. Hier wurde eine Schlagwortsuche mit „computing education“, „computer science education“, „computer science for non-majors“, „computing curriculum“ bzw. „CS0“ durchgeführt und jeweils die ersten 200 Treffer begutachtet. Innerhalb Deutschlands wurden die Vorlesungsverzeichnisse aller staatlichen Universitäten hinsichtlich der Begriffe „Informatik“, „Computer“ und „digital“ durchsucht. Für die endgültige Auswahl wurden die entsprechenden Kurse anschließend auf Relevanz nach o. g. Kriterien hin untersucht. Ergänzt wurde dies durch eine ausführliche Internetrecherche. Diese lieferte neben zusätzlichen Informationen zu den Kursen,

wie Wochenplänen und Skripten, auch Material von weiteren Veranstaltungen. Ständen zu einem Kurs mehrere Quellen zur Verfügung, so wurde jeweils eine Auswahl getroffen, die insbesondere auf Detailgrad und Aktualität der Dokumente basierte. Eine Übersicht über Universitätsstandorte der untersuchten Kurse findet sich in Abbildung 1.

Der nächste Schritt ist die Entwicklung eines Kategoriensystems. Ein solches kann entweder deduktiv aus bestehender Theorie oder induktiv aus der Datengrundlage abgeleitet werden. Bei den Untersuchungsaspekten *Ziele und Inhalte* wurde eine induktive Kategorienbildung vorgenommen. Damit reduziert sich das Risiko, wichtige Aspekte aufgrund vorher festgelegter Kategorien nicht zu berücksichtigen. Wie die Klassifizierung von didaktischen Programmiersystemen von Kelleher und Pausch [KP05] zeigt, können Programmierwerkzeuge auch bestimmte Ziele unterschiedlich gut unterstützen. Daher wurde für den dritten Aspekt (Programmiersprachen und -werkzeuge) auf eine deduktive Entwicklung der Kategorien auf Basis dieses Ansatzes zurückgegriffen, es wurden jedoch induktive Ergänzungen zugelassen.

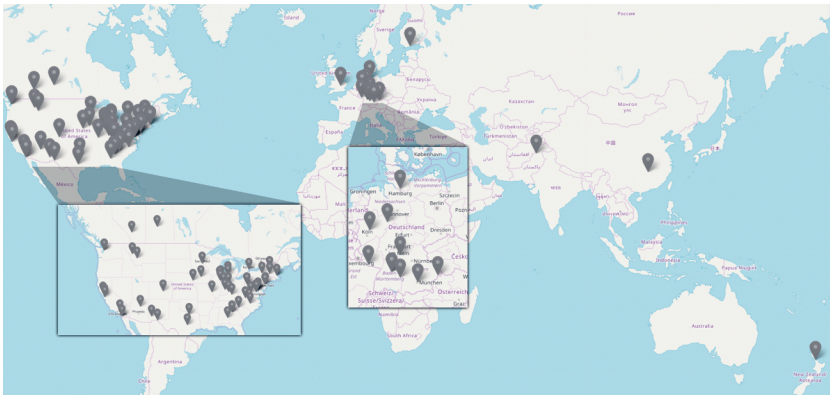


Abb. 1: Übersicht über die untersuchten Kurse für Studierende anderer Fachrichtungen

Die Darstellung der Ziele, Inhalte und Themen innerhalb der Kurse variierte teils deutlich. Während einige Veranstaltungskalender oder Syllabi sich auf eine Stichwortliste ihrer Inhalte beschränkten, enthielten andere weitaus detailliertere Informationen. Daher konnte eine zu kodierende Textpassage (Coding) sowohl aus einem Einzelwort, wie dem Titel einer Vorlesung, als auch aus einem ganzen Satz, beispielsweise einem Lernziel eines Kurses, be-

stehen. Insbesondere aus einem einzelnen Wort bestehende Codings wurden stets im Kontext interpretiert: Bei einer Verwendung des Wortes in anderem Sinnzusammenhang oder zur Abgrenzung von anderen Begriffen erfolgte keine Kodierung. Soweit verfügbar wurde zusätzliches Material zur Erklärung herangezogen.

Nach Auswahl des Materials sowie Festlegung der Kategoriensysteme und Kodierungskriterien wurde die eigentliche Analyse softwaregestützt mit der Analysesoftware MaxQDA durchgeführt. Ausgangspunkt war ein Set bestehend aus zehn Kursen, sukzessive erweitert um je drei bis fünf weitere Kurse. Dabei wurde für jedes neu auftretende Coding überprüft, ob dieses in eine bereits bestehende Kategorie eingeordnet werden kann oder, wenn nicht, eine neue Kategorie angelegt. Sollte eine neue Kodierung hingegen in mehr als eine Kategorie passen, so waren die betroffenen Kategorien zu speziell und wurden zu einer neuen Kategorie mit entsprechender Benennung zusammengefasst. Nach jeder dieser Erweiterungen wurde auch das zuvor bereits analysierte Material neu bewertet und gegebenenfalls bereits existierende Codings neu eingeordnet oder Kategorien vereinigt.

Wegen des unterschiedlichen Detailgrads in der Darstellung durch die verschiedenen Autoren ist eine Aussage aufgrund der Anzahl an Kodierungen pro Dokument und deren jeweiliger Länge nicht valide möglich. Eine Vielzahl an Kodierungen eines Aspekts lässt nicht zwangsläufig auf eine höhere Bedeutung dieses Aspekts schließen. Zur Darstellung und Interpretation der Ergebnisse wurden daher die relativen Häufigkeiten ermittelt. Dabei wurde das Auftreten einer Kategorie pro Dokumentgruppe als Wahrheitswert kodiert. Da vor allem die Bedeutung einzelner Themen relativ zu anderen von Interesse war, ist eine solche Quantifizierung der Beantwortung der Fragestellungen zuträglich. Ähnliche Ansätze wurden bereits mehrfach zur Analyse von Curricula angewendet (bspw. bei [BKM15], [GR14]).

5 Ergebnisse

5.1 Verfolgte Ziele

Die Kursverantwortlichen verfolgen mit ihren Angeboten bestimmte Ziele. Während einige Kurse vor allem ein einzelnes Ziel in den Vordergrund stellten, verfolgten andere eine Kombination mehrerer Ansätze. Die inkrementelle Kategorisierung der intendierten Ziele führte zu vier verschiedenen Typen. Ein Kurs verfolgte dabei im Schnitt Ziele aus 1,87 dieser Kategorien (Modalwert 2).

(G1) Denkweisen. In diese Kategorie fallen Kurse, die das Vermitteln bestimmter Denkweisen als ein primäres Ziel herausstellen. Dazu gehören das Vermitteln von Computational Thinking, algorithmischem, kreativem oder auch problemlösendem Denken.

(G2) Fluency. Durch informatische Bildung wird ein tieferes Verständnis der verwendeten Technologien vermittelt. Studierende sollen befähigt werden, Informatiksysteme effizient und gewinnbringend zur Lösung von Problemen einzusetzen.

(G3) Wissenschaft. Kurse sollen einen Überblick über das Fachgebiet Informatik geben. Den Studierenden werden zentrale Ideen und Schlüsselkonzepte der Wissenschaft Informatik aufgezeigt. Es geht häufig auch darum, ein (breites) Bild der Disziplin zu vermitteln und gleichzeitig das Verständnis für dessen grundlegende Konzepte zu schaffen.

(G4) Gesellschaft. Studierende sollen den Einfluss und die Auswirkungen von Informatik und Informatiksystemen auf die Gesellschaft und auf ihr persönliches zukünftiges Leben verstehen. Es gilt ausreichend Informatikwissen zu erwerben, um beispielsweise Auswirkungen von Informatiksystemen diskutieren zu können.

Bezogen auf die Datenbasis zeichnet sich ein Trend zu den Zielkategorien (G3) Wissenschaft und (G1) Denkweisen ab, die insgesamt in 41 (59 %) bzw. 38 Kursen (54 %) vorkamen. (G2) Fluency wurde in 29 (41 %), (G4) Gesellschaft zumindest in 23 Kursen (33 %) kodiert. Unter den Publikationsquellen dominiert die Kategorie (G1) Denkweisen mit 75 %. Obwohl die Kategorie bereits zu Beginn der betrachteten Zeitspanne kodiert werden konnte, ist insbesondere nach Popularisierung des Begriffs „Computational Thinking“ ein deutlicher Anstieg zu verzeichnen.

Zusätzlich zu den Intentionen der Kurse fanden sich in den Dokumenten auch Begründungen, warum jeder Studierende informatische Bildung erwerben sollte. Diese Begründungen überschneiden sich mit den von Döbeli Honegger genannten Argumenten [Dö16], erweitern diese jedoch auch teilweise. In Bezug auf die ersten beiden Kategorien werden insbesondere die Notwendigkeit der Vorbereitung auf zukünftige berufliche Tätigkeiten, die Aussicht auf höher bezahlte Jobs und eine steigende Problemlösekompetenz (*Problemlöseargument*) angeführt. Die Denkprozesse, die beim Lösen informatischer Probleme involviert sind, helfen auch, Probleme in anderen Fach-

bereichen zu bewältigen oder dort neue Impulse zu liefern. In Zusammenhang mit der Kategorie (G1) Denkweisen wird oft auch das *Wissenschaftsargument* – Informatikkompetenz hilft neue wissenschaftliche Erkenntnisse zu gewinnen – betont. Dass informatische Bildung eine effiziente Nutzung von Informatiksystemen ermöglicht, basiert auf dem *Konzeptwissenargument*. Um die Möglichkeiten von Informatiksystemen voll auszuschöpfen, reichen Anwenderkompetenzen nicht aus, sondern es werden auch Informatikkompetenzen benötigt. Vorwiegend zeigt es sich bei (G2) Fluency. Kernprinzipien oder ein breites Bild der Wissenschaft zu vermitteln (G3), geht gerade im amerikanischen Raum teilweise mit dem Wunsch einher, Studierende für ein Nebenfach oder für eine Vertiefung in Informatik zu gewinnen (*Berufswahlargument*). Außerdem thematisieren diese Kurse die Grundlagen digitaler Technologien und entmystifizieren damit die digitale Welt (*Welterklärungsargument*). Kurse der Kategorie (G4) rücken sichtbare Phänomene der Informatik, ein Verständnis der technischen Welt und vor allem ihre Wechselwirkungen mit der Gesellschaft in den Vordergrund. Informatik wird in diesem Zusammenhang als Teil mündiger Staatsbürgerschaft gesehen. Es stehen vor allem das *Mündigkeitsargument* und *Denkobjektargument* – Vorstellungen über wichtige Konzepte unseres Lebens schärfen – dahinter. Hinzu kommt hier ein *Ethikargument* – die Notwendigkeit unterscheiden zu können zwischen dem, was technisch möglich und dem, was ethisch vertretbar ist.

In der „Dagstuhl-Erklärung zur Bildung in einer digitalisierten Welt“ wird deutlich gemacht, dass eine gesellschaftlich-kulturelle, eine anwendungsorientierte und eine technologische Perspektive auf Phänomene, Artefakte, Systeme oder Situationen der digitalen vernetzten Welt eingenommen werden sollte [BD17]. Die hier identifizierten Zielkategorien lassen sich auf diese Perspektiven abbilden. So rücken sie bestimmte Perspektiven unterschiedlich stark in den Vordergrund: Gesellschaft („Wie wirkt das?“), Fluency („Wie nutze ich das?“) oder Wissenschaft und Denkweisen („Wie funktioniert das?“). Informatikbildung stellt eine Basis für alle diese Perspektiven dar und unterstützt alle Zieldimensionen, beispielsweise um Informatiksysteme gewinnbringend zu nutzen oder gesellschaftliche Auswirkungen der Digitalisierung erst zu reflektieren.

5.2 Betrachtete Themen

Die analysierten Dokumente zeigen eine Vielfalt an Zugängen zu informatischer Bildung für Studierende anderer Fachrichtungen, bspw. über Geoinformationssysteme oder eine Datenzentrierung. Die Analyse im letzten Abschnitt offenbarte die verschiedenen vorhandenen Ziele, deren Fokus sich teils deutlich unterscheidet. Aus diesem Grund sollten die Themen auch im Kontext der jeweiligen Intentionen betrachtet werden. Die aggregierten Themennennungen wurden hierzu mit den identifizierten Zielen der Kurse ins Verhältnis gesetzt. Die relativen Häufigkeiten einzelner Themen bezüglich der Zielkategorien ist in Abbildung 2 dargestellt.

Trotz der unterschiedlichen Zielsetzungen lassen sich gemeinsame Themen feststellen, die in jeder Zielkategorie in über 50% der Kurse kodiert wurden. Diese gemeinsame Basis wird aus den sieben Themenbereichen *Algorithmik*, *Programmierung*, *Repräsentation von Daten*, *Computerorganisation*, *soziale Implikationen*, *Datennutzung*, sowie *Netzwerke* gebildet. Damit wurde beispielsweise der Umgang – z. B. Analyse oder Visualisierung – mit (mehr oder weniger großen) Datenmengen relativ betrachtet deutlich häufiger thematisiert als traditionelle Themen wie *Datenbanken* oder *formale Sprachen*.

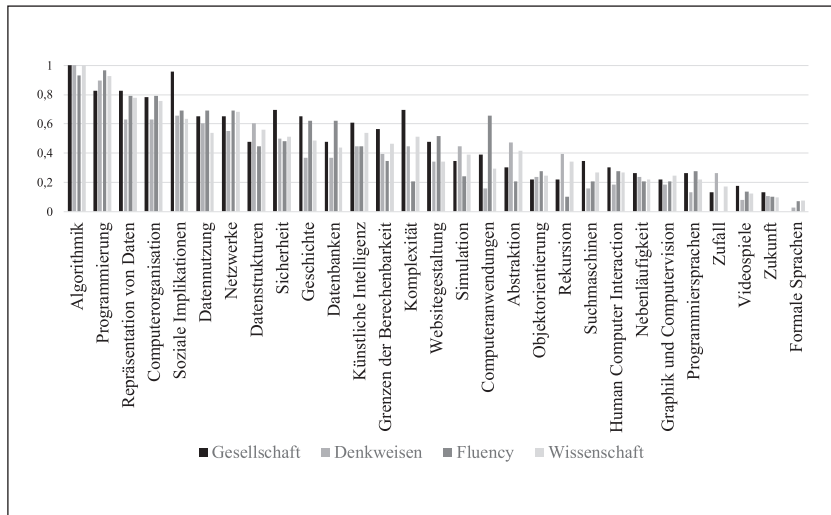


Abb. 2: Betrachtete Themenbereiche dargestellt in relativer Häufigkeit

Es zeigt sich, dass neben absolut grundlegenden Themen wie *Algorithmik*, *Programmierung* oder *Repräsentation von Daten*, auch eher technische Themen, wie *Computerorganisation* (Komponenten eines Rechners, die Von-Neumann-Architektur und logische Schaltungen) oder *Netzwerke* (insbesondere Aufbau und die Funktionsweise des Internets) im Rahmen eines einsemestrigen Angebots von einer Mehrheit der untersuchten Kurse als relevant eingestuft wurden.

Eine Analyse der Häufigkeiten bezogen auf einzelne Zielkategorien erlaubt weitere Einsichten und lässt bestimmte Zusammenhänge erkennen. Tabelle 1 zeigt die Themen, die neben o. g. Themen in über 50% der Kurse einer Kategorie kodiert wurden. So enthalten Kurse aus der Kategorie (G2) Fluency eher die Themen *Anwendungssoftware*, *Geschichte*, *Datenbanken* oder *Webentwicklung*. Kurse mit gesellschaftsbezogenen Zielen betonen *Sicherheit*, *Komplexität*, *Grenzen der Berechenbarkeit* und geschichtliche Aspekte stärker, während sich bei der Kategorie (G3) Wissenschaft *Datenstrukturen*, *künstliche Intelligenz*, *Komplexität* oder *Sicherheit* hervortun. Sofern primäre Kursziele vor allem das Vermitteln von Denkweisen und Problemlösestrategien sind, ist die Anzahl weiterer (häufiger) Themen im Vergleich geringer.

Tab. 1: Zielkategoriespezifische Themen

(G1) Denkweisen	(G2) Fluency	(G3) Wissenschaft	(G4) Gesellschaft
Datenstrukturen	Anwendungssoftware	Datenstrukturen	Sicherheit
Sicherheit	Geschichte	Künstliche Intelligenz	Komplexität
	Datenbanken	Komplexität	Geschichte
	Webentwicklung	Sicherheit	Künstliche Intelligenz
			Grenzen der Berechenbarkeit

Vergleicht man die Liste an Themen aus der induktiven Analyse mit bekannten Katalogen, zeigt sich, dass alle zehn „Big Ideas“ nach Bell et al. [BTY18] in den Kategorien enthalten sind. Auch die sieben Ideen der AP CS Principles

Standards finden sich bis auf Kreativität, welche nicht als eigenständiges Thema auftritt, wieder. Schulcurricula scheinen teils andere Schwerpunkte zu setzen. Im Vergleich zur Untersuchung der Schulcurricula durch Hubwieser et al. [Hu15] weisen die Ergebnisse dieser Analyse ein abweichendes Kategoriensystem auf. Nichtsdestoweniger verfügen einige Kategorien über ein Äquivalent in der jeweils anderen Untersuchung. Die Gegenüberstellung zeigt, dass *Anwendungssoftware*, *Datenstrukturen* oder *Betriebssysteme* und *Geräte* in den acht untersuchten Schulcurricula anteilig deutlich häufiger kodiert wurden als in Hochschulkursen. Auf der anderen Seite war der Stellenwert von Themen wie *künstliche Intelligenz* oder *soziale Implikationen* im hier untersuchten Material höher. Allerdings war die Stichprobe bei der Untersuchung der Schulcurricula auch deutlich geringer.

Unterschiede zeigen sich aber auch zu traditionellen Hochschulkursen. Viele der bei Bröker et al. [BKM15] identifizierten Themenbereiche für Informatikstudierende sind in Veranstaltungen für Studierende anderer Fachrichtungen nicht relevant. Auswirkungen auf die Gesellschaft werden hingegen in Modulen für Informatikerinnen und Informatiker deutlich seltener thematisiert. Indes hat auch das Thema *Simulationen* relativ betrachtet bei Kursen für Studierende anderer Fachrichtungen eine höhere Bedeutung.

5.3 Typen eingesetzter Programmiersprachen und -werkzeuge

Die Analyse zeigte auch, dass Kurse häufig eng mit ihren verwendeten Werkzeugen verzahnt sind. Während einige Kurse aber das Programmieren stark in den Fokus rückten und eine oder mehrere Programmiersprachen bzw. -werkzeuge einsetzten, verzichteten 10% der Kurse explizit auf die Verwendung eines Programmierwerkzeugs.

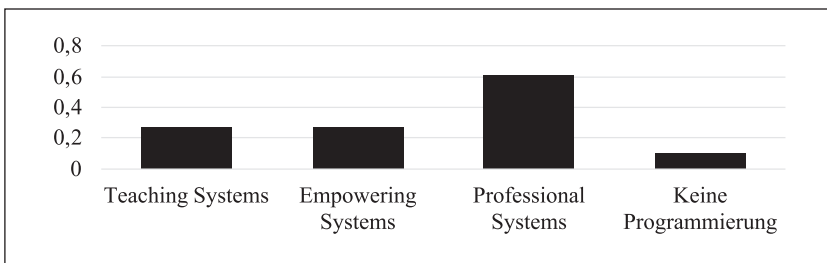


Abb. 3: Relativer Anteil verwendete Werkzeuge gruppiert nach Typ

Wie Abbildung 3 zeigt, finden professionell genutzte Werkzeuge die größte Verbreitung. In mehr als der Hälfte der Kurse werden auch – oder sogar ausschließlich – professionell genutzte Systeme eingesetzt, vor allem basierend auf Python. In den Kategorien (G1), (G2) und (G3) zeigt sich ein ähnliches Bild: Mit 65–72% liegt ein hoher Anteil professionell genutzter Systeme bzw. Sprachen vor, während Empowering Systems in rund einem Viertel der Kurse eingesetzt werden und nur 7–8% auf Programmiersysteme verzichten. In der Kategorie (G4) Gesellschaft hingegen setzen anteilig mehr Kurse auf Empowering Systems (35%) bzw. verzichten auf Programmierung (17%). Nur 43% verwendeten hier professionell genutzte Systeme. Bei Empowering Systems ist nicht entscheidend wie gut das erlernte Wissen auf in der Praxis übliche Programmierwerkzeuge übertragen werden kann, sondern dass die Nutzer mit diesen so viel wie möglich umsetzen können [KP05]. Obwohl Empowering Systems auch im universitären Kontext eingesetzt werden, zeichnet sich dennoch eine Tendenz hin zu transferierbarem Wissen oder professionellen Systemen ab. Gründe dafür liegen möglicherweise in der Verbreitung von Sprachen wie Python beispielsweise bei wissenschaftlichen Berechnungen.

6 Fazit

Die Ergebnisse lassen Aussagen über informatische Hochschulveranstaltungen für Studierende anderer Fachrichtungen hinsichtlich verfolgter Ziele, betrachteter Themen und verwendeter Werkzeugtypen zu. Zentral waren dabei weitestgehend unstrittige Themen wie Algorithmik, Programmierung oder Repräsentation von Daten, aber auch eher technische Grundlagen von Computern und dem Internet. Gegenüber traditionellen Angeboten zeigten sich vor allem Unterschiede hinsichtlich der Berücksichtigung reflexiver Komponenten zu sozialen Implikationen, insbesondere zu Privatsphäre, der Rolle der Informatik in der Gesellschaft oder Urheberrecht und der Berücksichtigung von Datenanalysen und -visualisierungen. Im Hinblick auf Programmierumgebungen lässt sich erkennen, dass Hochschulkurse für Studierende anderer Fachrichtungen teils didaktische Programmierumgebungen, insbesondere auch solche, die nicht für expliziten Transfer entwickelt wurden (Empowering Systems), einsetzen. Trotzdem wurden professionell genutzte Programmiersysteme in einer Mehrheit der Kurse verwendet.

Die Ergebnisse bieten einen Einblick in die verschiedenen Themen, die im Rahmen solcher Kurse betrachtet werden könnten. Damit können Ideen generiert, aber auch eine Einordnung bestehender Kurse getroffen werden.

Ein stärkerer Fokus auf Themen wie Datennutzung und Simulationen zeigt eben auch, dass ein Informatikkurs, der für Studierende anderer Fächer der einzige Kontakt zum Fach ist, neue Themen in Erwägung ziehen sollte, die traditionelle Veranstaltungen auslassen. Soll beispielsweise ein Kursangebot vor dem Hintergrund der Digitalisierung geplant werden, bieten sich neben der gemeinsamen Basis, insbesondere für die Förderung reflexiver Kompetenzen, auch Themen aus der Kategorie (G4) Gesellschaft an.

Außerdem liefern die Ergebnisse Anhaltspunkte dafür, welche Themen bei der Planung entsprechender Angebote direkt oder zumindest mit Anpassungen berücksichtigt werden sollten. Die am häufigsten betrachteten Themen bieten dabei einen Rahmen, der auch aus unterschiedlichen fachlichen Sichten betrachtet werden kann, um bspw. wichtige Algorithmen oder digital repräsentierte Informationen im jeweiligen Fachgebiet zu identifizieren. Repräsentation von Daten könnte sich für Geographinnen und Geographen beispielsweise auf Raster- und Vektordaten oder für Musikerinnen und Musiker auf MIDI und MP3 beziehen. Der Themenbereich künstliche Intelligenz könnte sich exemplarisch auf die Proteinstrukturvorhersage mit neuronalen Netzen für Biologinnen und Biologen oder auf computergenerierte Texte für geisteswissenschaftliche Studiengänge fokussieren.

Die Ergebnisse helfen darüber hinaus, die als für *jeden* als bedeutsam angenommenen Aspekte der Informatik empirisch zu belegen. Damit helfen sie nicht nur konkrete Kursinhalte zu rechtfertigen, sondern können auch zur Diskussion um die Bedeutung und Inhalte informatischer Bildung beitragen. Die quantitativen Betrachtungen erlauben auf Grundlage der Stichprobe Aussagen darüber, welche Themen anteilig als wichtiger erachtet wurden als andere und geben damit Hinweise darauf, welche Inhalte aus dem Themenfeld Informatik für jeden Studierenden relevant sind.

Literaturverzeichnis

- [Ba10] Barr, J. et al.: What everyone needs to know about computation. In: Proceedings of the 41st ACM technical symposium on Computer Science Education. ACM, New York, 2010, S. 127–128.
- [BTY18] Bell, T.; Tymann, P.; Yehudai, A.: The Big Ideas of K-12 Computer Science Education. In: Bulletin of EATCS Bd. 124 (2018).
- [BD17] Brinda, T.; Diethelm, I.: Education in the Digital Networked World. In: International Federation for Information Processing (IFIP): Proceedings of the World Conference on Computers in Education (WCCE 2017): Springer, 2017, S. 653–657.
- [BKM15] Bröker, K.; Kastens, U.; Magenheimer, J.: Competences of Undergraduate Computer Science Students. In: KEYCIT 2014 – Key Competencies in Informatics and ICT. Universitätsverlag Potsdam, Potsdam, 2015, S. 77–96.
- [Bu16] Bundesministerium für Wirtschaft und Energie: Digitale Strategie 2025, 2016. – www.bmwi.de/Redaktion/DE/Publikationen/Digitale-Welt/digitale-strategie-2025.pdf, Stand: 27.06.2018
- [BV18] Bundesarbeitgeberverband Chemie e. V.; Verband der chemischen Industrie e. V.: Digitale Bildung – Positionen und Forderungen der chemischen Industrie, 2018. – www.vci.de/langfassungen/langfassungen-pdf/2018-02-12-vci-bavc-digitale-bildung-positionen-forderungen-chemische-industrie.docx, Stand: 27.06.2018
- [De03] Denning, P.: Great Principles of Computing. In: Communications of the ACM Bd. 46 (2003), Nr. 11, S. 15–20.
- [Di11] Dierbach, C. et al.: A Model for Piloting Pathways for Computational Thinking in a General Education Curriculum. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education. ACM, New York, 2011, S. 257–262.
- [Dö16] Döbeli Honegger, B.: Mehr als 0 und 1: Schule in einer digitalisierten Welt. Bern, Schweiz: hep verlag, 2016.
- [Ge16] Gesellschaft für Informatik (GI): Bildungsstandards Informatik SI und SII, 2016. – www.informatikstandards.de, Stand: 27.06.2018
- [GR14] Grillenberger, A.; Romeike, R.: A Comparison of the Field Data Management and its Representation in Secondary CS Curricula. In: Proceedings of the 9th Workshop in Primary and Secondary Computing Education. ACM, New York, 2014, S. 29–36.
- [Hu15] Hubwieser, P. et al.: A Global Snapshot of Computer Science Education in K-12 Schools. In: Proceedings of the 2015 ITiCSE on Working Group Reports. ACM, New York, 2015, S. 65–83.

- [KP05] Kelleher, C.; Pausch, R.: Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. In: ACM Computing Surveys (CSUR) Bd. 37 (2005), Nr. 2, S. 83–137.
- [Ma00] Mayring, P.: Qualitative Content Analysis. In: Forum Qualitative Sozialforschung/Forum: Qualitative Social Research Bd. 1 (2000), Nr. 2.
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik Bd. 25 (1993), Nr. 1, S. 20–31.
- [VSC17] Vogel, S.; Santo, R.; Ching, D.: Visions of Computer Science Education: Unpacking Arguments for and Projected Impacts of CS4All Initiatives. In: Proceedings of the 2017 ACM Technical Symposium on Computer Science Education. ACM, New York, 2017, S. 609–614.
- [Wi06] Wing, J.: Computational thinking. In: Communications of the ACM Bd. 49 (2006), Nr. 3, S. 33–35.