EasyChair Preprint

№ 1052

TPA-Modell

Urs Meier

June 10, 2019

TPA-Modell

Vom Denken zum Handeln in der informatischen Bildung

Urs Leo Meier[1]

Abstract: Durch die Einführung des neuen Fachs Medien und Informatik im Rahmen des Lehrplans 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 weiterzuentwickeln, um ihre Schüler*innen im Lernprozess zu unterstützen. Insbesondere der Programmierunterricht benötigt viele Fertigkeiten auf verschiedenen Ebenen. Um die Abläufe eines Problemprozesses beim Programmieren für Lehrpersonen 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. Nach der Entwicklung des TPA-Modells wurde es in der Praxis eingesetzt, um problembasierte Aufgaben zu reduzieren und die grundlegenden Programmierkonzepte 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 und 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 anschliessend 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.

[1] Pädagogische Hochschule Luzern, Dozent Medien und Informatik, Sentimatt 1, 6003 Luzern, urs.meier@phlu.ch

2 TPA-Modell

2.1 Thinking

Das Denken steht im Modell an erster Stelle. Ein Problem aus der analogen Welt soll so modelliert werden, 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 vorrangig ein Problem analysiert und aufbereitet, damit das Problem von einem Computer verarbeitet werden kann.
Das vorliegende TPA-Modell stützt sich bildlich wie inhaltlich auf das Computational-Thinking-Modell aus England, das nach der Bildungsinitiative im Jahre 2014 von BBC [2] veröffentlicht wurde. Das Computational-Thinking-Modell der BBC ist heute weit verbreitet und wird unter anderem auch bei Google im Education-Programm [3] eingesetzt und basiert auf folgenden vier Begriffen.

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

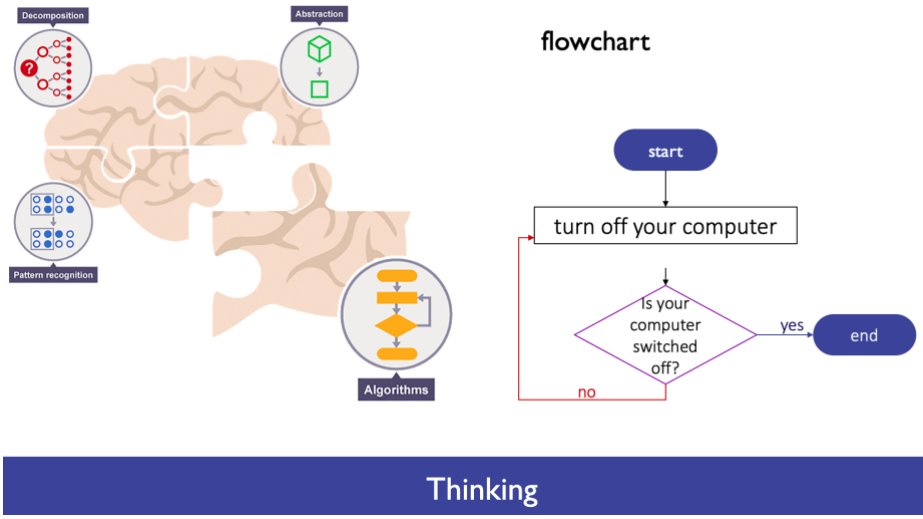

Abbildung 1 TPA-Modell: Thinking

Im ersten Teil Thinking wird ein Problem so analysiert und aufbereitet, damit als Ziel eine Abfolge von Anweisungen (Algorithmen) ausgearbeitet wird, die einem Computer übergeben werden können. Als symbolische Darstellung für den Algorithmus eignet sich ein Flussdiagramm oder Pseudocode.

2.2 Processing

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

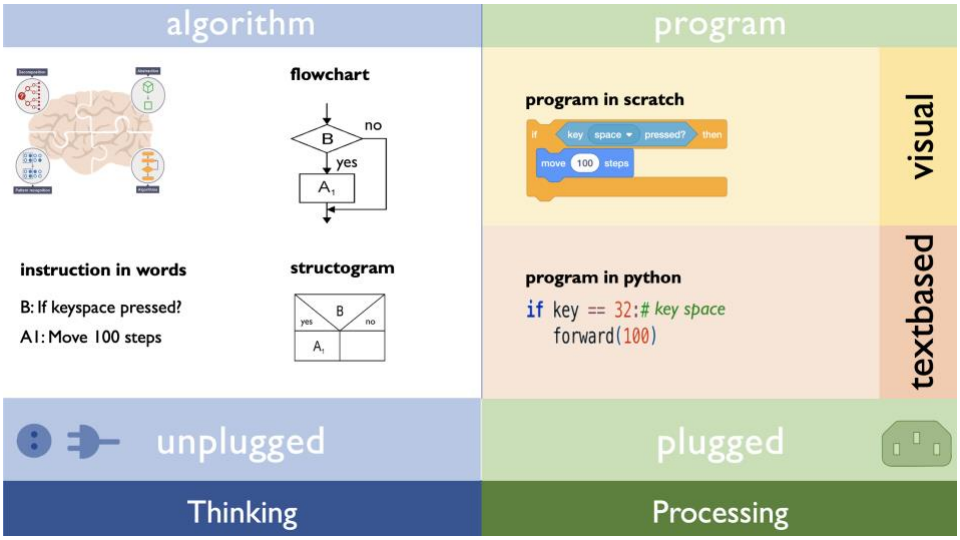

Abbildung 2 TPA-Modell – Processing

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.

Um die Arbeitsweise von Computern zu verdeutlichen, wird im TPA-Modell in Abbildung 3 eine zusätzliche kommunikative und technische Ebene eingeführt. Sie zeigt das Zusammenspiel von Software und Hardware. Die Anweisung der menschlichen Sprache führt über die Programmiersprache zu den Anweisungen in Maschinensprache, die das Ausführen eines Programmes mit der Recheneinheit ermöglicht. Auf die Unterscheidung von Interpreter und Compiler sowie JiT (Just in Time) -Compiler wird bewusst nicht eingegangen, sondern der Vorgang wird einfachheitshalber als Kompilieren dargestellt. Wichtig ist hier die Information, dass der Programmcode nochmals in einen Maschinencode übersetzt wird, der anschliessend von der Recheneinheit ausgeführt werden kann. Die Recheneinheit wird zusätzlich mit einem Transitor als kleinste Schalteinheit abgebildet.

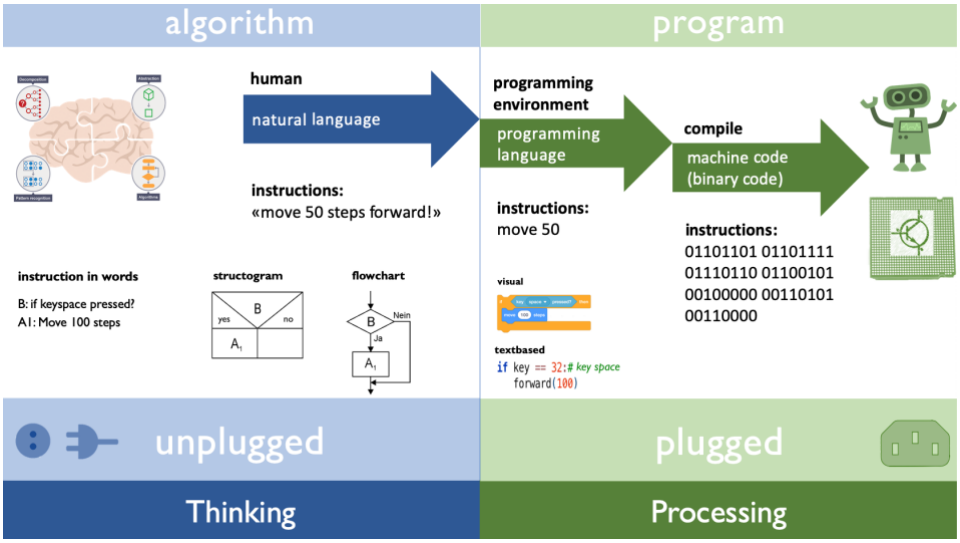

Abbildung 3 TPA-Modell - Processing and Compiling

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, deren Verarbeitung in einem Prozessor sowie die Ausgabe auf einem Bildschirm bzw. einem Aktor.

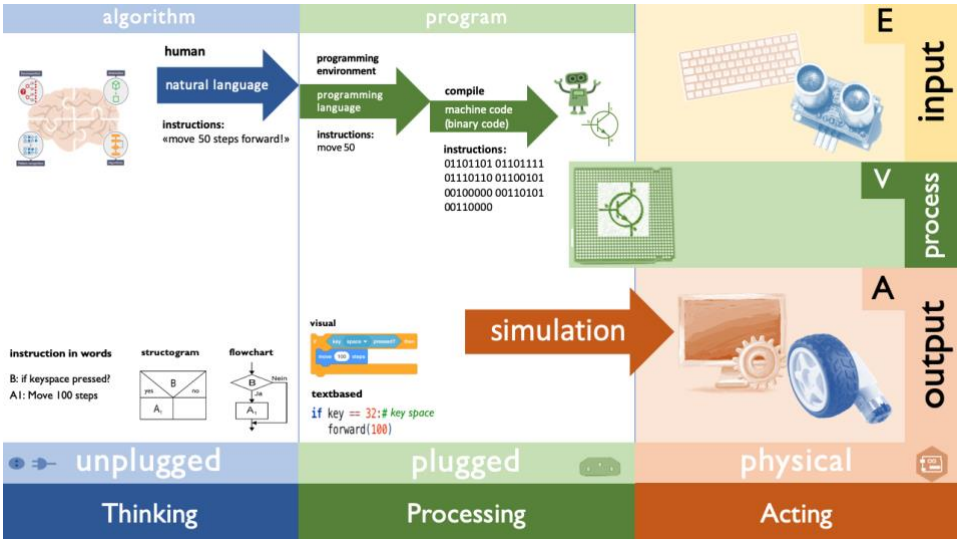

Abbildung 4 TPA-Modell – Acting

Das TPA-Modell zeigt im Überblick auf, wie vom Denkprozess bis hin zur Anzeige auf dem Bildschirm viele kleine Schritte nötig sind. Daraus resultierend kann aufgezeigt werden, wie und wo Fehler bei der Entwicklung von Software entstehen können. Besonders wertvoll ist die Sichtweise für Lehrpersonen, die zukünftig Algorithmen und Programmieren unterrichten. Der Überblick zeigt auf, wo bei Projekten von Schüler*innen nach möglichen Fehlern gesucht werden kann. Fehlt beispielsweise eine physikalische Verbindung beim Input, habe ich mich vertippt (plus/minus) im Programmcode oder ist meine Überlegung im Flussdiagramm falsch?

Für die Fachdidaktik bietet das TPA-Modell die Möglichkeit, einzelne Aspekte aufzunehmen und auf diese im Unterricht einzugehen.

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. Es wird bewusst bei kleinen und einfachen Problemen angewendet. Die grundlegenden Programmierkonzepte können dadurch vermittelt werden.

Im konkreten Einsatz bei problembasierten Aufgaben werden folgende Ziele verfolgt:

- 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 den Lehrpersonen einerseits 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 Begriffe erklärt und richtig verortet werden können. Es wird unter anderem dargelegt, wie man Programmieren in Schritten vermitteln kann. Ein Schwerpunkt ist die Förderung des algorithmischen Denkens und das Lösen von Problemen.

Konkret wird das TPA-Modell in Aus- und Weiterbildungen herangezogen, um aus einer Problemstellung kleine Teilprobleme herauszuschälen und zu vermitteln.

Sollen Schülerinnen und Schüler beispielsweise ein Rennspiel programmieren, könnten sie folgende Teilproblemstellungen ermitteln:

- Steuerung eines Autos mit Tasten
- Das Auto so steuern, damit es immer auf der Bahn bleibt
- Runden zählen
- Tempo messen
- Akkustand bei Elektro-Formel 1 anzeigen
- Hindernisse einbauen
- …

Für eine konkrete 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.

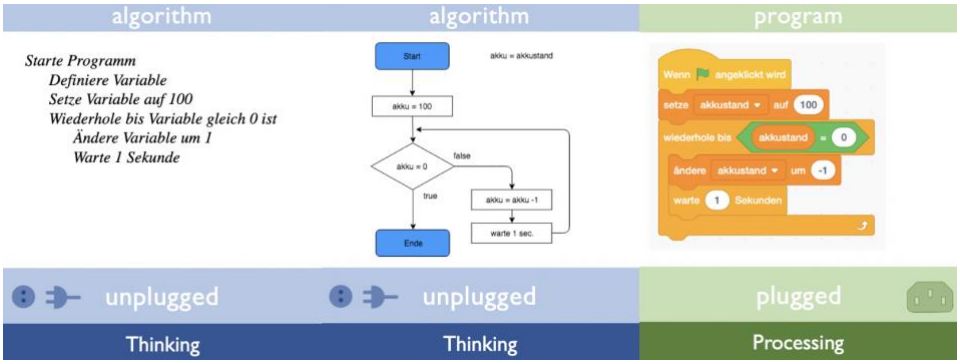

Abbildung 5 Problembasierte Aufgaben - Teilschritte

Das so analysierte Teilproblem zeigt die verschiedenen Darstellungsformen mit entsprechender Abstraktionsebene gut auf und hilft, die erwähnten Ziele zu verfolgen.

3.2 Beispiel-Umsetzung

Das vorgestellte TPA-Modell wurde 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 bei einem Spiel das Teil-Problem «zufälliges Auftauchen eines Käfers» analysiert und schrittweise in einem Programm umgesetzt wird.

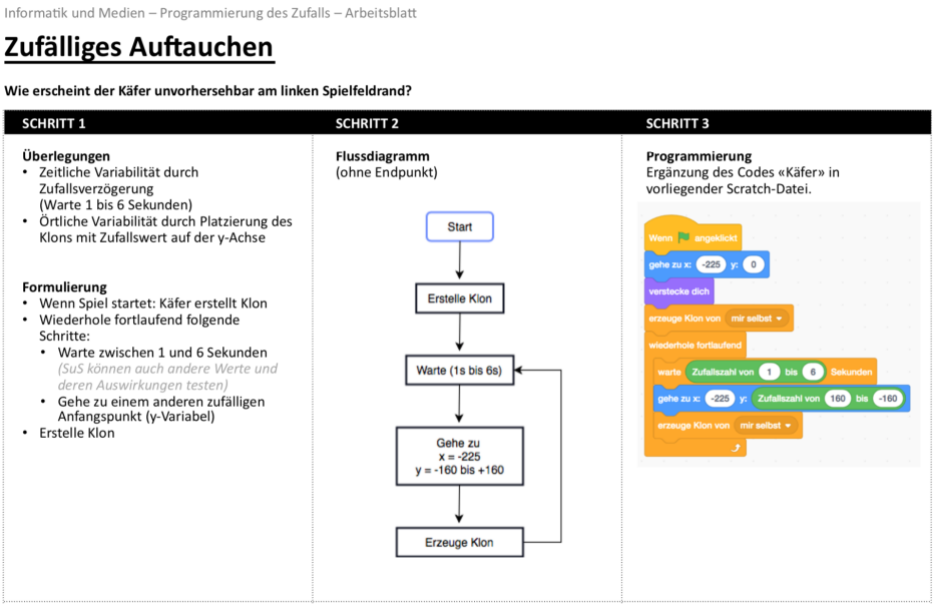

Abbildung 6: Modul problembasierte Aufgaben – Ausschnitt Semesterarbeit Berger

3.3 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 deutlicher verortet werden. Weiter wird mit der Unterscheidung von *unplugged* und *plugged* die 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 Modul pbA 3. Semester Medien und Informatik," PHLU, Luzern, 2018.

[7] I. (GI), „gi.de Website," 2016. [Online]. Available: https://gi.de/themen/beitrag/dagstuhl-erklaerung-bildung-in-der-digital-vernetzten-welt-1/. [Zugriff am 02 02 2019].