

Sei selbst der Igel! – Umsetzung ausgewählter Elemente der phänomenorientierten Informatik

Daniel Siebrecht¹

Abstract: In diesem Beitrag wird ein Konzept vorgestellt, das Bewegung bewusst als informatischen Unterrichtsgegenstand auffasst und ihn mit der Algorithmik verknüpft. Dadurch bieten sich nicht nur die klassischen Formen der symbolischen oder ikonischen Visualisierung von Problemstellungen an: Naheliegender kann ebenso ein enaktiver Zugang gewonnen werden. Die Lernenden führen im vorliegenden Konzept die Algorithmen selbst durch, sodass sich schließlich das Ausführen algorithmisierter Bewegung als ergänzende Alternative zur gewohnten Implementation am Informatiksystem herausstellt.

Keywords: Implementation; Bewegung; Algorithmik; enaktiv

1 Bewegung als informatischer Unterrichtsgegenstand

Üblicherweise wird Bewegung im Zusammenhang mit Informatikunterricht *methodisch* assoziiert. Namhafte Arbeiten in der internationalen Informatikdidaktik legen nahe, dass sich die enaktive Erarbeitung informatischer Konzepte in Form größerer Motivation und ergiebigeren Lernerfolgs rentiert (z. B. [BWF15]). Rollenspiele und Aktivitäten sind eindeutig einer *methodischen* Aufwertung des Informatikunterrichts zuzurechnen. Ein Informatikunterricht, der aber beispielsweise die Straßenüberquerung oder den „Tanz der Bienen“ analysiert und modelliert, versteht Bewegung als *Sachgegenstand*.

Als wesentliche Bezugsthematik ist die Algorithmik der Bewegung zuzuordnen: Bei der Notation von Bewegungsabläufen (Kinetographie) werden algorithmische Strukturen verwendet. Verkehrsnetze – etwa auch Labyrinth – bergen vornehmlich „Weg-Problemstellungen“. Hüpfkästchen und schrittmusterartiges Bewegen oder gar Tanzen bieten hingegen keine direkte Problemstellung, setzen sich aber aus Strukturen algorithmischer Elemente zusammen. Nicht kinetisch bedingte Algorithmen wären hingegen beispielsweise der Euklid-Algorithmus, String-Matching-Algorithmen oder Algorithmen zur Datenbankanalyse. Somit lässt sich eine Schnittmenge zwischen Algorithmik und Bewegung (Kinalgorithmik) finden, „die sich mit der Untersuchung, Lösung und Umsetzung von kinetographischen Problemstellungen auseinandersetzt“ ([Si15, S. 27]). Zieldimension ist das Analysieren und Modellieren kinetischer Strukturen und Probleme, deren Lösung perspektivisch zu einer Ausführung gebracht werden soll, sei es nun auf einem Informatiksystem oder durch einen Menschen.

¹ Bergische Universität Wuppertal, Didaktik der Informatik – siebrecht@uni-wuppertal.de

2 Alternative Implementation durch Bewegung

Der Vorschlag ist also eine Alternative zur „klassischen“ Implementation am Informatiksystem in Form menschlicher Bewegungsausführung (anthropokinetische Implementationsalternative). Dabei mögen virtuelle Systeme, die vornehmlich den visuellen Lernkanal bedienen, in den Hintergrund treten, um durch Haptik und Motorik enaktiv Lernprozesse zu initiieren und zu fördern, welche zur Aneignung informatischer Kompetenzen auf *konzeptueller* Ebene befähigen sollen. Statt einen Editor zu öffnen, um Quelltext zu verfassen, und diesen anschließend kompilieren zu *lassen* oder interpretieren zu *lassen*, böte sich den Schülerinnen und Schülern die Möglichkeit, nach einer zeiteffizienten Notation der Programmsequenz mit Stift und Papier dieses Programm unmittelbar im Anschluss an die Entwicklung *selbst* als Programm ausführende Instanz zu übersetzen und zwar in eigene Körperbewegungen. Dies kann aber nur umgesetzt werden, wenn der modellierte Gegenstand kinalgorithmischer Natur ist. „Die eigentliche Sprache, die es dabei zu lernen gilt, ist [nämlich] weniger eine beliebige Programmiersprache, sondern eher die Algorithmik selbst“ ([Si15, S. 28]). DISSMANN fordert in diesem Sinne eine zielgerichtete Handlungsfähigkeit „im Umgang mit den programmtechnischen Konzepten“ ([Di03, S. 226]); der „Einsatz der natürlichen Fähigkeiten“ ([Di03, S. 228]) sei dabei besonders wichtig; er verweist darauf, dass die Lernenden bei einer Informatiksystem gestützten Implementation mit dem „trial and error“-Prinzip auf die Rückmeldungen des Compilers antworten. Bei der angedachten Alternative zeigt sich ein eindeutiger Vorteil darin, dass die Lernenden dazu angehalten wären, sich *selbst* hinsichtlich einer fehlerbehaftet notierten Programmsequenz korrigieren zu müssen – ohne dabei die Schuld auf „Dritte“ schieben zu können.

Literatur

- [BWF15] Bell, T.; Witten, I. H.; Fellows, M.: Teaching Computer Science Unplugged. Teachers' Edition. An enrichment and extension programme for primary-aged students. Computer Science Unplugged, 2015.
- [Di03] Dißmann, S.: Handlungsorientiertes Erlernen von Programmkonstruktionen anhand von Rollenspielen. In (Hubwieser, P., Hrsg.): Informatische Fachkonzepte im Unterricht, INFOS 2003, 10. GI-Fachtagung Informatik und Schule, 17.-19. September 2003 in Garching bei München. Bd. 32. Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn, S. 225–235, 2003.
- [Si15] Siebrecht, D.: Einführung algorithmischer Elemente unter kinetographischen Aspekten, 2015, URL: <http://bscw.ham.nw.schule.de/pub/bscw.cgi/6586882>, Stand: 17.08.2017.