

Informatiktools – Gestaltung einer Plattform für Werkzeuge für den Informatikunterricht

Ralf Romeike

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

Abstract: Informatiktools sind Werkzeuge, die Informatikunterricht bei der Kompetenzvermittlung unterstützen. Vielfältige Werkzeuge, Dokumentationen dazu sowie Unterrichtsbeispiele sind bisher nur verstreut im Internet und anderen Publikationen erschienen. In diesem Paper werden die zugrunde liegenden Überlegungen zur Erstellung einer Plattform dargestellt.

1 Einleitung

Informatikunterricht orientiert sich häufig an seinen Werkzeugen, ebenso sind neuentwickelte Werkzeuge regelmäßig Gegenstand fachdidaktischer Diskussionen. Diese Werkzeuge können als Informatiktools¹ bezeichnet werden. Informatiker, Fachlehrer und Informatikdidaktiker nutzen regelmäßig ihre Fertigkeiten, in dem sie ein Tool kreieren, mit dem die Hoffnung verbunden ist, einen bestimmten Lerninhalt einfacher zu vermitteln. Vor allem Programmieren gilt als schwieriges Thema, sodass hierzu viele verschiedene Tools zur Einführung in die Programmierung existieren. Zu den frühen und bekanntesten Tools zählen *LOGO* [Pa80] und *Karel the robot* [Pa81], die bereits in den 80er Jahren eingesetzt wurden und Vorbild für verschiedene weitere Mini-Sprachen waren (bspw. Kara, Java-Hamster [RNH04; Bo05]). Moderne Programmierlernumgebungen reichen von Tools zur Erstellung von Animationen und Spielen (z.B. Scratch [Ma04] und Greenfoot [Kö08]), zum Gestalten von interaktiven 3D-Welten (z.B. Alice [CDP00]) über Tools zur Simulation von Schwarmverhalten (z.B. Starlogo [Re96]) und Robotersteuerung (Lego Mindstorms [LSI98]) hin zu Tools zur Programmierung von Smartphone-Apps (AppInventor [Ab09]). Neben diesen Tools, die das Erlernen des Programmierens vereinfachen bzw. einer breiten Masse zugänglich machen wollen, existieren viele weitere Tools für den Informatikunterricht, z.B. Editoren (z.B. UML, Java, Prolog-, HTML-Editor), Simulatoren (z.B. JFLAP, Jeliot, AtoCC, Digitalsimulatorsimulator), Lern- und Experimentierprogramme (z.B. Infotraffic, CrypTool) und viele

¹ Tool: Werkzeug, Instrumentarium, das man für eine bestimmte Aufgabe benötigt (Duden Fremdwörterbuch).

mehr. Darüber hinaus steht eine Vielzahl freier und kommerzieller Soft- und Hardware zur Verfügung, welche den Erwerb informatischer Kompetenzen unterstützt. Gemeinsam ist allen, dass sie in ihrer Gesamtheit schwer zu überblicken und nur mit viel Aufwand für den eigenen Unterrichtseinsatz zu evaluieren sind. Guzdial [Gu04] regt an, statt weitere Tools zu entwickeln, zunächst eine Einordnung und Bewertung der vorhandenen Tools vorzunehmen. Vereinzelt finden in der informatikdidaktischen Forschung solche Bewertungen statt (bspw. im empirischen Vergleich von BlueJ und Fujaba [Do07]). Es zeigt sich aber, mit welchem Aufwand eine solche Analyse verbunden ist. Da viele Tools erfolgreich eingesetzt werden, ist es bedauerlich, dass bisher keine umfassende Plattform zur Vorstellung und Diskussion der Werkzeuge und damit verbundenen Unterrichtsmaterialien existiert. Die Existenz verschiedener Websites mit kleineren Linksammlungen zu Tools unterstreicht den Bedarf.

Im Folgenden werden grundlegende Gedanken zur Entwicklung einer Informatiktools-Plattform beschrieben, welche die angesprochene Lücke füllen soll.² Hierzu werden in Kapitel 2 die Problemlage basierend auf fachdidaktischer Forschung zu Informatiktools erörtert und existierende Plattformen hinsichtlich ihrer Struktur und Probleme beschrieben, in Kapitel 3 Taxonomien und Strukturierungskonzepte analysiert und in Kapitel 4 resultierende Gestaltungsprinzipien abgeleitet. Im abschließenden Kapitel erfolgt eine Diskussion der Erkenntnisse.

2 Forschungsstand

Regelmäßiger Gegenstand fachdidaktischer Untersuchungen sind Werkzeuge zum Programmieren lernen, vor allem mit Fokus auf die Programmieranfängerausbildung an Hochschulen: 22% aller auf dem SIGCSE-Symposium zwischen 1984 und 2003 präsentierten Paper beschrieben Tools als Hilfsmittel für Studenten oder Lehrende [Va04]. Das Kreieren eines für Informatiklehre oder -unterricht interessanten Tools ist in der Regel motiviert durch Probleme, die Anfänger mit zu komplexen Programmiersprachen, Entwicklungsumgebungen, anderer komplexer Software oder Sachverhalten haben (vgl. [PSM07]), durch eine bestimmte didaktische Idee, mit dem Ziel, die Einführung eines Paradigmas zu unterstützen oder durch übergreifende Ziele, wie z.B. der Förderung von Kreativität (bspw. Scratch). Ein weiterer Grund für die Entwicklung von Werkzeugen sind institutionelle Probleme, z.B. zur Bewältigung von zu vielen Seminarteilnehmern (vgl. [Ko06]), zur Verringerung oder Vereinfachung des Arbeitsaufwands für den Lehrenden (z.B. durch Automatic Assessment Tools) oder aktuelle Forschungsinteressen (z.B. [St06]). Trotz des mitunter großen Forschungsaufwands zur Entwicklung von Tools, werden nur wenige außerhalb ihrer Heimatinstitution verwendet [PSM07] und dies, obwohl viele kostenfrei zum Download zu Verfügung stehen. Eine Ursache hierfür liegt in der Motivation: Entwickler verlassen sich häufig allein auf die Präsentation und den (teilweise individuellen) Mehrwert eines Tools. Lehrer als möglicherweise interes-

² Dieser Artikel bildet die Grundlage zur Umsetzung des dargestellten Vorhabens. Einzelheiten zur konkreten Umsetzung können aufgrund des beschränkten Platzes an dieser Stelle nicht dargestellt werden und sollen gemeinsam mit ersten Erfahrungen zu späterer Zeit publiziert werden.

sierte Zielgruppe haben aber kaum Gelegenheit, mit dem Werkzeug in Berührung zu kommen. Auch als interessierter Informatiklehrer ist es nahezu unmöglich, einen umfassenden Überblick über aktuelle Entwicklungen zu behalten.

Verschiedene Plattformen präsentieren bereits Inhalte und Werkzeuge für den Informatikunterricht und versuchen, als „Sammelstelle“ für informatikunterrichtsspezifische Inhalte zu fungieren.

HyperForum Informatik in der Schule

Das *HyperForum Informatik in der Schule*³ beinhaltet eine umfangreiche Linksammlung zu Werkzeugen, Unterrichtsmaterialien und fachwissenschaftlichen und fachdidaktischen Artikeln *ohne nachvollziehbare Struktur*. Die Plattform wird kaum noch *aktualisiert* und *gepflegt*; viele verlinkte Dokumente sind nicht mehr erreichbar.

Zoom-Wiki

Der Bereich Informatik des *Zoom-Wiki*⁴ ist ein umfangreiches Nachschlagewerk, vermischt allerdings Unterrichtsinhalte, Enzyklopädie und Tools. Die mögliche Mitarbeit der Informatiklehrer im Wiki beschränkt sich auf wenige Personen. Ebenso scheint es keine fachspezifische *Betreuung* des Wikis zu geben. Dies führt zu einer eigenartigen Themengliederung: „Ideen für den Unterricht“, „eher Grundschule, Sek I und ITG“, „eher Sek II“, „Programmieren im Wiki“, „Informatik am WBK“, „ohne Zuordnung“, „Thema Hardware“. In jüngerer Zeit gab es *kaum Aktualisierungen*.

SwissEduc

Eine bedeutende Rolle in der Präsentation zumeist selbst entwickelter Werkzeuge für den Informatikunterricht und im Verfügbarmachen von Unterrichtsmaterialien spielt die Schweizer Plattform *SwissEduc*, welche aus dem Bildungsserver EducETH entstand. EducETH wurde 1995 als Prototyp eines Education Servers entwickelt [SE11] und war eine der ersten Quellen für Unterrichtsmaterialien im WWW [AHS96]. Die präsentierten Tools und Materialien sind qualitativ hochwertig und sind fachthematisch strukturiert. Vorgestellt werden vor allem lokal entwickelte Tools und Unterrichtsvorschläge. Eine Möglichkeit, *eigene Materialien einzustellen* oder Materialien zu *kommentieren*, existiert nicht.

Weitere Plattformen

Neben den dargestellten überregionalen Angeboten bieten verschiedene Landesbildungsserver exemplarische Unterrichtseinheiten in *variierender Qualität*. Teilweise hochqualitative Quellen stellen ebenso private und schuleigene Websites von Informatiklehrern dar, welche zumeist schulspezifisch aktuelle Unterrichtsinhalte und -sequenzen anbieten.⁵ Zu weiteren nicht weiter aktualisierten und/oder inhaltlich sehr beschränkten

³ <http://www.hyfisch.de>.

⁴ <http://wiki.zum.de/Informatik>.

⁵ Bspw. „Werkzeuge für den Informatikunterricht“ von S. Spolwig (http://www.spolwig.de/sp-hu/spolwig/einfuehrung_02/dokumente/reinhold/start.htm), Linksammlung des Gymnasiums Neustadt (<http://gym-neu.dyndns.org/~hws/werkzeuge.html>), Informatik on Stick von T. Hempel (<http://www.tinohempel.de/info/info/loStick/index.html>).

Sammlungen gehören das life³-Projekt⁶ und Sammlungen von Schulen und Privatpersonen. Trotz dieser Fülle von Angeboten tut man sich schwer, fundierte Informationen für die Auswahl oder den Einsatz eines bestimmten Informatiktools zu finden. Viele Plattformen haben keinen *Nachschlagecharakter*, bieten *keine Diskussionsmöglichkeit*, setzen *regionale Schwerpunkte* und sind *nicht evaluiert* bzw. *referenziert*.

Zusammenfassend stellen sich folgende Punkte als problematisch dar:

- Fehlende Struktur
- Keine Aktualisierung und Pflege existierender Inhalte
- Betreuung der Plattform
- Inhalte sind nicht evaluiert bzw. referenziert
- Keine Diskussionsmöglichkeit
- Regionale Schwerpunkte der einzelnen Plattformen

3 Strukturierungs- und Bewertungskonzepte

Mit dem Vorhaben der Realisierung einer Plattform für Informatiktools stellt sich die Frage nach der Darstellung und Aufbereitung der Inhalte unter Berücksichtigung der beschriebenen Probleme. Ansätze können Taxonomien und Bewertungsversuche geben. In verschiedenen Aufsätzen wurde versucht, zumeist Programmierumgebungen (z.B. [GP05; KP05; AK09; Ne09]) oder Werkzeuge zur Programmvisualisierung (z.B. [My90; KPN08]) einzuordnen und zu bewerten. Mit dem Ziel, Ansätze für die Strukturierung und Bewertung von Informatiktools zu erhalten, werden die wesentlichen Aussagen im Folgenden dargestellt.

Taxonomien für Programmierlernumgebungen

Programmiersprachen werden vordergründig für Programmierer entwickelt. Die Komplexität der Werkzeuge stellt für Anfänger ein Problem dar, sich zurechtzufinden und Fehlermeldungen zu interpretieren. Kelleher und Pausch [KP05] kategorisieren Programmierumgebungen und -sprachen für Programmieranfänger in einer Baumstruktur, basierend auf der Frage nach der Motivation, Programmieren zu lernen: Um des Programmieren willens („Teaching Systems“) oder aus „höheren“, z.B. allgemein bildenden, Gründen („Empowering Systems“). Diese werden hinsichtlich ihrer Merkmale weiter untergliedert:

1. Teaching Systems (helfen korrekt Programmieren zu lernen)
 - a. Mechanics of Programming
 - i. Expressing Programs
 - ii. Structuring Programs
 - iii. Understanding Program Execution
 - b. Learning Support

⁶ <http://life.uni-paderborn.de/>

- i. Social Learning
 - ii. Providing a motivating context
 - 2. Empowering Systems (helfen, viele Ideen umzusetzen, ggf. auch unoptimal)
 - a. Mechanics of Programming
 - i. Code is too difficult
 - ii. Improve Programming languages
 - b. Activities enhanced by programming
 - i. Entertainment
 - ii. Education

Die Kategorien bieten interessante Anhaltspunkte, über den Einsatz eines Werkzeugs zu reflektieren. So herrscht zwar weitgehend Konsens, dass Programmierung einen wichtigen Aspekt der Schulinformatik darstellt, es ist aber strittig, ob korrektes Programmieren⁷ (1.) oder das Umsetzen möglichst vieler Ideen (2.) das primäre Unterrichtsziel darstellt. [KP05] ordnen verfügbare Tools je einer Kategorie zu. Dies ist allerdings aus zwei Gründen problematisch. Zum einen sind die Kategorien nicht trennscharf. Scratch kann bspw. verschiedenen Kategorien zugeordnet werden. Zum anderen bauen viele Tools auf den Ideen existierender Tools auf und führen auch hier zu einer Vermischung. In einem zusätzlichen Raster machen [KP05] Zuordnungen zu den Attributen eines Tools: *Style of programming, programming constructs, representation of code, construction of programs, support to understand programs, preventing syntax errors, designing accessible languages, support communication, choice of task*. Dies sind sinnvolle Attribute zur Charakterisierung von Programmierumgebungen. Es bleibt zu berücksichtigen, dass z.B. die Attributwerte von „choice of tasks“ wie *fun & motivating, useful* und *educational* subjektive Charakteristika eines Tools darstellen. [PSM07] analysieren Veröffentlichungen zu Programmierereinführungskursen und identifizieren bzgl. Tools folgende über Programmierumgebungen hinausgehende Kategorien:

1. Visualization Tools
2. Automated Assessment Tools
3. Programming Environments
 - a. Programming Support Tools
 - b. Microworlds
4. Other Tools

Alle genannten Kategorien sind auch für den Informatikunterricht relevant, decken aber nicht die Vielzahl der verschiedenartigen Tools ab. [GP06] präzisieren folgende (nicht trennscharfe) Kategorien für Programmierlernumgebungen:

- Microworlds (z.B. Logo, Karel, Alice)
- Visual programming environments (iconic and textual) (z.B. RoboLab, JPie)
- Flow Model Environments (z.B. RAPTOR)
- Object workbench environments (z.B. BlueJ, jGRASP)
- Algorithm realization environment (kinesthetic, multimedia, animation, graphics) (z.B. Lego Mindstorms)

⁷ bspw. unter Nutzung textueller Programmierung und effizientem Programmcode.

Es wird deutlich, dass allein im Bereich der Programmierwerkzeuge verschiedenste Kategorisierungen möglich sind. Eine Struktur lässt sich, auch vor dem Hintergrund der Verschiedenartigkeit der Werkzeuge, nicht ausmachen. Zur Charakterisierung eignen sich Stichworte, die z.B. in einer dazugehörigen Tag-Cloud visualisiert werden.

Bewertung von Lernumgebungen

Verschiedene Ansätze wurden entwickelt, Lernumgebungen der Informatik zu bewerten, insbesondere Programmierlernumgebungen. Gängige Methoden sind *anekdotische*, *analytische* und *empirische Überprüfung* (vgl. [GP05]). Häufig finden sich *anekdotische Angaben* zum Einsatz eines Werkzeugs. Zumeist stellt der Entwickler seine Eindrücke und Beobachtungen vom Einsatz des Tools im Unterricht dar (z.B. [Re07]). Zwar sind Anekdoten allein nicht überzeugend. Werden die Eindrücke allerdings von vielen geteilt, lassen sich hieraus Hypothesen für weitergehende Evaluation ableiten. Einer *Analyse* liegen Kriterien zugrunde, hinsichtlich derer ein Tool bewertet wird. Dies kann bspw. die Einordnung hinsichtlich o.g. Taxonomien sein, das zugrunde liegende Paradigma oder Kriterien hinsichtlich der Unterstützung des Lernens. Eine *empirische Überprüfung* erfordert das qualitative oder quantitative Auswerten von beobachteten oder erhobenen Daten im Prozess der Nutzung eines Tools, z.B. Zensuren, Fragebögen oder Interviews. Aufgrund der Nachvollziehbarkeit und Wiederholbarkeit solcher Studien stellen sie zwar eine solide Grundlage zur Bewertung von Tools dar, sind aber auch sehr aufwändig (vgl. [Do07]).

Die dargestellten Möglichkeiten der Bewertung scheinen umfassend zu sein, eignen sich in der Praxis aber nur bedingt aus folgenden Gründen. Ob sich ein Tool vielversprechend im Unterricht einsetzen lässt, hängt nicht nur von seiner didaktischen Idee, sondern von einer Vielzahl weiterer Aspekte ab: z.B. Stabilität, Verfügbarkeit von Unterrichtsmaterialien, Kosten, Erweiterbarkeit etc., die unterschiedlich stark zu gewichten sind. Dazu kommen individuelle Gründe wie Kursziele, bevorzugte Sprache und/oder bevorzugte Vorgehensweise. Einen möglichen Ausweg aus dem Dilemma stellt die Möglichkeit der Abgabe erfahrungsbasierter Bewertungen dar, welche in ihrer Gesamtheit wieder ein stimmiges Gesamtbild erzeugen könnten. Die dargestellten Kriterien können dafür als Anhaltspunkte dienen. Die Informatiktools-Plattform kann helfen, Studien zusammenzutragen und zusammenzufassen, darüber hinaus aber vor allem anekdotisch und analytisch zur Bewertung der präsentierten Tools beitragen. Die GI-Empfehlungen zu Bildungsstandards Informatik bieten einen Rahmen, durch ein Tool geförderte Kompetenzen zu kategorisieren. Vor dem Hintergrund des Potentials von Softwarewerkzeugen, Kreativität im Informatikunterricht zu fördern, bieten die Kriterien für kreativitätsunterstützende Softwarewerkzeuge im Informatikunterricht [Ro09] einen weiteren Ansatzpunkt zur Bewertung entsprechender Werkzeuge.

4 Konzeption und Umsetzung

Hinter *informatiktools.de* steckt die Idee, eine Plattform zu erstellen, welche die im WWW verteilten „Schätze“ zusammenführt und unter dem Dach der Werkzeuge er-

reichbar macht. [AHS96] charakterisierten bereits 1996 die Möglichkeiten des WWW für den Austausch von Unterrichtsmaterialien treffend: „WWW stellt eine erstklassige Plattform für den einfachen, kostengünstigen und schnellen Austausch von Unterrichtsmaterialien dar.“ Existierende Plattformen fokussieren allerdings überwiegend auf den Aspekt des „Zurverfügungstellens“. Ein weiterer genannter Aspekt wird auch dort nicht umgesetzt: „Die Benutzer können auch zu einzelnen Unterrichtsmaterialien Kommentare und Ergänzungen anbringen oder selbst Unterrichtsmaterialien auf den Server hinaufladen“. Tatsächlich hat sich mit der Entwicklung des *Web 2.0* gezeigt, wie viel Potenzial hinter der Beteiligung der Nutzer steckt. [AHS96] greifen die Metapher eines Strickwarenladens als Analogie für Fachserver auf:

„In einem Strickwarenladen kann man nicht nur Wolle und Stricknadeln kaufen. Man kann sich auch beraten lassen: Auf Wunsch werden individuelle Strickmuster angefertigt, Anpassungen oder Überarbeitungen bestehender Strickmuster vorgenommen. Der Kontakt Kunde/Geschäft ist viel enger, und es kann durchaus sein, daß eine Kundin ihre Strickwaren im Geschäft zum Verkauf anbietet. Genauso kann man sich speziell auf Schulen ausgerichtete Fachserver vorstellen.“

Aus diesem Grund soll die Plattform für *Informatiktools* es ermöglichen, Werkzeuge zu bewerten, Referenzen einzustellen, zu finden sowie zu diskutieren. Die Plattform wird betreut und gepflegt, ermöglicht aber jedem Interessierten, sowohl Entwicklern als auch Nutzern von Tools, sich an der Erstellung, Weiterentwicklung und Diskussion zu beteiligen. Der Bedarf an einer solchen Website wird immer wieder in Lehrerfortbildungen deutlich, in welchen Workshops zu Werkzeugen für den Informatikunterricht eine große Rolle spielen und stark besucht werden.

Wesentliche der erforderlichen Funktionen bietet ein moderiertes und gepflegtes Wiki: Entwickler und Nutzer eines Tools können sich an der Vorstellung bzw. Diskussion des Tools beteiligen. Wikis sind bekannt, werden von Informatiklehrern teilweise selbst eingesetzt und regelmäßig verwendet (Wikipedia). Zur besseren Erfassung der Strukturen wird auf die Möglichkeiten eines *Semantic-Wiki* [SBB06] zurückgegriffen. Initiierung und wissenschaftliche Begleitung obliegen einem Informatikdidaktiklehrstuhl und können somit auch im Rahmen der Informatiklehrerbildung durch Studierende regelmäßigen Input erhalten. Die in den GI-Empfehlungen zu Bildungsstandards Informatik beschriebenen Inhalts- und Prozessbereiche können neben den oben beschriebenen als Schlagwörter zur Charakterisierung der Tools herangezogen werden. Das Template eines *Informatiktools* strukturiert sich damit folgendermaßen: Beschreibung des Tools, Schlagwörter, Altersstufen, Voraussetzungen, Stärken/Schwächen (Kritik), Unterrichtsbeispiele, Literatur zum Tool, Wissenschaftliche Veröffentlichungen, weitere Referenzen/Quellen, Referenzen zu Institutionen/Schulen, welche das Tool verwenden, Verweise auf Ansprech- und Diskussionspartner, Nutzerbewertungen. Für jedes Tool/jede Wikiseite stellt ein Template die Struktur zur Verfügung. Entsprechend den vielen interessierten Nutzern der verschiedenen Tools scheint es realisierbar, zumindest erfahrungsbasiert eine Einschätzung der im Informatikunterricht eingesetzten Tools vorzunehmen.

5 Diskussion

Im WWW existieren verschiedene Versuche, Materialien und Ideen für den Informatikunterricht verfügbar zu machen. Probleme der Plattformen sind vor allem mangelnde Aktualität und Pflege und fehlende Möglichkeiten der Mitgestaltung. Die Praxis zeigt, dass Tools einen prägenden Einfluss auf den Informatikunterricht haben, das Beschaffen von Informationen zu Tools sich aber oft als problematisch erweist. Erfahrungen auf den Lehrerweiterbildungsveranstaltungen der Informatiktage in den einzelnen Bundesländern zeigen, dass bei den interessierten Lehrern ein hoher Informationsbedarf und großes Interesse hinsichtlich der Werkzeuge für den Informatikunterricht besteht. Eine Mehrzahl der angebotenen Workshops beschäftigt sich mit Werkzeugen und ist regelmäßig stark nachgefragt⁸. Das Problem von Lehrenden hinsichtlich neuer Tools, Aufgaben zu einem noch nicht vertrauten Tool entwerfen zu müssen, stellt oft eine Hemmschwelle für den Einsatz dar. Vermutlich liegt der Erfolg von Tools wie z.B. Kara auch darin, dass sie eine umfangreiche Sammlung an Aufgaben zur Verfügung stellen. Auch bei Visualisierungswerkzeugen zeigt sich, dass Visualisierung allein nicht ausreicht, Sachverhalte zu verstehen, sondern erst das aktive Auseinandersetzen, z.B. indem Fragen beantwortet werden, Vermutungen aufgestellt werden oder experimentiert wird, ein besseres Lernergebnis erzielt [HDS02].

Sollen (noch mehr) Tools im Informatikunterricht eingesetzt werden? Sicherlich nicht, im Informatikunterricht hat der Einsatz von Software bereits einen hohen Stellenwert. Die Praxis der Lehrerweiterbildungen zeigt, dass Lehrer regelmäßig mit dem Angebot an verschiedenen Tools überfordert sind und z.B. aufgrund von Unkenntnis oder wenig Zeit Tools einsetzen, die in ihrer didaktischen Idee überholt oder in ihrer Komplexität durch deutlich einfachere Tools ersetzbar sind. So setzen z.B. viele Lehrer *GIMP* oder *Photoshop* zur Bildbearbeitung in der Sekundarstufe 1 ein, beschränken sich dabei aber auf Bildbearbeitungswerkzeuge, die nicht die Komplexität der Software benötigen und z.B. einfacher in *Pixlr* [Pi11] verfügbar und damit für Schüler altersgemäßer sind.

Ein Mehrwert der Plattform ergibt sich damit vordergründig aus dem praktischen Nutzen: Die Vielfalt an Informatiktools und korrespondierenden Unterrichtsentwürfen kann Lehrern besser zugänglich gemacht werden. Dies gilt auch für Tools, die in den Fachdidaktiken der Hochschulen entwickelt wurden. Häufig werden diese zwar publiziert und in Fachzeitschriften veröffentlicht, erreichen aber die Unterrichtswirklichkeit nicht, da nur wenige Fachlehrer Fachzeitschriften lesen. Als Beispiele hierfür können *InfoTraffic* [AH07], *Leo* [US01] und *Puck* (außerhalb Thüringens) [Ko06] genannt werden. Darüber hinaus kann die Plattform Ideen zu vielen interessanten Unterrichtsmethoden durch den Einsatz von Tools liefern, z.B. das gegenseitigen Stellen und Lösen von Quiz- bzw. Übungsaufgaben durch Lerngruppen [DHL08] oder gegenseitiges Begutachten innerhalb von Lerngruppen [Ko06].

Eine unüberschaubare Anzahl an Informatiktools wurde entwickelt, von denen Informatikunterricht profitieren kann. Leider ist wenig über den tatsächlichen Einfluss dieser

⁸ So stehen bspw. auf dem aktuellen Berlin-Brandenburger Informatiktag in 7 der 13 angebotenen Workshops Werkzeuge im Mittelpunkt, beim SH-HILL sind es sogar 11 von 16.

Tools bekannt. Aus wissenschaftlichen Gesichtspunkten bietet die Plattform die Möglichkeiten, Entwicklungs- und Diskussionsprozesse informatikdidaktischer Forschung (Werkzeuge) und Praxis sichtbar zu machen. Zugriffszahlen, Diskussionsaktivitäten und Publikationsmöglichkeiten für neue Werkzeuge sollen für den wissenschaftlichen Diskurs herangezogen werden.

Die Eröffnung der Plattform ist zur INFOS 2011 geplant.

Literatur

- [Ab09] Abelson, H.: App Inventor for Android. In Google Research Blog, July 31, 2009.
- [AHS96] Ackermann, S.; Hartmann, W.; Stumm, M.: Unterrichtsmaterial über WWW. In LOG IN 16(1), 1996; S. 61-68.
- [AK09] Ali, A.; Kohun, F. G.: Considerations for selecting a programming language to teach perspective teachers. Proc. Alice Symposium, 2009.
- [AH07] Arnold, R.; Hartmann, W.: Pragmatische Empfehlungen zur Entwicklung von interaktiven Lernumgebungen. Proc. INFOS 2007, 12. GI-Fachtagung Informatik und Schule, 2007.
- [Bo05] Boles, D.: Spielerisches Erlernen der Programmierung mit dem Java-Hamster-Modell. In Lecture Notes in Informatics (LNI)-Proceedings 60, 2005; S. 243-252.
- [CDP00] Cooper, S.; Dann, W.; Pausch, R.: Alice: a 3-D tool for introductory programming concepts. Proceedings of the fifth annual CCSC northeastern conference on The journal of computing in small colleges. Ramapo College of New Jersey, Mahwah, New Jersey, United States, Consortium for Computing Sciences in Colleges, 2000.
- [DHL08] Denny, P.; Hamer, J.; Luxton-Reilly, A.; Purchase, H.: PeerWise. Proceedings of the 8th International Conference on Computing Education Research. Koli, Finland, ACM, 2008.
- [Do07] Dohmen, M.: Empirisches Untersuchungsdesign zum Medieneinsatz im objektorientierten Informatikunterricht Proc. Didaktik der Informatik in Theorie und Praxis, INFOS 2007, 12. GI-Fachtagung Informatik und Schule, Siegen, GI, 2007.
- [GP05] Gross, P.; Powers, K.: Evaluating assessments of novice programming environments. Proc., ACM, 2005.
- [GP06] Gross, P.; Powers, K.: Work in progress-a meta-study of software tools for introductory programming. Proc., IEEE, 2006.
- [Gu04] Guzdial, M.: Programming environments for novices. In Computer science education research, 2004; S. 127-154.
- [HDS02] Hundhausen, C. D.; Douglas, S. A.; Stasko, J. T.: A Meta-Study of Algorithm Visualization Effectiveness. In Journal of Visual Languages & Computing 13(3), 2002; S. 259-290.
- [KPN08] Kasurinen, J.; Purmonen, M.; Nikula, U.: A Study of Visualization in Introductory Programming. Proc. 20th annual Meeting of Psychology of Programming Interest Group, Lancaster, UK, 2008.
- [KP05] Kelleher, C.; Pausch, R.: Lowering the barriers to programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. In ACM Computing Surveys 37(2), 2005; S. 83-137.
- [Ko06] Kohl, L.: Puck-eine visuelle Programmiersprache für die Schule. In *informatica didactica* 8(7), 2006.
- [Kö08] Kölling, M.: Greenfoot: a highly graphical ide for learning object-oriented programming. Proc., ACM, 2008.
- [Ko06] Koubek, J.: Das Gutachtersystem als asynchrone nicht-kooperative Lernumgebung. Proc. Grundfragen multimedialen Lehrens und Lernens, 2006.

- [LSI98] LEGO_SYSTEMS_INC.: Lego Mindstorms Robotics Invention System. 1998. <http://mindstorms.lego.com>
- [Ma04] Maloney, B., Kafai, Rusk, Silverman, Resnick: Scratch: A Sneak Preview. In IEEE Computer Society, 2004; S. 104 - 109.
- [My90] Myers, B. A.: Taxonomies of visual programming and program visualization. In J. Vis. Lang. Comput. 1(1), 1990; S. 97-123.
- [NRA02] Naps, T. L.; Rößling, G.; Almstrum, V.; Dann, W.; Fleischer, R.; Hundhausen, C.; Korhonen, A.; Malmi, L.; McNally, M.; Rodger, S.; Velazquez-Iturbide, J. A.: Exploring the role of visualization and engagement in computer science education. Working group reports from ITiCSE on Innovation and technology in computer science education. Aarhus, Denmark, ACM, 2002.
- [Ne09] Nesbit, T.: The Teaching of Introductory Programming: Issues of Context. In, 2009.
- [Pa80] Papert, S.: Mindstorms : children, computers, and powerful ideas. Basic Books, New York, 1980.
- [Pa81] Pattis, R. E.: Karel the robot: a gentle introduction to the art of programming. John Wiley & Sons, 1981.
- [PSM07] Pears, A.; Seidman, S.; Malmi, L.; Mannila, L.; Adams, E.; Bennedsen, J.; Devlin, M.; Paterson, J.: A survey of literature on the teaching of introductory programming. Working group reports on ITiCSE on Innovation and technology in computer science education. Dundee, Scotland, ACM, 2007.
- [Pi11] pixlr.com: pixlr photo editing services. 2011. <http://pixlr.com> (15.01.2011).
- [RNH04] Reichert, R.; Nievergelt, J.; Hartmann, W.: Programmieren mit Kara: ein spielerischer Zugang zur Informatik. Springer, 2004.
- [Re96] Resnick, M.: StarLogo: an environment for decentralized modeling and decentralized thinking. Proc., ACM, 1996.
- [Re07] Resnick, M.: Sowing the Seeds for a More Creative Society. Proc. Learning & Leading with Technology, International Society for Technology in Education (ISTE), 2007.
- [Ro09] Romeike, R.: Softwaretools für kreatives Lernen im Informatikunterricht. In Tagungsband zur GI-Fachtagung Informatik und Schule INFOS Berlin, 2009.
- [SBB06] Schaffert, S.; Bischof, D.; Bürger, T.; Gruber, A.; Hilzensauer, W.: Learning with semantic wikis. Proc. First Workshop on Semantic Wikis -- From Wiki To Semantics, Citeseer, 2006.
- [St06] Stechert, P.: Unterrichtsmodellentwicklung zur Förderung des Informatiksystemverständnisses mit Entwurfsmustern. Proc. Workshop der GI-Fachgruppe „Didaktik der Informatik“, 2006.
- [SE11] SwissEduc: Informationen über SwissEduc. 2011. <http://www.swisseduc.ch/about/geschichte/index.html>
- [US01] Universität Siegen: LEO - Lernumgebung für objektorientiertes Modellieren im Informatikunterricht. 2001. <http://www.die.informatik.uni-siegen.de/pgleo> (15.01.2011).
- [Va04] Valentine, D. W.: CS educational research: a meta-analysis of SIGCSE technical symposium proceedings. In SIGCSE Bull. 36(1), 2004; S. 255-259.