von Andreas Bollin

COOLe Informatik

1. EINE FRAGE DES VERSTEHENS

Wie viele Code-Zeilen sind in einer Waschmaschine oder in einem Auto? Tausende oder mehr? Schon im Jahr 2012 waren in einem durchschnittlichen Auto von General Motors mehr als 100 Millionen Code-Zeilen. Das ist erstaunlich komplex. Der große Einfluss von Technologie und Komplexität auf den Alltag unserer heutigen Gesellschaft wirft auch die Frage auf, was wir (als Forschende und Lehrende) tun können, damit die Gesellschaft auf aktuelle und zukünftige Anforderungen, Probleme und Chancen vorbereit ist.

Kristen Nygaard (1926 – 2002), ein norwegischer Informatiker und Programmiersprachen-Pionier, merkte einmal an, dass das Erstellen eines Modells in SIMULA für Entwickler nützlicher ist, als das Ausführen des Programms SIMULA selbst. Diese Überzeugung hat zu dem berühmten Aphorismus geführt: „Zu beschreiben bedeutet zu verstehen. Zu programmieren bedeutet zu beschreiben. Daraus folgt – zu programmieren bedeutet zu verstehen." Wenn man über diese Aussage Nygaards nachdenkt, wird klar, dass jemand, der eine Sache (in diesem Fall eine formale Sprache) beschreibt, sie wahrscheinlich umfassend verstanden hat. Die Kenntnis einer Programmiersprache reicht jedoch nicht aus, sondern es ist eine besondere Art des Denkens erforderlich.

2. EINE FRAGE DES DENKENS

Jeannette Wing zeigt in einem Artikel in Communications of the ACM einen Weg auf, wie die oben angeführten Herausforderungen gemeistert werden können [Wing 06]. Ihre Antwort lautet: Machen wir Computational Thinking (CT) selbstverständlich, machen wir es zu einer vierten Kulturtechnik. Sie versteht CT als eine Reihe von Gedankenprozessen, die an der Formulierung und Lösung von Problemen beteiligt sind. Ziel ist es, sie so darzustellen, dass sie von einem Computer erfolgreich ausgeführt werden können.

Die gute Nachricht ist, dass sich seit dem Jahr 2006 einige Forschungsarbeiten mit der Frage nach einer Einführung des CT in den Bildungsbereich beschäftigt haben. In Abbildung 1 wird die Zunahme der Forschungsarbeiten in diesem Bereich grafisch dargestellt. Die schlechte Nachricht ist, dass dieses Thema vor allem in der Primar- und Sekundarstufe nur langsam die notwendige Aufmerksamkeit erhält. Das liegt teilweise an der Art und Weise, wie CT an Schulen unterrichtet werden soll.

3. EINE FRAGE DES LEHRENS

Es gibt verschiedene Gruppen, die unterschiedliche Ansichten über die Art und Weise haben, wie CT an Schulen zu unterrichten ist. Zuerst gibt es einmal jene, die es eng gefasst betrachten und das mit Coding Skills verbinden [Seiter 13]. Dann sind da jene, die es als Reihe von Problemlösungskompetenzen betrachten [Dierbach 11]. Und schließlich gibt es die Zwischenposition, wo Coding eine wichtige Rolle spielt, aber erst nachdem spielerisch viele Erfahrungen mit Fertigkeiten gesammelt wurden, die mit CT in Zusammenhang stehen. [Cole 15].

Nun ist Coding sicher wichtig, aber CT erfordert mehr. Es erfordert Denken auf vielen Abstraktionsniveaus, das Umgehen mit Informationen und – am wichtigsten – das Arbeiten mit Modellen in unterschiedlichen Sprachen und Umgebungen. Hier stoßen alle drei Gruppen an ihre Grenzen, daher verfolgen wir auf der Alpen-Adria-Universität Klagenfurt einen ganzheitlicheren Ansatz, den wir COOL Informatics nennen. COOL steht für Cooperative Open Learning, für Computer-Supported Open Learning und auch für cool (motivierend, erfolgreich). Dieser Ansatz basiert auf vier neurodidaktischen Prinzipien: Entdeckung, Zusammenarbeit, Individualität und Aktivität. Im Sinne von Wing werden im CT Verhalten und Fertigkeiten trainiert, wobei sie multidisziplinär in unterschiedliche Schulfächer eingebettet sind, und zwar mit Schwerpunkt auf Sprachen und Modellen.

Begleitende Studien haben gezeigt, dass der COOL Ansatz sehr erfolgreich ist. Wir haben ihn bereits in Schulen der Primar- und Sekundarstufe verwendet und unsere Informatikwerkstatt ist für die Öffentlichkeit zugänglich. Das Konzept wurde schon in unterschiedlichem Rahmen verwendet, von 5-jährigen SchülerInnen bis zu 70-jährigen SeniorInnen. Noch wichtiger ist aber, dass es uns gelungen ist, mit dieser Lehrmethode spielerisch das Niveau für Technologieverständnis und Problemlösungskompetenzen anzuheben. Unsere Offenheit für alle Altersstufen ermöglicht es uns, eine Vision von Jeannette Wing zu erfüllen, dass CT zur Realität wird, wenn es so sehr zu einem fixen Bestandteil des menschlichen Strebens geworden ist, dass es als explizite Philosophie verschwindet. <<

REFERENZEN:

[Wing 06] Computational Thinking. J. Wing. CACM 49, 3 (2006).

[Dierbach 11] A model for piloting pathways for computational thinking in general education. C. Dierbach, et.al. SIGCSE '11. ACM, 257-262 NY (2011).[Seite 13]

Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. L. Seiter, et al. 9th international ACM conference on International Computing Education Research. 59-66 NY (2013).

[Cole 15] On Pre-requisite Skills for Universal Computational Thinking Education. E. Cole. ICER'15, Omaha, USA (2015).