



TPA-Modell

Urs Meier

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 28, 2019

TPA-Modell

Vom Denken zum Handeln in der informatischen Bildung

Autor

Abstract: Durch die Einführung des neuen Fachs Medien und Informatik im Rahmen des Lehrplan 21 an Deutschschweizer Volksschulen, halten u.a. informatische und medienpädagogische Inhalte Einzug in den Unterricht ab Kindergartenstufe. Lehrpersonen aller Schulstufen sind daher gefordert, ihre eigenen Kompetenzen zu weiterzuentwickeln, um ihre Schüler*innen im Lernprozess zu unterstützen. Insbesondere der Programmierunterricht benötigt viel Handlungskompetenzen. Um die Abläufe eines Problemprozesses beim Programmieren (Computational Thinking) für Lehrpersonen an Weiterbildungen verständlich aufzuzeigen, hat der Autor ein Modell entwickelt. Das sogenannte TPA-Modell (Thinking, Processing, Acting) zeigt die grundlegenden Arbeitsschritte beim Lösen eines Problems aus dem Alltag bis hin zur Verwirklichung in einem physikalischen Modell, wie z.B. einem Roboter oder Microboard. In einem weiteren Entwicklungsschritt wurde das TPA-Modell bei praktischen Arbeiten eingesetzt, um problembasierte Aufgaben zu reduzieren und die grundlegenden Programmierkonzepten zu vermitteln.

Keywords: Computational Thinking, CS unplugged, Algorithmen, Programmieren, Physical Computing, Problembasierte Aufgaben

1 Einleitung

Aktuell wird an Deutschschweizer Volksschulen mit Inkraftsetzung des Lehrplans 21 das Fach Medien und Informatik ab Kindergartenstufe eingeführt. Inhaltlich wird dabei zwischen medienpädagogischen, informatischen Inhalten sowie Anwendungen unterschieden. Im Bereich Informatik sind Begriffe wie Computer Science Unplugged (folgend als „unplugged“ bezeichnet), Algorithmen, Programmieren, etc. neu. In Weiterbildungen bei Lehrpersonen hat der Autor festgestellt, dass die genannten Begrifflichkeiten oft nicht verstanden sowie falsch angewendet werden. Aus diesem Grund hat der Autor ein Modell entwickelt, welches die Begriffe richtig verortet und einen Überblick vermittelt. Das Modell zeigt erstens das Denken (Thinking oder Computational Thinking) auf, welches zu einem Programm (Processing) führt und in einem dritten Schritt im Physikalischen (Acting) umgesetzt wird. Das TPA-Modell zeigt im Überblick die grundlegenden Arbeitsschritte beim Lösen eines Problems aus dem Alltag bis hin zum Verwirklichen in einem physikalischen Modell, wie einem Roboter oder Microboard.

2 TPA-Modell

2.1 Thinking

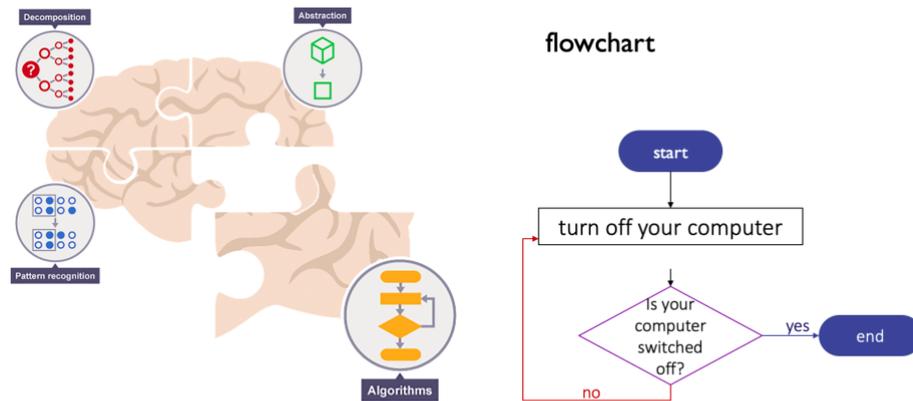
Das Denken steht im Modell an erster Stelle. Dabei geht es darum, ein Problem aus der analogen Welt so zu modellieren, damit es von einem Computer gelöst werden kann. Im Paper *New frameworks for studying and assessing the development of computational thinking* haben Brennan und Resnik (2012) folgendes Zitat für Computational Thinking verwendet: *“the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent”*. [1, p. 2]

Das Zitat setzt den Schwerpunkt auf einen Denkprozess, der vorgängig ein Problem analysiert und aufbereitet, damit ein Computer verarbeitet werden kann.

Das vorliegende TPA-Modell stützt sich bildlich wie inhaltlich aber auf das Computational-Thinking-Modell aus England, das nach der Bildungsinitiative im Jahre 2014 von BBC [2] veröffentlicht wurde. Die Definition für Computational Thinking von BBC ist weit verbreitet und unter anderem stützt sich Google im Education-Programm auch auf die gleichen vier Begriffe [3].

Diese sind:

- **Decomposition:**
Grosse Problemstellungen auf kleinere lösbare reduzieren
- **Pattern Recognition:**
Erkennen von Mustern und Regelmässigkeiten in Daten
- **Abstraction:**
Unnötige Informationen finden und auf wichtiges Reduzieren.
- **Algorithm:**
Entwickeln von Anweisungen, die das Problem oder ähnliche Probleme lösen.



Thinking

Abbildung 1 TPA-Modell: Thinking

Im ersten Teil Thinking wird ein Problem so aufbereitet, dass eine Abfolge von Anweisungen und Entscheidungen ausgearbeitet wird, die einem Computer übergeben werden können. Als Darstellung dazu eignet sich ein Flussdiagramm oder Pseudocode.

2.2 Processing

Die Erkenntnisse aus dem Denkprozess kann nun an einen Computer übergeben werden, der sie ausführen kann. Die Kommunikation vom Menschen an den Computer wird programmieren genannt. Hromkovič (2009) [4, p. 34] bezeichnet das Programmieren als eine *Sprache zur Kommunikation mit dem Rechner*.

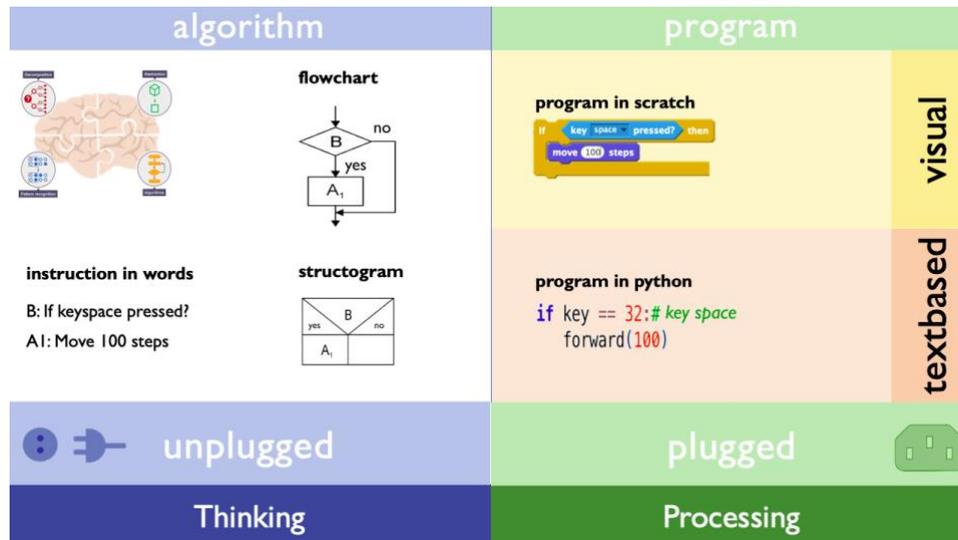


Abbildung 2 TPA-Modell – Processing

Das TPA-Modell zeigt einen Überblick der Tätigkeiten ohne Computer (unplugged) und mit dem Computer (plugged) auf. Des Weiteren wird die Unterscheidung von Algorithmen und Programmen deutlich und verständlich.

Je nach Aufgabe und Unterrichtsziel eignet sich eine textbasierte, beziehungsweise eine visuelle Programmiersprache besser für die Anweisungen an den Computer.

Damit das Verständnis für die Arbeitsweise von Computern deutlich wird, zeigt das TPA-Modell in Abbildung 3 eine zusätzliche Ebene, die das Ausführen eines Programmes mit der Recheneinheit darstellt. Dazu gehört das Kompilieren und das Übergeben des Maschinen-Codes an die kleinste Einheit, einer Schaltung mit einem Transistor.

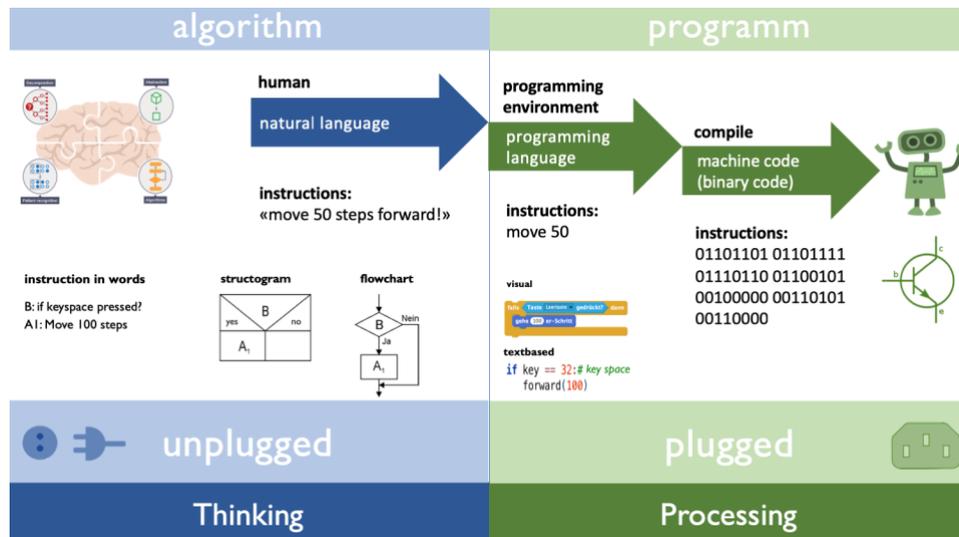


Abbildung 3 TPA-Modell - Processing and Compiling

2.3 Acting

Das Acting zeigt das Programm in Aktion bzw. Handlung und dient als Rückmeldung, ob das eingangs gestellte Problem gelöst wird. Das EVA-Prinzip [5] wird hier vertikal dargestellt und zeigt die physikalische Eingabe mit Tastatur oder Sensor, der Verarbeitung in einem Prozessor sowie der Ausgabe auf einem Bildschirm, bzw. einem Aktor.

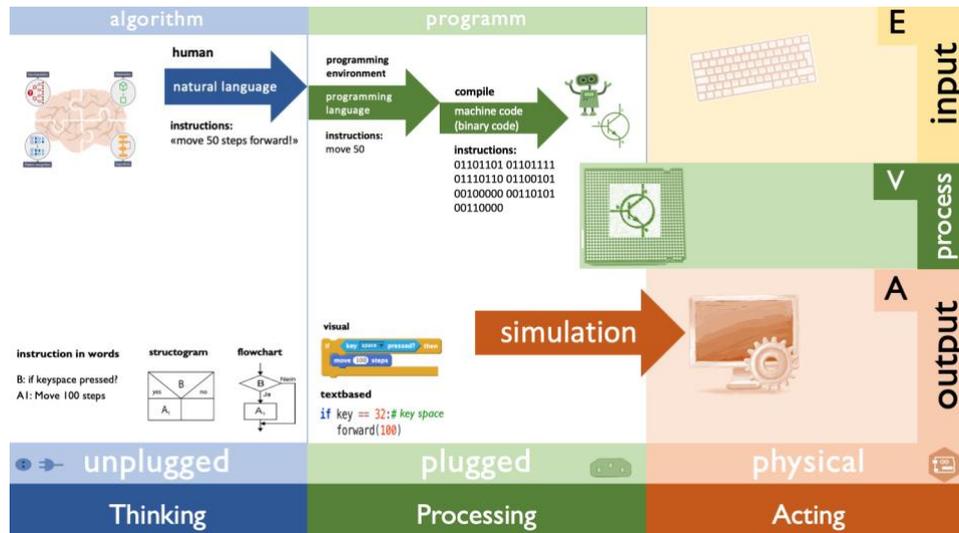


Abbildung 4 TPA-Modell – Acting

Das TPA-Modell zeigt im Überblick auf, wie vom Denkprozess bis hin zur Anzeige auf dem Bildschirm viele kleine Schritte nötig sind. Weiter wird deutlich, wie komplex die Entwicklung von Software ist. Für die Fehlersuche ist der Überblick hilfreich und zeigt auf, wo nach möglichen Fehlerquellen gesucht werden kann.

3 Umsetzung

Das TPA-Modell wird in der Aus- und Weiterbildung im Rahmen von Semestermodulen oder CAS Medien und Informatik für Lehrpersonen eingesetzt. Es hilft den Teilnehmenden Begriffe richtig zu verorten und einen Überblick zu gewinnen. Als Schwerpunkt wird das TPA-Modell bei problembasierten Aufgaben eingesetzt und dabei bewusst bei kleinen und einfachen Problemen angewendet und so grundlegenden Programmierkonzepte vermittelt.

Ziele dabei sind:

- Algorithmisches Denken fördern
- Ein Problem in kleine lösbare Probleme reduzieren
- Lösungen für Probleme in Worten und mit Diagrammen darstellen
- Umsetzung von Algorithmen in Programmen
- Testen der angedachten Lösungsstrategien in Programmen

3.1 Aus- und Weiterbildung

Das TPA-Modell eignet sich sehr gut, damit die Begriffe der Informatik korrekt eingesetzt und verwendet werden. Das TPA-Modell zeigt einerseits den Lehrpersonen die Komplexität des Informatik-, bzw. Programmierunterrichts, hilft aber andererseits mit dem Überblick auch Ängste und Unsicherheiten abzubauen sowie Vertrauen zu schaffen.

Weiter soll das TPA-Modell aufzeigen, wie fachdidaktische Prinzipien anhand der Darstellung aufgezeigt und in den Schritten verortet werden können. Es wird unter anderem dargelegt, wie man Programmieren vermitteln kann. Ein Schwerpunkt ist die Förderung des algorithmischen Denkens und das Lösen von Problemen.

Konkret wird das TPA-Modell bisher in Weiterbildungen herangezogen, um aus einer Problemstellung, beispielsweise der Spielmechanik, kleine Teilprobleme heraus zu schälen und zu vermitteln.

Ausgehend von einem Rennspiel können beispielsweise folgenden Problemstellungen ermittelt werden:

- Steuerung eines Autos mit Tasten
- Immer auf der Bahn bleiben
- Runden zählen
- Tempo messen
- Aus der Bahn schleudern
- Akkustand bei Elektro-Formel 1 anzeigen
- Hindernisse einbauen
- ...

Als Beispiel für eine konkret Umsetzung wird hier das Teilproblem eines Akkustands dargelegt. Das Teilproblem wird reduziert auf das Zählen mit einer Variablen. Damit das Problem analysiert werden kann, müssen in einem ersten Schritt Fragen gestellt werden.

- Welche Informationen benötige ich?
- Wie zähle ich zurück?
- Wo starte ich?
- Welche Schritte nehme ich?
- Wie benenne ich die Variable?
- ...

In den nächsten Schritten wird das Teilproblem in Worten (Pseudo-Code) als Flussdiagramm und als blockbasiertes Programm dargestellt.

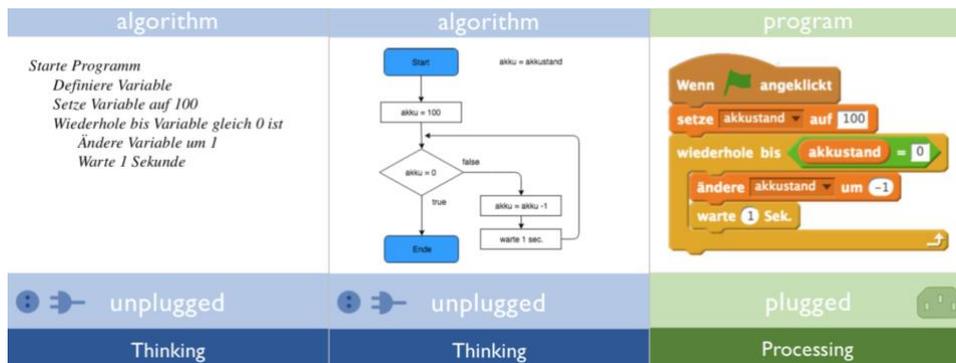


Abbildung 5 Problembasierte Aufgaben - Teilschritte

Als weitere Anwendung wurde das vorgestellte TPA-Modell für problembasierte Aufgabenstellungen in der Ausbildung angehender Lehrer*innen für die Sekundarstufe 1 während eines Semestermoduls eingesetzt. Exemplarisch wird folgend eine gelungene Semesterarbeit von Berger [6] aufgezeigt. Der vorliegende Ausschnitt zeigt auf, wie ein Problem analysiert und schrittweise in ein Programm umgesetzt wird.

Zufälliges Auftauchen

Wie erscheint der Käfer unvorhersehbar am linken Spielfeldrand?

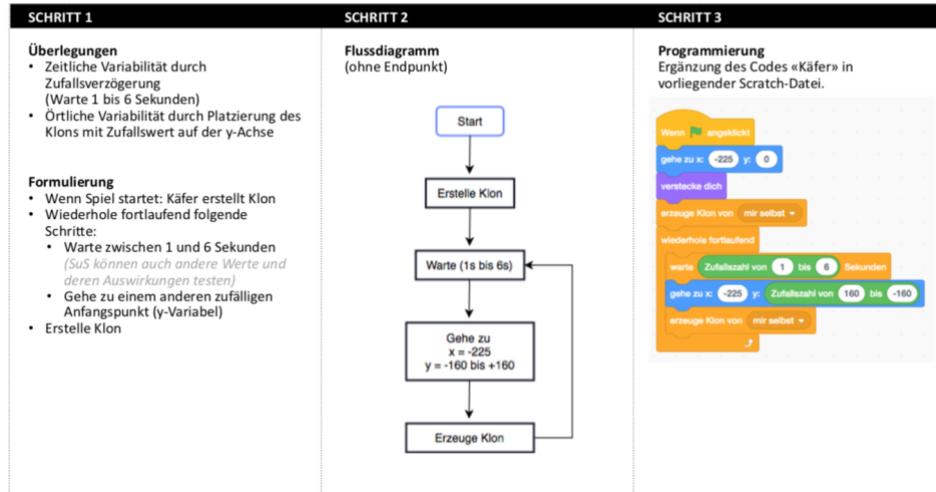


Abbildung 6 PH Luzern: Modul problembasiert Aufgaben – Ausschnitt Semesterarbeit Berger

3.2 Unterricht SEK I (7-9 Schuljahr)

Das TPA-Modell mit dem Schwerpunkt der problembasierten Aufgaben wird im Frühjahr 2019 an zwei Klassen des 9. Schuljahres an einem Gymnasium umgesetzt. Relevante Erkenntnisse können an der *GI-Fachtagung Informatik und Schule "Informatik für alle"* im Rahmen des Workshops präsentiert werden.

4 Fazit und Ausblick

Als erstes Fazit lässt sich auf Grund der Semesterarbeiten der Studierenden feststellen, dass die Fokussierung auf das Problemlösen in kleinen Teilschritten für den Lernprozess unterstützend war und die Programmierkonzepte so besser vermittelt werden konnten.

Bei der Weiterbildung von Lehrpersonen konnte mit dem TPA-Modell der Bereich Informatik des Dag-Stuhl-Dreiecks [7] klarer vermittelt und vor allem die Begriffe Algorithmen und Programmieren korrekt verortet werden. Weiter wird mit der Unterscheidung von *unplugged* und *plugged* eine Denkweise vermittelt, dass Informatikunterricht auch ohne Computer stattfinden kann.

Die praktische Umsetzung auf der Zielstufe wird zeigen, wie nachhaltig die Art und Weise des Programmierunterrichts gestaltet werden kann.

5 Literaturverzeichnis

- [1] K. Brennan und M. Resnick, „harvard.edu,“ 2012. [Online]. Available: <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>. [Zugriff am 26 01 2019].
- [2] BBC, „<https://www.bbc.com/bitesize/guides/zp92mp3/revision/1>,“ 2014. [Online]. Available: <https://www.bbc.com/bitesize/guides/zp92mp3/revision/1>. [Zugriff am 26 01 2019].
- [3] computationalthinkingcourse, „computationalthinkingcourse.withgoogle.com,“ google.com, 2015. [Online]. Available: <https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1>. [Zugriff am 26 01 2019].
- [4] J. Hromkovič, Sieben Wunder der Informatik, Wiesbaden: Vieweg+Teubner, 2009.
- [5] P. D. M. Rieger, P. Eisoldt, D. Schlichtenberger und T. Scheible, Applied Computer Systems, Hochschule Albstadt-Sigmaringen: Institut für wissenschaftliche Weiterbildung, 2017.
- [6] J. Berger, „Leistungsnachweis PH Luzern Modul pbA 3. Semester Medien und Informatik,“ PH Luzern, Luzern, 2018.
- [7] I. (GI), „[gi.de Website](https://gi.de/themen/beitrag/dagstuhl-erklaerung-bildung-in-der-digital-vernetzten-welt-1/),“ 2016. [Online]. Available: <https://gi.de/themen/beitrag/dagstuhl-erklaerung-bildung-in-der-digital-vernetzten-welt-1/>. [Zugriff am 02 02 2019].