

Sichtweisen einer kreativen Informatik

Ralf Romeike

Didaktik der Informatik
Universität Potsdam
A.-Bebel-Str. 89
14482 Potsdam
romeike@cs.uni-potsdam.de

Abstract: In diesem Artikel werden verschiedene Sichtweisen dargestellt, in denen Informatik mit Kreativität in Zusammenhang gebracht werden kann. Diese Sichtweisen lassen sich in der Schulinformatik thematisch und methodisch berücksichtigen und können damit für ein attraktiveres und facettenreiches Bild der Informatik sorgen. Ziel dieses Artikels ist es, damit eine Grundlage für eine kreativitätsorientierte Informatikbildung zu legen und die Rolle der Kreativität in der Informatik zu betonen. Damit wird die Diskussion um Kreativität in der Informatik auf eine Grundlage gestellt, die eine gezielte Auseinandersetzung und Argumentation ermöglicht. Implikationen und Anknüpfungspunkte für die Schulinformatik werden gegeben.

1 Einleitung

Hochschulen finden immer weniger Studienanfänger, die Informatik studieren möchten, das Privatfernsehen stellt Informatiker gar als „Freaks“ und unsoziale Personen dar,¹ die öffentliche Meinung von der Informatik könnte besser sein. Vor dem Hintergrund, dass Kinder und Jugendliche immer früher und intensiver mit dem Computer, und damit mit Informatik, in Berührung kommen, stellt sich die Frage, was läuft verkehrt in der informatischen Bildung? Maaß und Wiesner [MW06] stellen fest, dass sich Informatik für viele Schüler auf „Programmieren, Mathe und ein bisschen Hardware“ zu beschränken scheint. Dabei wird dieses Bild weder dem Informatikstudium noch den Tätigkeiten im Informatikerberuf gerecht. Gefordert wird, ein facettenreiches Bild der Informatik zu vermitteln.

In Freizeitaktivitäten wird der Computer häufig als kreatives Werkzeug genutzt. Dabei eignen sich die Nutzer viele Grundlagen, Techniken und Wissen der Informatik an, um eigene Vorhaben umzusetzen. Daran wird auch deutlich, dass Informatik als vierte Kulturtechnik inzwischen in das tägliche Leben Einzug gehalten hat. Die Idee, dass Schule den Auftrag hat, Schüler auf diese „neue“ Gesellschaft vorzubereiten, ist nicht

¹ So werden bspw. in der Pro7 – Show „Das Model und der Freak“ vorwiegend informatik- und computerinteressierte Jugendliche als unglücklich und lebensfremd dargestellt.

neu. Resnick [Re07] unterstreicht in diesem Zusammenhang die Notwendigkeit von Kreativität als Schlüsselqualifikation jedes Menschen in der Wissensgesellschaft. Mit der Informatik existiert unserer Auffassung nach ein Unterrichtsfach, welches im Umgang mit und durch die Thematisierung von Computertechnologien großes Potential besitzt, Kreativität der Schüler zu nutzen und zu fördern. Im Unterricht wird diese attraktive Betrachtungsweise von Informatik bisher allerdings nur unzureichend deutlich [vgl. MW06, Ro07a].

Im Folgenden werden wir verschiedene Sichtweisen darstellen, in denen Informatik mit Kreativität in Zusammenhang gebracht werden kann. Diese Sichtweisen lassen sich in der Schulinformatik thematisch und methodisch berücksichtigen und können damit für ein attraktiveres und facettenreiches Bild der Informatik sorgen. Ziel dieses Artikels ist es, damit eine Grundlage für eine kreativitätsorientierte Informatikbildung zu legen und die Rolle der Kreativität in der Informatik zu betonen. Damit wird die Diskussion um Kreativität in der Informatik auf eine Grundlage gestellt, die eine gezielte Auseinandersetzung und Argumentation ermöglicht. Implikationen und Anknüpfungspunkte für die Schulinformatik werden gegeben.

2 Kreativitätsperspektiven

2.1 Standpunkte

„Computer science is the first engineering discipline ever in which the complexity of the objects created is limited by the skill of the creator and not limited by the strength of the raw materials” [Brian K. Reid zitiert nach Fr87]. Die Grundlagen, die notwendig sind, um innerhalb und mit der Informatik kreativ tätig zu werden, verbessern sich durch einfachere Software und intuitive Benutzerinterfaces kontinuierlich. Schon Kinder und Jugendliche beschäftigen sich kreativ mit Informatiksystemen und Inhalten der Informatik und wählen sogar ein Informatikstudium aus diesem Grund [RR96, KR08]. Innerhalb der Informatik wird vor allem dem Gebiet der Softwareentwicklung und des Programmierens eine kreative Rolle zugeschrieben [LA05, ST05, GI06]. Die öffentliche Meinung zur Kreativität in der Informatik unterscheidet sich allerdings deutlich von der vieler Informatiker. Studien zeigen, dass Informatik eher als un kreativ, unsozial und anwendungsfern attribuiert wird [AA00, MF02, RPG04, BGR05]. Nach einer Studie zum Image des Ingenieurberufs² bezeichnen drei Viertel der befragten Jungingenieure ihr Berufsfeld als kreativ, während diese Meinung nur von 27% der Öffentlichkeit in Deutschland geteilt wird [VDE03].

Während Befürworter eines kreativen Informatikbilds vor allem den schöpferischen Aspekt informatischer Tätigkeiten betonen, beziehen sich Kritiker v. a. auf

² Die erwähnte Studie bezieht sich auf den Ingenieurberufs insgesamt. Ein Großteil der Befragten Jungingenieure stammt allerdings aus dem Bereich Elektro- und Informationstechnik/Informatik. Es ist zu vermuten, dass das Bild der Informatik in der Öffentlichkeit nicht wesentlich anders ist. Eine Studie hierzu ist uns nicht bekannt.

Charaktermerkmale kreativer Personen, wie Unangepasstheit und Eigensinn. So wird von Problemen berichtet, die „kreative“ Informatiker verursachen können, weil sie sich nicht an Absprachen halten und eigenmächtig „coole Features“ implementieren [Mc07] oder weil Kreativität in der professionellen Softwareentwicklung einen Kontrollverlust herbeiführen könnte [Cu91]. Dass selbst formale Methoden Kreativität nicht ausschließen müssen und auch in der professionellen Softwareentwicklung ein kreativer Ansatz erfolversprechend sein kann, verdeutlicht dagegen Glass [Gl06].

Welches Bild von der Informatik haben Informatikschüler? Erfahrungen zeigen, dass viele Schüler das gesellschaftliche Bild von der Informatik teilen. Der schulische Informatikunterricht ist daran nicht schuldlos, wird doch Kreativität in der Unterrichtsgestaltung nur selten berücksichtigt [Ro07a, Ro08b]. Auch Schüler, welche die kreative Seite der Informatik für sich entdeckt haben, attribuieren den Informatikunterricht eher als un kreativ und negativ, sehen das Fach selbst allerdings durchaus als kreativ an [KR08]. Ein explizit kreativ gestalteter Unterricht kann das Schülerbild für die Informatik allerdings positiv beeinflussen [Ro08a].

2.2 Kreative Persönlichkeiten der Informatik

Nicht selten sind es Personen, an denen Kreativität verdeutlicht wird – Mozart oder Einstein werden bspw. unstrittig als kreative Persönlichkeiten angesehen. Auch (oder gerade) in der Informatik gibt es eine Vielzahl kreativer Persönlichkeiten, welche die Entwicklung der Informatik vorangetrieben haben. Ebenso vermag das Fach kreative Personen herauszubilden. Leach und Ayers [LA05] untersuchten Aufsätze über und Interviews mit prominenten Informatikern mit dem Ziel, die Rolle von Kreativität in deren Biographien zu ergründen. Übereinstimmend finden sich bei nahezu allen Informatikern: Eine frühe Begegnung mit Kreativität, ein frühes selbständiges Auseinandersetzen mit algorithmischen oder mathematischen Problemen, welche die Informatiker herausforderten, Situationen, die sich für sie als unbefriedigend oder herausfordernd darstellten und zur weiteren Auseinandersetzung motivierten, Stimuli, welche die Informatiker zu weiteren Gedanken inspirierten (andere Arbeiten, Hintergrundwissen, nahe stehende Personen, interessante Uni-Kurse, provokante Fragen) sowie die Möglichkeit, eigenen Fragestellungen nachzugehen und mit möglichen Lösungen zu experimentieren (eine stimulierende Umgebung).

Für den Informatikunterricht sind diese Aspekte aus zwei Gründen interessant. Zum einen, kann anhand der Personen und ihrer Errungenschaften der kreative Charakter der Informatik herausgestellt werden. Zum anderen bieten die Aussagen Ansatzpunkte, wie eine Informatikunterrichtsumgebung gestaltet werden kann, um den Schülern einen kreativen Start in die Informatik zu ermöglichen.

2.3 Innovationen der Informatik

Innovation bezeichnet die Verwirklichung kreativen Denkens. Trotz der relativ jungen Geschichte der Informatik, ist aus ihr bereits eine Fülle von Innovationen hervorgegangen. Damit hat die Informatik – vielleicht wie keine andere Wissenschaft

vor ihr – in kürzester Zeit die Gesellschaft verändert. Auch das Selbstverständnis der Informatik wird zunehmend von dem bestimmt, was sie geschaffen hat. So illustriert die Gesellschaft für Informatik ihr Verständnis für Informatik auf einer 24-seitigen Darstellung von Innovationen durch Informatik [GI06].

Eine Möglichkeit, Innovationen im Unterricht herauszustellen, ist die historische Betrachtung der Informatik [bspw. in Th05], wie es auch in verschiedenen Schulbüchern gemacht wird.³ Viele dieser geschichtlichen Darstellungen haben allerdings gemeinsam, dass nur die (Hardware-) technische Seite der Entwicklung thematisiert wird. Der Großteil heutiger Innovationen vollzieht sich allerdings softwareseitig. Ebenso ist es Schülern fast ausschließlich im Bereich der Software möglich, eigenständig informatisch kreativ tätig zu werden. Eine Betrachtung der Softwareinnovationen kann dies zusätzlich anregen. Eine ausführliche Darstellung dieser findet sich bei [Wh08].

2.4 Kreativität in der Softwareentwicklung

Kein Bereich der Informatik ist so untrennbar mit dem Schaffen kreativer Produkte verbunden wie die Softwareentwicklung. Kreativität ist in verschiedenen Phasen des Softwareentwicklungsprozesse erforderlich:

Finden und Bestimmen einer „Challenge“⁴: Präzise definierte Aufgaben sind für die Softwareentwicklung eher untypisch. Vielmehr steht am Anfang in der Regel das Sondieren von Problematik, Möglichkeiten und Anforderungen sowie eine Präzisierung dessen, was man erreichen möchte. Situationen, in welchen aus eigenem Antrieb Software entwickelt wird, gehen sogar darüber hinaus, indem vor dem Hintergrund der eigenen Möglichkeiten und Fertigkeiten erst ein realisierbares Ziel gefunden werden muss. Dieser Prozess des Problemsuchens und -findens erfordert viel Kreativität. Klar definierte Aufgaben dagegen, wie sie häufig im Unterricht zu finden sind, ignorieren diesen kreativen Teil der Softwareentwicklung. Schüler sollen doch auch entdecken können, wofür sie ihr Gelerntes anwenden können! Dies kann nur in einem Unterricht funktionieren, in dem eigene Ideen eingebracht werden können [vgl. Ro08b].

Problemmanagementphase: Glass [GI06] beschreibt diesen Teil als hochkreativen kognitiven Prozess, welcher auch Versuch und Irrtum und heuristische Herangehensweisen involviert. Dieser Aspekt wird häufig übersehen, wenn Lehrer eher einen Top-Down-Modellierungsansatz unterstützen. Vielmehr ist auch hier ein wiederholtes Generieren von Hypothesen und ggf. Prototypen angebracht und üblich (welches natürlich kein planloses Versuch-Irrtum-Vorgehen beinhaltet). In diesem Teil sind zahlreiche Lösungswege und Vorgehensweisen möglich, wessen sich die Schüler bewusst sein sollten. Durch den Informatiklehrer kann dieses Phase des kreativen Prozesses unterstützt werden, indem dafür gesorgt wird, dass fachrelevantes Wissen,

³ Hierbei darf allerdings nicht vergessen werden, dass es sich bei vielen Innovationen nur um Vorläufer der Informatik, nicht um Innovationen der Informatik selbst, handelt.

⁴ Challenge lässt sich übersetzen als Herausforderung oder „lockende Aufgabe“. Leider enthält die deutsche Sprache kein Wort, welches es so gut trifft.

Fertigkeiten und Motivation vorhanden sind, das Sammeln von Erfahrungen ermöglicht und unterschiedliche Vorgehensweisen thematisiert werden.

Produkt: Typisch für die Informatik ist es, dass es für eine Problematik verschiedenartige Lösungen gibt. Ein kreativer Unterricht streicht diese Besonderheit heraus. Nach Fertigstellung eines Produktes ergibt sich dank offener Aufgaben immer die Möglichkeit, weitere Optimierungen oder funktionale Erweiterungen vorzunehmen.

Präsentation: Das Verwirklichen von Softwarelösungen ist nicht nur ein Selbstzweck zum Testen eines Entwurfs, sondern erst das funktionierende Produkt stellt die Belohnung für das Engagement dar. Die Präsentation des Projekterfolgs (z.B. vor der Klasse oder online) schließt den Softwareentwicklungsprozess im Informatikunterricht gebührend ab.

Für den Informatikunterricht ist das Entwickeln von Software vor allem dann reizvoll, wenn die Schüler die Möglichkeit bekommen, eigene Ideen umzusetzen und Programme zu entwickeln, die für sie selbst relevant sind. Erst hieraus kann sich die intrinsische Motivation entwickeln, die für kreative Leistung unabdingbar ist.

2.5 Motivation durch kreatives Tun

“Software Development is and always has been fun” [G106]. Motivation ist die treibende Kraft hinter jeder menschlichen Tätigkeit und grundlegend für Lernprozesse. Während Intelligenz, Begabung und sozialer Hintergrund Faktoren darstellen, die Lernen beeinflussen, aber außerhalb der pädagogischen Wirkungsmöglichkeiten liegen, kann die Lernmotivation durch geschickte Steuerung des Lernprozesses vom Lehrer gestärkt werden. Intrinsisch motivierte Schüler erreichen auch bessere Leistungen.

Kreativität und Motivation stehen in einem wechselseitigen Verhältnis. Studien zur Motivation von Open-Source-Softwareentwicklern identifizieren intrinsische Motivation durch das Gefühl, etwas Kreatives zu tun, als wichtigsten Faktor, warum sich diese Programmierer hier engagieren [LW05]. Auch im Informatikunterricht kann dieser Faktor eine Rolle spielen und genutzt werden. So berichten auch Schüler von Motivation durch den schöpferischen Charakter der Informatik und die Möglichkeit, sich kreativ zu verwirklichen [Ro06].

2.6 Persönlichkeitsfaktoren

Im Kontext der Psychodiagnostik und der individuellen Einschätzung der Schüler gibt es Versuche, nicht nur hochbegabte, sondern auch hochkreative Schüler frühzeitig zu erkennen und zu fördern. Aber woran lässt sich eine besonders kreative Person erkennen? Hohe Motivation, Ausdauer und die Bereitschaft, Hindernisse zu überwinden, intellektuelle Neugier und Offenheit für neue Erfahrungen, starker Einsatz, Nonkonformismus, Unabhängigkeit in Tun und Handeln, Toleranz für Unklarheit, der Wille zu wachsen, Risikobereitschaft, Glaube an Überzeugungen, Selbstsicherheit und breit gefächerte Interessen werden als typische Persönlichkeitsmerkmale kreativer

Personen angesehen [vgl. SL91, Fu00, Cr01]. Diese Merkmale können Lehrern Hinweise auf kreative Schüler geben und können im Unterricht unterstützt werden. Kreativitätstests sind allerdings in der psychologischen Forschung umstritten und bestimmen kreatives Potential nur unzuverlässig. Vor dem Hintergrund, dass Kreativität förderbar ist, empfiehlt es sich statt dessen ein kreatives Unterrichtsklima zu gestalten, das alle Schüler zur Entfaltung und Entwicklung von Kreativität anregt.

2.7 Kunstperspektive

Kunst, die *schöpferische Tätigkeit des Menschen mit Tönen, Sprache oder verschiedenen Materialien* [Be04] ist der Inbegriff von Kreativität in traditionellem Sinn.⁵ Auch das Entwickeln von Software ist durchaus mit kreativen Prozessen in der Musik [vgl. Ro07b] oder anderen Künsten vergleichbar ist. So erstaunt es nicht, dass einige Informatiker ihr Tun als Kunst verstehen: Der Informatiker als Künstler. Hier finden sich zwei verschiedene Sichtweisen. Zum einen richtet sich die Aufmerksamkeit auf den Prozess selbst (meist programmieren); der Informatiker *versteht* das, was er tut, als Kunst, basierend auf Fertigkeiten, Wissen und Freude am Schaffen schöner Software [Kn74]. Diesem Verständnis entspringt das ästhetische Empfinden für „schönen Code“ und „ansprechenden“ Softwareentwurf. Zum anderen wird weniger der Prozess reflektiert, sondern der Informatiker (auch hier oftmals Programmierer) *betätigt sich* als Künstler. In dieser Sichtweise richtet sich die Aufmerksamkeit auf das kreative, künstlerische Produkt, ein meist programmiertes Kunstwerk, welches Expertise, Eleganz, oft auch Effizienz und Kreativität gleichermaßen ausdrückt. Festivals und eine aktive Community/Szene bieten diesen Programmierern ein Podium.

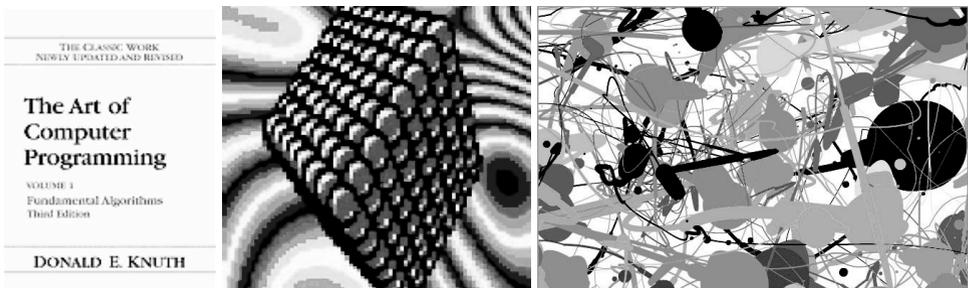


Abb. 1: Knuth: The Art of Computer Programming (l), Kunstwerk aus der DEMO-Szene: PC text mode demo: Bolognese von Alpha Design (m), Interaktive Kunst nach Pollock (r).

In einem anderen Zusammenhang steht Informatik mit Kunst dort, wo sich Künstler als Informatiker oder Programmierer betätigen. Diese haben die Methoden der Informatik und die Möglichkeiten, die sich daraus ergeben, für sich entdeckt und wenden sie zum Erschaffen von Kunstwerken an [vgl. Ma04, Tr05]. Im Vordergrund stehen hierbei nicht

⁵ Allerdings verhält es sich mit vielen Kunstwerken so, dass ein Kriterium vieler Definitionen von Kreativität verletzt wird: die Nützlichkeit. Der Nutzen eines Kunstwerks liegt oft in der subjektiven Betrachtung und am Genuss des Kunstwerks per se, ohne weiteren Nutzen darüber hinaus.

Effizienz, Eleganz oder Zuverlässigkeit der Software, sondern die Weiterentwicklung von Ideen und das ästhetische Gelingen der Arbeit.

Einen zusätzlichen Beitrag für die Erschaffung von Kunstwerken und für die künstlerische Betätigung leistet die Informatik in dem sie Werkzeuge bereitstellt, die künstlerisch genutzt werden können. Hierzu zählt unterschiedlichste Software für Bild-, Musik- und Videoverarbeitung bis hin zu interaktiven Komponenten für Liveinstallationen. Mit diesen Hilfsmitteln gelingt es immer mehr Menschen, sich künstlerisch kreativ zu betätigen.

2.8 Technologie-Perspektive

Das kreativitätsfördernde Potential von Informatiksystemen ist nicht nur auf künstlerische Bereiche beschränkt, sondern geht weit darüber hinaus. So beschäftigt sich ein großer Teil der Forschung im Spannungsbereich von Kreativität und Informatik mit den Möglichkeiten zur Förderung von Kreativität mit IT in professionellen Bereichen und Lernumgebungen [SFC05]. Neue Technologien unterstützen Anwender demnach bei der Entwicklung und beim Verbinden von Ideen, beim Erschließen von Bedeutungen, Kreieren, Zusammenarbeiten, Kommunizieren, Publizieren und Weiterentwickeln. Hierbei kann IT verschiedene Rollen übernehmen: Als Nanny (Aufpasser), Netzwerker und Unterstützer, Coach oder Kollege [Lu05]. Eine kreativitätsfördernde Umgebung ist ein wesentlicher Faktor für die Förderung von Kreativität. Ein herausragendes Beispiel für den Informatikunterricht ist die Entwicklungsumgebung Scratch⁶. Scratch ermöglicht das intuitive Erstellen von interaktiven Programmen und Animationen durch Exploration und Experimentieren, eine visuelle Darstellung der Strukturen und Programmierbausteine, Inspiration, schrittweises Entwickeln und das problemlose Veröffentlichen und Präsentieren der Ergebnisse auf einer Plattform im Internet. Damit erfüllt Scratch Shneidermans Kriterien für kreativitätsunterstützende Softwaretools [Sh02].

2.9 Artificielle Kreativität

Ziel verschiedener informatischer Bestrebungen ist es, Systeme zu konstruieren welche Kreativität nachbilden. Vier in diesem Zusammenhang zentrale Fragen wurden von Boden [Bo95] aufgestellt:

1. Können computerwissenschaftliche Vorstellungen beim *Verständnis* menschlicher Kreativität helfen?
2. Können Computer jemals etwas tun, das schöpferisch *wirkt*?
3. Kann ein Computer menschliche Kreativität *erkennen*?
4. Kann ein Computer jemals *tatsächlich kreativ* sein (im Gegensatz zu reiner Produktion scheinbar kreativer Leistungen, deren Originalität ausschließlich auf den menschlichen Programmierer zurückzuführen ist)?

Die ersten drei Fragen können inzwischen sicherlich mit „ja“ beantwortet werden,

⁶ <http://scratch.mit.edu>.

bleiben aber in ihrer Bedeutung nicht ohne Brisanz. Für die Diskussion der vierten Frage können verschiedenartigste Projekte herangezogen werden. Eines der bekanntesten und erfolgreichsten Projekte im Bereich artifizieller Kreativität ist das Programm Aaron [Co95]. Die Software generiert originäre komplexe Malereien und Zeichnungen, welche durchaus mit Werken versierter Künstler vergleichbar und sogar in verschiedenen Kunstausstellungen zu sehen sind. Dargestellt werden häufig menschliche Figuren, die vor verschiedenen Hintergründen bei Aktivitäten zu sehen sind. Hofstadter [Ho96] charakterisiert die Zeichnungen gar als „oftmals lustig und von einer bezaubernden Naivität, die ein unverkennbares Stilmerkmal darstellt“. Computerprogramme, welche *eigenständig* komponieren, existieren ebenfalls seit einiger Zeit [bspw. Lu08], genauso wie das erste Buch, welches durch ein Computerprogramm „geschrieben“ wurde [vgl. Ra84]. In akademischen Kreisen wurde der Paper-Generator populär, welcher pseudo-wissenschaftliche Artikel generiert, von denen nach Berichten sogar verschiedene Artikel auf Konferenzen oder in Journals angenommen wurden [SKA08].

In der Schulinformatik eröffnet die Diskussion von Bodens Fragen Anreize, über die Grenzen der Computerisierung nachzudenken oder sogar Versuche zu unternehmen, durch geschicktes Einbinden von Zufallselementen z.B. Ansätze artifizieller Musik oder Malerei aktiv nachzuvollziehen.

3 Diskussion

Gegenüber der systematischen, technischen oder historischen Betrachtung ist die Beschäftigung mit den kreativen Seiten der Informatik für die Schulinformatik äußerst attraktiv. Voraussetzend, dass Kreativität eine Eigenschaft ist, die sowohl förderungswürdig ist, als auch die Attraktivität eines Fachs erhöht, ergeben sich hieraus zahlreiche Anknüpfungsmöglichkeiten, anhand derer systematisch Unterrichtsinhalte thematisch verankert werden können.

Die verschiedenen Sichtweisen auf Kreativität in der Informatik weisen auf beeindruckende Möglichkeiten, Kreativität in der Informatik und durch die Informatik erlebbar zu machen. Diese Charakterisierung erhebt keinen Anspruch auf Vollständigkeit – weitaus mehr Sichtweisen können eingenommen werden, dieses spannende Interaktionsfeld zu betrachten und anzuwenden. Für die weitere Diskussion um Kreativität in der Informatik können sie durchaus eine Grundlage legen, die verschiedenen Verständnisse von Kreativität in Äußerungen und Argumenten einzuordnen und zu bewerten.

Die kreative Nutzung der Informatik und der Informatiksysteme sind als neue Kompetenzen in der Schulinformatik unserer Meinung nach zwingend notwendig. Aktuelle Bemühungen, Schülern ein Informatiksystemverständnis zu vermitteln, können demnach immer nur erste Schritte sein, welche erst in der kreativen Anwendung ihre Bedeutung erhalten. Betrachtet man Kreativität als Beitrag der Informatik an die Allgemeinbildung, eröffnet und vereinfacht Informatikunterricht zahlreiche Betätigungsfelder, die nicht nur dabei helfen, in einer kreativen Wissensgesellschaft zurecht zu kommen, sondern auch noch Spaß machen.

Literaturverzeichnis

- [AA00] American Association of University Women: Tech-Savvy: Educating Girls in the New Computer Age. American Association of University Women, Washington, DC, 2000.
- [BGR05] Barker, L. J.; Garvin-Doxas, K.; Roberts, E.: What can computer science learn from a fine arts approach to teaching? In SIGCSE Bull. 37(1), 2005; S. 421-425.
- [Be04] Bertelsmann: Bertelsmann Wörterbuch der deutschen Sprache. Wissen Media Verlag, Güters-loh/München, 2004.
- [Bo95] Boden, M. A.: Die Flügel des Geistes: Kreativität und künstliche Intelligenz. Dt. Taschenbuch-Verl., München, 1995.
- [Co95] Cohen, H.: The further exploits of Aaron, painter. In Stanford Hum. Rev. 4(2), 1995; S. 141-158.
- [Cr01] Craft, A.: An analysis of research and literature on creativity in education. In Report prepared for the qualification and curriculum authority. http://www.ncaction.org.uk/creativity/creativity_report.pdf (10.06.2008), 2001.
- [Cu91] Cusumano, M. A.: Japan's software factories: a challenge to U.S. management. Oxford University Press, Inc., 1991.
- [Fr87] Frenkel, K. A.: Profiles in computing: Brian K. Reid: a graphics tale of a hacker tracker. In Commun. ACM 30(10), 1987; S. 820-823.
- [Fu00] Funke, J.: Psychologie der Kreativität. In (Holm-Hadulla, R. M., Hrsg.): Kreativität, Heidelberger Jahrbücher Band 44, Heidelberg, Springer, 44,2000; S. 283-300.
- [GI06] GI: Was ist Informatik? Gesellschaft für Informatik e.V. (GI), Bonn, 2006.
- [GI06] Glass, R. L.: Software creativity 2.0. developer.* Books, Atlanta, 2006.
- [Ho96] Hofstadter, D. R.: Die FARGonauten. Über Analogie und Kreativität. Stuttgart: Klett-Cotta, 1996.
- [KR08] Knobelsdorf, M.; Romeike, R.: Creativity as a Pathway to Computer Science. Proc. 13th Annual Conference on Innovation and Technology in Computer Science Education (ITICSE 2008), Madrid, ACM Press, 2008.
- [Kn74] Knuth, D. E.: Computer Programming as an Art. In Communications of the ACM 17(12), 1974; S. 667 - 673.
- [LW05] Lakhani, K.; Wolf, R.: Why Hackers Do What They Do: Understanding Motivation Effort in Free/Open Source Software Projects. In (Feller, J.; Fitzgerald, B.; Hissam, S.; Lakhani, K. R., Hrsg.): Perspectives on Free and Open Source Software, MIT Press, 2005; S. 3-22.
- [LA05] Leach, R. J.; Ayers, C. A.: The Psychology of Invention in Computer Science. Proc. 17th Annual Workshop of the PPIG, University of Sussex, Brighton UK, 2005.
- [Lu05] Lubart, T.: How can computers be partners in the creative process: classification and commentary on the special issue. In Int. J. Hum.-Comput. Stud. 63(4-5), 2005; S. 365-369.
- [Lu08] Ludwig: Komponiersoftware. 2008. <http://www.komponieren.de/> (20.05. 2008).
- [MW06] Maaß, S.; Wiesner, H.: Programmieren, Mathe und ein bisschen Hardware... Wen lockt dies Bild der Informatik? In Informatik-Spektrum 29(2), 2006; S. 125-132.
- [Ma04] Maeda, J.: Creative code. Thames & Hudson London, New York, 2004.
- [MF02] Margolis, J.; Fisher, A.: Unlocking the Clubhouse: Women in Computing. MIT Press, 2002.
- [Mc07] McIlree, R.: Yet Another "Creativity" Rant Emerges. 2007. http://enterprisearchitect.typepad.com/ea/2007/01/yet_another_cre.html (25.04.2008).
- [Ra84] Racter: The Policeman's Beard Is Half Constructed. Warner Software/Warner Books, 1984.

- [Re07] Resnick, M.: Sowing the Seeds for a More Creative Society. Proc. Learning & Leading with Technology, International Society for Technology in Education (ISTE), 2007.
- [RR96] Resnick, M.; Rusk, N.: The Computer Clubhouse: Preparing for life in a digital world. In IBM Systems Journal 35, 1996; S. 431 - 439.
- [RPG04] Rich, L.; Perry, H.; Guzdial, M.: A CS1 course designed to address interests of women. Hrsg.): Proceedings of the 35th SIGCSE technical symposium on Computer science education, Norfolk, Virginia, USA, ACM Press, 2004; S. 190-194.
- [Ro06] Romeike, R.: Creative Students - What Can We Learn From Them for Teaching Computer Science? Proc. 6th Baltic Sea Conference on Computing Education Research, Koli Calling, Uppsala University, Uppsala, Sweden, 2006.
- [Ro07a] Romeike, R.: Kriterien kreativen Informatikunterrichts. Proc. 12. GI-Fachtagung "Informatik und Schule - INFOS 2007", Siegen, Germany, Köllen, 2007.
- [Ro07b] Romeike, R.: Three Drivers for Creativity in Computer Science Education,. Proc. IFIP-Conference on "Informatics, Mathematics and ICT: a golden triangle", Boston, USA, 2007.
- [Ro08a] Romeike, R.: Applying Creativity in CS High School Education - Criteria, Teaching Example and Evaluation. Proc. 7th Baltic Sea Conference on Computing Education Research, Koli Calling, Koli, 2008.
- [Ro08b] Romeike, R.: Where's my Challenge? The Forgotten Part of Problem Solving in Computer Science Education. Proc. 3rd ISSEP Intern. Conf. on Informatics in Secondary Schools - Evolution and perspectives, Torun, Polen, 2008.
- [ST05] Saunders, D.; Thagard, P.: Creativity in Computer Science. In (Kaufman, J. C.; Baer, J., Hrsg.): Creativity across domains: Faces of the muse, Mahwah, NJ, Lawrence Erlbaum Associates, 2005.
- [Sh02] Shneiderman, B.: Creativity support tools. In Commun. ACM 45(10), 2002; S. 116-120.
- [SFC05] Shneiderman, B.; Fischer, G.; Czerwinski, M.; Myers, B.; Resnick, M.: Creativity Support Tools. Workshop Report on Creativity Support Tools. National Science Foundation, Washington, DC, 2005.
- [SL91] Sternberg, R. J.; Lubart, T. I.: Creating Creative Minds. In Phi Delta Kappan 72(8), 1991; S. 608-614.
- [SKA08] Stribling, J.; Krohn, M.; Aguayo, D.: SCigen - An Automatic CS Paper Generator. 2008. <http://pdos.csail.mit.edu/scigen/> (22.05.2008).
- [Th05] Thomas, M.: Geschichten aus der Geschichte der Informatik. In LOG IN 136/137, 2005; S. 41-46.
- [Tr05] Trogmann, G.; Viehoff, J.: Code@ Art, Eine elementare Einführung in die Programmierung als künstlerische Praktik, Springer-Verlag, Wien, New York, 2005.
- [VDE03] VDE: VDE Studie "Young Professionals". Frankfurt am Main, VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V., 2003.
- [Wh08] Wheeler, D. A.: The most important software innovations. 2008. <http://www.dwheeler.com/innovation/innovation.html> (15.04.2008).