

**WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER**

Vergleich bildlich-grafischer Repräsentationen der Turing Maschine

Bachelorarbeit

von
Torben Schultz
Mat.-Nr. 384453

vorgelegt bei
Prof. Dr. Marco Thomas
und
Prof. Dr. Jan Vahrenhold

22.06.2015

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Tabellenverzeichnis	II
1. Motivation	1
2. Die Turing Maschine als thematischer Schwerpunkt	2
2.1 Theoretische Grundlagen der Turing Maschine:	2
2.2 Die TM in der Schule/ als Unterrichtseinheit	4
2.3 Anforderungsdefinition an Visualisierung von Algorithmen/Konzepten der Informatik.....	6
3. Aufstellung und Anwendung eines Systems zur Bewertung von Turing Maschinen Simulationen.....	11
3.1 Analysen existierender Visualisierungen.....	11
3.2 Betrachtung existierender Visualisierungen der Turing Maschine	17
4. Der Entwicklung einer schulisch wertvollen Turing Maschine	28
5. Abschließende Diskussion	31
Literaturverzeichnis	a
Glossar	c
Abkürzungsverzeichnis	c
Anhang.....	c
Anhang 1: Begründung der Kriterien-Bewertung	d

Abbildungsverzeichnis:

<i>Abbildung 1: Aufbau einer Turing Maschine.</i>	<i>3</i>
<i>Abbildung 2: Beispielhaftes Programm einer Turing Maschine über dem Alphabet $=\{0,1\}$.....</i>	<i>4</i>
<i>Abbildung 3: Verteilung existierender Visualisieren nach Shaffer et al. 2007.....</i>	<i>9</i>
<i>Abbildung 4: Turing-Simulator nach Start des Programms.....</i>	<i>19</i>
<i>Abbildung 5: TuringKara nach Start des Programms.</i>	<i>21</i>
<i>Abbildung 6: JFlap nach Start des Programms</i>	<i>24</i>
<i>Abbildung 7: Ausschnitt von Morphetts Simulator nach Öffnen der Website.</i>	<i>26</i>
<i>Abbildung 8: Gliederung der Teilgebiete</i>	<i>28</i>

Tabellenverzeichnis:

<i>Tabelle 1: Kriterien für die Wahl von Unterrichtsinhalten inklusive Wertung.....</i>	<i>5</i>
<i>Tabelle 2: Auflistung der als relevant erkannten Kriterien zuzüglich möglicher Einzelbetrachtungen.....</i>	<i>17</i>
<i>Tabelle 3: Liste verfügbarer Simulatoren.....</i>	<i>18</i>
<i>Tabelle 4: Betrachtung des Turing-Simulators.....</i>	<i>20</i>
<i>Tabelle 5: Betrachtung von TuringKara.....</i>	<i>23</i>
<i>Tabelle 6: Betrachtung von JFlap</i>	<i>25</i>
<i>Tabelle 7: Betrachtung des webbasierten Turing Maschinen Simulators</i>	<i>27</i>
<i>Tabelle 8: Sortierte Kriterien-Liste</i>	<i>28</i>

1. Motivation

Die Geschichte von Alan Turing und seinem Konzept der Turing Maschine beginnt bei David Hilbert, welcher 1928 das „Entscheidungsproblem“ publizierte (Hilbert und Ackermann 2013, S. 119). Dieses fragt nach einem Verfahren, welches für ausreichend formalisierte Aussagen entscheidet, ob diese wahr oder falsch sind.

Schon kurz darauf erschien das Werk von Alan Turing mit dem Titel „On computable numbers, with an application to the Entscheidungsproblem“ (Turing 1936).

In diesem nimmt er Bezug auf das von Hilbert geäußerte Problem und erklärt die Idee und Grundlagen seiner Turing Maschine, die ein einfaches, aber gleichzeitig sehr mächtiges Werkzeug der theoretischen Informatik darstellt. Eingesetzt im Unterricht erlaubt sie die Behandlung einer Vielzahl relevanter Thematiken, beispielsweise Grammatiken, Zustandsdiagramme oder Aussagen über die Grenzen der Berechenbarkeit.

Der Arbeitsprozess der Turing Maschine verbindet (in der Theorie) mechanische Arbeiten mit vorgegebenen Formalismen. Dies ist auch ein Grund, wieso die Arbeit mit Turing Maschinen in der Schule oft mithilfe von Simulationen betrachtet wird, welche das konkrete Arbeiten an einem theoretischen Konzept realisieren. Hier liegt eine große Chance für den Informatikunterricht: Das selbstständige Arbeiten mit einem theoretischen Konzept fördert unter anderem das Verständnis für die behandelten Inhalte und das eigenständige Handeln.

Diese Bachelorarbeit soll einen Beitrag zur qualifizierten Auswahl geeigneter Simulationen leisten. Zu diesem Zweck wird zunächst die Turing Maschine als Unterrichtsmedium betrachtet. Anschließend erfolgt eine Aufstellung von Kriterien, welche eine Grundlage der Bewertung für Simulationen bilden soll. Um dies zu verdeutlichen werden im Folgenden die Kriterien angewandt, um ausgewählte Visualisierungen zu betrachten. Die positiven oder negativen Umsetzungen der Kriterien werden im letzten Schritt thematisiert.

2. Die Turing Maschine als thematischer Schwerpunkt

2.1 Theoretische Grundlagen der Turing Maschine¹:

Ein unendliches Band, eine Steuereinheit und ein Lese/Schreibkopf bilden aus schematischer Sicht ein Werkzeug ab, welches die Grenzen der Berechenbarkeit definiert.

„The Church-Turing thesis [...] says that any real-world computation can be translated into an equivalent computation involving a Turing machine. In Church's original formulation (Church 1935, 1936), the thesis says that real-world calculation can be done using the lambda calculus, which is equivalent to using general recursive functions.“ (Rowland)

Mit anderen Worten: Jede intuitiv berechenbare Funktion kann mithilfe einer Turing Maschine abgebildet werden. Somit gilt: Ist eine Turing Maschine nicht in der Lage, in endlicher Zeit eine Berechnung durchzuführen, ist diese nicht berechenbar.

Unter Betrachtung des oben aufgeführten Aufbaus der Turing Maschine fällt auf, dass der versuchte Nachbau einer Turing Maschine an der Bereitstellung eines unendlichen Bands scheitert. So existiert für eine reale Umsetzung eine obere Grenze in Form der verfügbaren Bandlänge. Folglich lassen sich Berechnungen auf einem hohem Komplexitätsniveau nur bedingt auf einer realen Turing Maschine (TM) durchführen.

Die TM hat in der Gegenwart vielmehr eine Bedeutung als mathematisches und theoretisches Modell.

Dieses Modell kann durch ein sieben Tupel-System wiedergegeben werden:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_A, q_V)$$

- mit:
- Q als endliche Menge aus Zustände
 - Σ als endliches Alphabet
 - Γ als endliches Eingabealphabet
 - δ als Übergangsfunktion zwischen Zuständen
 - $(q_0, q_A, q_V) \in Q$ als Start-, akzeptierender und verwerfender Zustand

Auf Basis des Eingabealphabets, welches eine endliche Menge Zeichen enthält, kann ein

¹ Die folgende Abhandlung basiert auf dem Werk Turgings: „On computable numbers, with an application to the Entscheidungsproblem“ Turing 1936.

beliebiges Wort dieser Sprache ausgewählt werden. Dieses wird mithilfe des Schreib-/ Lesekopfs beginnend an einer Stelle x Zeichen für Zeichen rechtsgerichtet auf das Band geschrieben.

Ist dieser Prozess abgeschlossen, bewegt sich der Kopf wieder zurück auf die Ausgangsposition. Die Turing Maschine kann mit ihrer Arbeit beginnen.

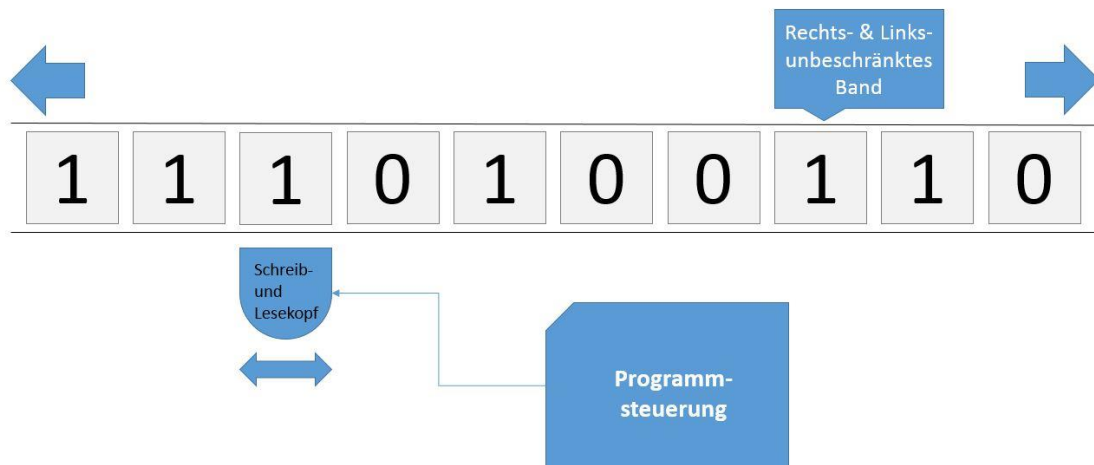


Abbildung 1: Aufbau einer Turing Maschine, bestehend aus einem unendlichen Band, Schreib- und Lesekopf sowie einer Programmsteuerung.

Der Kopf liest nun das erste Zeichen der Eingabe ein. An dieser Stelle interessiert ein weiteres Konzept der TM: Die TM fungiert als Zustandsautomat. Auf der Grundlage des eingelesenen Zeichens kann nun ausgehend von Anfangszustand q_0 anhand der Übergangsfunktion erkannt werden, welcher Folgezustand gewählt werden kann. Die Übergangsfunktion enthält des Weiteren Befehle an die Turing Maschine. So wird im nächsten Schritt aufgezeigt, ob und was der Kopf auf dem Feld schreiben soll. Am Ende des Übergangs wird festgehalten, inwiefern sich der Kopf nach links oder rechts auf dem Band bewegen soll.

Der skizzierte Prozess wird weitergeführt und die TM arbeitet die Befehle des Zustandsautomaten ab. Der Prozess endet durch Erreichen eines der zwei Sonderzustände: q_A und q_V , der akzeptierende Zustand und der verwerfende Zustand.

Das Erreichen des Zustands q_A bedeutet, dass die TM das eingebende Wort w akzeptiert.

Daneben existieren zwei weitere Fälle:

1. Die TM erreicht den verwerfenden Zustand q_V . Die TM akzeptiert das Wort nicht.
2. Die TM erreicht nie einen Zustand q_A / q_V . Die TM stoppt nicht.

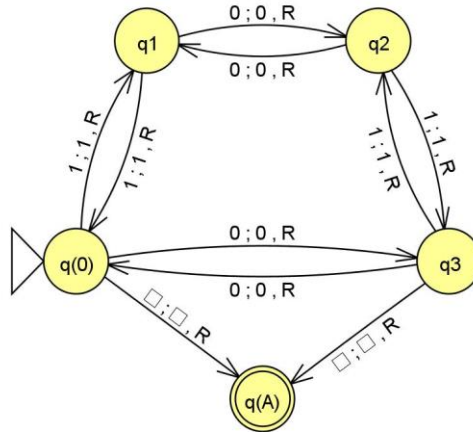


Abbildung 2: Beispielhaftes Programm einer Turing Maschine über dem Alphabet $=\{0,1\}$, welches eine Eingabe mit einer geraden Anzahl des Zeichens „1“ akzeptiert, erstellt mithilfe von JFlap.

Neben der genuine 1-Band Turing Maschine liegen verschiedene Modifikationen vor: Mehrband-TM, Halbband-TM, Nichtdeterministische TM, Universelle TM und viele weitere. Ein wesentlicher Grund der Berühmtheit der TM ist die Universelle TM, deren Idee ist, einer TM U den codierten Programmcode einer TM M zu übergeben. So simuliert die TM U die Maschine M.

Mithilfe des Konzepts der Turing Maschine können spezifische Probleme konstruiert werden, die aus theoretischer Sicht von Bedeutung sind oder alltäglich mögliche Situationen beschreiben: Anhand des Halteproblems kann gezeigt werden, wieso gelegentlich Programme abstürzen. Weiterhin kann durch geschickte Reduktionen die Grenzen der Berechenbarkeit und Entscheidbarkeit aufgezeigt werden.

2.2 Die TM in der Schule/ als Unterrichtseinheit

Ziel dieser Bachelorarbeit soll sein, ein geeignetes Werkzeug zur Behandlung der Thematik TM im Schulunterricht zu finden. Dieser Schritt kann jedoch nicht ohne kritisches Hinterfragen der Thematik in Bezug auf seine Relevanz im Unterrichtsgeschehen erfolgen. Hierzu erfolgt ein Rückgriff auf die Kriterienwahl von Modrow für die Auswahl von Unterrichtseinheiten

(Modrow 1991, S. 71–72), da diese Arbeit nicht auf die Erforschung der Relevanz von informatischen Themen abzielt.

Kriterium	P/N
Die Schüler können selbstständig in diesem Gebiet arbeiten.	P
Das Thema eignet sich für arbeitsteiligen Unterricht und/oder Projektarbeit.	P
Das Thema ermöglicht längere selbstbestimmte Arbeitsphasen der Schüler/innen. Der Lehrer muss nicht kleinschrittig den Unterricht führen.	P
Es gibt nicht nur eine Standardlösung für das Problem. Die Schülerinnen und Schüler (SuS) können das Problem auf unterschiedlichen Wegen lösen.	P
Das Thema gestattet Spezialisierungen und Erweiterungen, so dass auf unterschiedlichem Niveau gearbeitet werden kann. Es ermöglicht durch die unterschiedlich komplexen Lösungen Binnendifferenzierung.	P
Die Relevanz des Themas ergibt sich nicht nur aus seiner Bedeutung für die derzeitige Technologie.	P
Die Relevanz des Themas ergibt sich nicht nur aus Prognosen über seine zukünftige Bedeutung.	P
Aussagen über gesellschaftliche Folgen der Datenverarbeitung sind möglich, sie können sachlich fundiert behandelt werden.	N
Das Thema spricht prinzipielle Fragen an. Es beschränkt sich nicht nur auf eine aktuelle Mode.	P
Der Zeitbedarf zum Erlernen von Benutzerwissen steht in einem angemessenen Verhältnis zum angestrebten Lernerfolg.	N
Die Ausgangsproblemstellung kann so einfach gehalten werden, dass die ohne großes Spezialwissen verstanden und bearbeitet werden kann. Der Zugang zum Thema ist also „naiv“ möglich.	N
Fachliche Inhalte können elementar behandelt werden. Spezialkenntnisse und -methoden sind aus dem Problem ableitbar, ihre Kenntnis nicht Voraussetzung zu seiner Bearbeitung.	P
Die entwickelte Fachsprache wird entwickelt und benutzt. Formale Begriffe werden nicht nur eingeführt, sondern auch angewandt. Die Schüler können die benutzten Begriffe erklären.	P
Die benutzten Modelle sind als solche erkennbar. Alternative Modellvorstellungen sind möglich. Der Modellbildungsprozess wird deutlich.	P
Die bei der Bearbeitung des Themas gewonnen Erfahrungen können verallgemeinert werden. Einblicke in informatikspezifische Themen sind möglich.	P
Das Thema deckt mehrere Ziele des Informatikunterrichts ab, ist multifunktional einzusetzen.	P
Das Thema steht nicht isoliert, sondern bereitet auch spätere Fragestellungen vor.	P

Tabelle 1: Kriterien für die Wahl von Unterrichtsinhalten inklusive Wertung herausgegeben von Modrow(1992); P drückt eine Zustimmung, N eine Verneinung bzw. Einschränkung aus. Eine Begründung der Bewertung liegt im Anhang vor.

Grundlage der Bewertung ist die subjektive Meinung des Autors. Das Verhältnis von 14:3 P:N ist ausreichend, um die Thematik im Unterricht als relevant einzustufen.

Dieser Einschätzung ist auch Modrow gefolgt. Bezugnehmend auf die Bearbeitung des Themas hält er fest: Das „Finden eines Weges zu Fragen der Berechenbarkeit und Entscheidbarkeit führt über die vertrauten Begrifflichkeiten der Automatentheorie“ (Modrow 1992, S. 361). Zu diesem Zweck erfolgt die Einführung der Turing Maschine in den Unterricht.

Modrow führt auf, dass die Behandlung der Turing Maschine strikt mit der tatsächlichen Verwendung dieser verbunden sei. Weiterhin weist er auf die Notwendigkeit eines Simulationsprogramms hin, um die Arbeitsweise der TM zu erproben. Das Aufzeigen der Mächtigkeit erfolgt durch Hintereinanderschalten mehrerer TM (Verkettung) mit dem Ziel komplex strukturierter Programme.

Darüber hinaus ist die Thematik der TM in mehreren Bundesländern in den Rahmenlehrplänen zu finden: In Niedersachsen wird die Turing Maschine in Leistungskursen thematisiert. Diese Ergänzung ist dem Schwerpunkt der theoretischen und technischen Modellvorstellungen untergeordnet. Inhaltlich werden drei Kompetenzen entwickelt: Problemlösung, Analyse und Erweiterung einer TM (Niedersächsisches Kultusministerium 2012). Ähnlich wird die Turing Maschine in den Lehrplänen für die gymnasiale Oberstufe in Berlin vorgesehen. In Leistungskursen soll die „Arbeitsweise eines allgemeinen Maschinenmodells“ analysiert werden (Berliner Landesinstitut 2006). Die Lehrpläne anderer Bundesländer beschreiben das Thema: „Grenzen der Berechenbarkeit“, ohne die Turing Maschine namentlich zu erwähnen (Ministerium für Bildung, Familie, Frauen und Kultur Saarland 2008).

2.3 Anforderungsdefinition an Visualisierung von Algorithmen/Konzepten der Informatik

Die Arbeit mit der TM an Schulen und Universitäten wird häufig durch Modelle und Simulationen gestützt. Die visuelle Betrachtung eines komplexen Sachverhalts erleichtert oft das Verständnis hierfür. Als Ausgangspunkt einer Visualisierung müssen zunächst einige Grundbegriffe eingeführt werden. So existiert in der Literatur kein einheitlicher Konsens über Begrifflichkeiten der Modelltheorie. Dies beginnt bereits bei dem Begriff Modell. Das Modell als Abbildung der Realität wäre ein kurzer Beschreibungsversuch.

Ein etwas tiefergehender Ansatz: In der Abhandlung zum Thema „Bemerkungen zur Modellbildung und zum Modellbegriff“ von Ernst Kirchner führt dieser den Erkenntnisprozess nach Stachowiak² an. Nach diesem beinhaltet der Modellbegriff die „elementarste Wahrnehmungsgegebenheit, sowie die komplizierteste umfassende Theorie“ (Kirchner 1983, S. 12). Der oben erwähnte Erkenntnisprozess hat seinen Ursprung in einem genuinen Gegenstand oder einer Theorie. Durch den Prozess der Abbildung entstehe laut Kirchner ein Modell, welches einen klaren Anspruch an die Nutzer, den Zweck und den Nutzungszeitpunkt stelle. Als Resultat dieses Prozesses stehe die Erkenntnis, welche „konstruktiv, subjektabhängig und subjektbezogen“ sei (vgl. Ebd.).

Mit Rückgriff auf die Modelltheorie Stachowiaks führt Kirchner (Kirchner 1983, S. 14) drei allgemeine Merkmale einer Visualisierung auf:

1. *Abbildungsmerkmal*
2. *Verkürzungsmerkmal*
3. *Pragmatisches Merkmal*

Kirchner stellt diese anhand von M-O, M-S Relationen vor (Kirchner 1983, S. 13–15).

Hierbei erläutert er zunächst das *Abbildungsmerkmal*: Das Subjekt möchte ein Objekt erforschen. Um ein besseres Verständnis zu gewinnen entwerfe es ein Modell, welches sich durch eine möglichst hohe Anzahl Analogien auszeichnet, um die Eigenschaften des Objekts abzubilden. Überprüft würden diese Analogien möglichst über geeignete Experimente.

Der abgebildete Prozess diene auch zur Beschreibung von Lern/Lehr – Szenarien.

So lassen sich die Annahmen treffen, dass das Subjekt (hier: der Schüler) bereits Vorstellungen, Erfahrungen und Vorkenntnisse bezüglich des Objektes besitze. Durch den Wissensgewinn anhand des Modells könne der Schüler das Modell weiterentwickeln. So entstehe ein kontinuierlicher Lernprozess.

Das *Verkürzungsmerkmal* erscheint selbsterklärend: Der Versuch, ein Objekt zu modellieren, erfolge über die Reduktion des Objektes auf grundlegende, vereinfachte Prinzipien. Dies sei insbesondere im Schulunterricht essentiell. Hier treten Parallelen zu Klafkis Sachanalyse oder

² Kirchner zitiert nach Stachowiak 1973.

der Elementarisierung auf (Klafki 1964). In beiden Fällen geht es um die Aufbereitung des zu Lehrenden und das Finden des Grundlegenden oder Einfachen als Teilgebiete der didaktischen Reduktion. Dieses Merkmal erlangt hohe Wichtigkeit durch seine Funktion als Selektor zwischen wichtigen Merkmalen, Schmuckwerk und Störfaktoren.

Das *pragmatische Merkmal* finde seine Bedeutung in der Zweckgebundenheit und Subjektorientierung. Der Pragmatismus der Betrachtung spiegele sich in der Beantwortung von Fragen bezüglich des Adressaten, des Grundes und des Zeitpunktes wider.

Da an Schulen eine andere Zielsetzung als an Universitäten verfolgen, können Modelle ideal für den Schulunterricht und unzureichend oder sogar nachteilig für den Einsatz an der Universität sein. Ein gutes Beispiel hierfür ist die Hinführung zum Atommodell im Chemieunterricht mithilfe des Bohr'schen Modells. Dieser Effekt kann in beide Richtungen auftreten.

Der Oberbegriff „Modell“ ist für die weitere Betrachtung zu allgemein. Zielführend erscheint, eine exakte Spezifizierung dieses bisher als Lemma fungierenden Begriffs vorzunehmen.

Musumbu (Musumbu 2014, S. 1) benennt drei Kategorien:

1. *Formale Modelle*
2. *Deskriptive Modelle*
3. *Graphisch-animierte Modelle*

Für exakte Definitionen der einzelnen Kategorien wird auf die angegebene und verwandte Literatur verwiesen. Das Hauptziel dieser Arbeit ist, wie bereits erläutert, die Analyse existierender graphischer Umsetzungen. Diesem Ziel folgend, sind die unter Punkt [3] erwähnten Modelle interessant. Der Einsatz dieser Umsetzung (im Folgenden: *Visualisierung* oder *Simulation*) kann unterschiedliche Ziele verfolgen:

Auf der einen Seite zeigen Simulationen den Ablauf eines Prozesses in Form der Quintessenz des Originals. Auf der anderen Seite ist die Aufgabe der Visualisierung die Unterstützung eines Nutzers bei der Bearbeitung einer Aufgabe (Hundhausen et al. 2002, S. 1).

Weiterführend ist es aber möglich, die Art dieser Hilfe zu klassifizieren(vgl. Ebd., S.1)³:

1. Hilfsprogramm: Finden von Bugs, Performance-Vergleiche und Fehlern
2. Wiedergabe: Verarbeitung und Wiedergabe größerer Programme als Überblick
3. Lernmodell: Lernen von Algorithmen u. Methoden, Einblicke in informatische Themen

Für diese Arbeit relevant sind ausschließlich Simulationen, die in der schulischen Lehre (Sek I+II) eingesetzt werden können. Die existierenden Visualisierungen im Bereich der Informatik verteilen sich ungleichmäßig über das Spektrum der Thematiken (Shaffer et al. 2007, S. 151–152). 38 % der erfassten Visualisierungen befassen sich mit Sortieralgorithmen, 17 % mit Such-Algorithmen. Eine zulässige Deduktion: Viele Themen sind kaum oder gar nicht mithilfe von Simulationen erfasst.

VERTEILUNG EXISTIERENDER VISUALISIERUNGEN

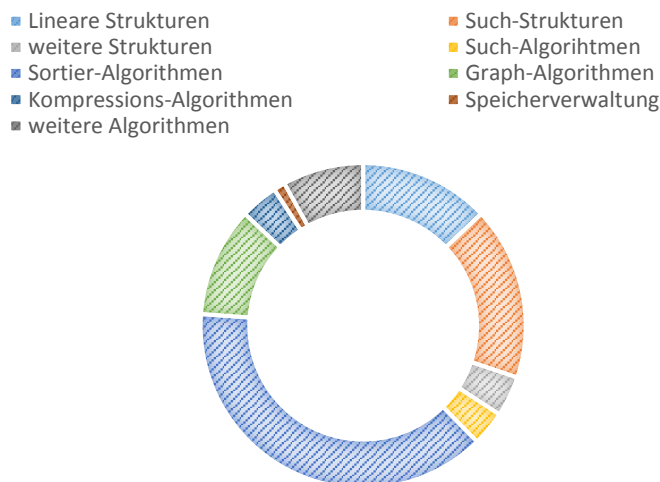


Abbildung 3: Verteilung existierender Visualisierungen nach Shaffer (Shaffer et al. 2007, S. 152). Betrachtet wurden 352 Visualisierungen kategorisiert.

In vielen Fächern gingen Visualisierungen als Nachfolger von Analogmodellen, also Ersetzungen der Realität durch ein anschauliches (reales) Beispiel hervor (Naumer und Martin 1983, S. 28). Die Verwendung von Simulationen im Unterricht eröffne neue Möglichkeiten. So können Theorie und Realität eng miteinander verflochten werden (Treitz 1983, S. 38). Dies ermögliche den SuS einen verständnisvolleren Zugang zu einem Thema. Möglich seien auch

³ Hundhausen zitiert nach Price, 1990; Naps, 1990; Heath&Etheridge, 1991; Gu et al.1994; Brown 1988a.

Rückschlüsse aus der Simulation auf eine Theorie. Insbesondere bieten Simulationen ein großes Potential für MINT-Fächer. Jedoch sind Simulationen nicht als die geeignetste Methode des Lernens in der Schule zu verstehen. Studien sind geteilter Meinung über den Einfluss von Visualisierungen im Unterricht. So sehen einige Studien keinen signifikanten Mehrwert für den Einsatz im Gegensatz zu herkömmlichen Medien. Andere Studien hingegen gelangten zu deutlich positiveren Resultaten (Shaffer et al. 2010, S. 6).

Dieser Überlegung nachgehend kann vermutet werden, dass Simulationen nicht für alle Fächer und Situationen produktiv und effektiv sind. Der Einsatz von Visualisierungen muss zielorientiert und überlegt erfolgen. So könne der Ersatz eines durchführbaren Experiments durch eine Simulation als Missbrauch gewertet werden (Treitz 1983, S. 38). Kritik müsse auch aus kognitiver und affektiver Sicht hervorgebracht werden: Vielen Schülern in der Sek I fehle das nötige Abstraktionsdenken, um die Analogie nachzuvollziehen oder diese zu akzeptieren (Naurer und Martin 1983, S. 30).

Der richtige Einsatz biete jedoch aussichtsreiche Möglichkeiten, denn Simulationen ermöglichen einen offenen Unterricht. Die SuS können in der für sie notwendigen Zeit lernen, die Bearbeitung kann an ihren individuellen Lerntyp und ihre Vorlieben angepasst werden. So lösen leistungsstarke Schülerinnen und Schüler (SuS) schwerere Aufgaben mithilfe einer Simulation, während leistungsschwache SuS kleinere Probleme lösen (Hundhausen et al. 2002, S. 39)⁴.

Als Weisungen an die Simulation im Unterricht führt Staudachers (Staudacher 1985, S. 10) die folgenden Punkte an:

Zum einen solle die Visualisierung sicherstellen, dass ausreichend kognitive Grundlagen vorhanden seien, damit die SuS diese in der Praxis anwenden können. Bestenfalls ist eine Entlastung des Lehrerkörpers möglich, welcher dann SuS besser fördern und fordern könne. Zudem sei die Simulation (insbesondere bei der steigenden Anzahl an Sicherheitsbedenken in den MINT-Fächern wie Chemie, Anmerkung des Autors) als Versuchslabor zu sehen und ermögliche Darstellungen, die sonst nur theoretisch abgehandelt werden konnten. Zum anderen könne die Glaubwürdigkeit von primitivisierten Experimenten erhöht werden. Viele in der Schule notwendigen Vereinfachungen und Annahmen könnten (in Grenzen) bestätigt oder ersetzt werden. So existiere in der Schule eine Vielzahl physischer Hilfsmittel, um Sachverhalte zu erläutern, ohne diesen voll zu erfassen. Die Variabilität von Programmen fungiere hier als Pluspunkt für Simulationen. Leider existiert ein weiteres schwerwiegendes Problem: Von den

⁴ Hundhausen zitiert nach Spradley & McCurdy 1972.

existierenden Bewertungen sind viele von minderwertiger Qualität und von keinem bzw. geringem pädagogischen Nutzen (Shaffer et al. 2010, S. 1). So fällt den Lehrern die schwierige Aufgabe der Selektion zwischen geeigneten und ungeeigneten Programmen zu. Eine reflektierte Auswahl ist hier von größter Wichtigkeit. Diese kann mithilfe einer Analyse vorgenommen werden. Eine exemplarische Umsetzung lässt sich im nachfolgenden Teil der Arbeit finden

3. Aufstellung und Anwendung eines Systems zur Bewertung von Turing Maschinen Simulationen

3.1 Analysen existierender Visualisierungen

Die Bewertung existierender Simulationen bezüglich ihrer Tauglichkeit zum Einsatz im Unterricht erfordert die Aggregation wichtiger Projektziele. Diese enthält bestmöglich alle relevanten Bereiche in Technik, Design und Pädagogik.

So muss als erster Schritt die Wahl und Aufstellung wichtiger Attribute erfolgen. Anschließend findet eine Wertung dieser Einzelinhalte statt, um ein allgemein anwendbares System zu entwerfen.

Identifikation und Benennung aller relevanten Elemente und Ereignisse:

Als Grundlage eines jeden Systems steht zunächst die Frage nach der Vollständigkeit. Das Merkmal der Vollständigkeit ist erfüllt, wenn alle wichtigen Bestandteile und Ereignisse identifiziert und im Endprodukt korrekt umgesetzt wurden (Hundhausen et al. 2002, S. 5). Dies ist jedoch kein rein quantitatives Merkmal. Je nach Umsetzung können Elemente geschickt kombiniert oder vereinfacht werden. Immerhin repräsentiert eine Visualisierung die Art und Weise wie der Herausgeber das System sieht.

Die Benennung der einzelnen Bestandteile kann auf unterschiedliche Art erfolgen. Da das Interesse vordergründig auf dem Einsatz in der Schule liegt, müssen zwangsweise alle Elemente des Programms für die Schüler identifizierbar sein. Verwendbar sind hierfür beispielsweise Pop-up Fenster, welche Erklärungen als Einblendungen beinhalten (Rößling und Naps 2002, S. 3). Denkbar schlecht wäre ein System, welches kontinuierlich Erklärungsbedarf seitens der Schüler induziert. Dies ist nicht nur kontra-produktiv für die Selbstständigkeit dieser, sondern birgt auch ein gewisses Frustpotential, da die SuS das System nicht erfassen können.

Das Kriterium gilt als vollständig erfüllt, wenn das System als vollständige Abstraktion seitens eines Experten angesehen werden kann.

*Klarheit und Angemessenheit (Hundhausen, 1996, S. 3)*⁵

Als Mittel zum Einsatz im Bildungsbereich sind die Übersichtlichkeit und Angemessenheit probate Indikatoren für die richtige Verwendung. Das Anforderungsniveau der Schule unterscheidet sich signifikant von dem Bildungsstandard der Universität. Auch muss zwischen Sekundarstufe I und II unterschieden werden. Die aufgezählten Punkte können jedoch von Subjekt zu Subjekt divergieren. Aus diesem Grund finden sich hier die betrachteten Indikatoren: *Klarheit*: Strukturierter Aufbau, Erklärungen und Hinweise, Verdeutlichung der Möglichkeiten und Grenzen, Einheitlichkeit und Konsistenz (Latzina und Wedekind 1986, S. 38).

Angemessenheit: Anforderungsniveau, thematische Durchdringung und Umfang.

Aus einem übersichtlichen Design folgt unweigerlich die Benutzbarkeit des Programms (Hundhausen et al. 2002, S. 5).

Steuerung der Simulationsaktivität

Durch Übergabe spezifischer Kontrollmuster kann ein Nutzer die Simulation und den visualisierten Prozess für sich erschließen (Hundhausen et al. 2002, S. 6).

Die Möglichkeit, die Simulation zeitlich zu steuern, ist von großer Relevanz. So sollen Visualisierungen über „Start“, „Stopp“, „Pause“, „Vorwärts“ und insbesondere „Rückwärts“ lenkbar sein. Dies offeriere dem Nutzer die Möglichkeit, zunächst unverständliche Schritte wiederholt oder langsamer zu betrachten. Unterstützung finde dieser Prozess durch Möglichkeiten zum Heran- und Herauszoomen, sowie die Justierung des Bildbereichs (Hundhausen et al. 2002, S. 33)⁶. Dies versetze SuS in die Lage, Schlüsselmomente präzisiert betrachten zu können (Rößling und Naps 2002, S. 2).

Aufbau des Programms

Die Benutzeroberfläche eines Programmes kann eine größere Herausforderung für das Verständnis eines Problems als die Simulation selbst sein (Hundhausen et al. 2002, S. 33). Das Entwurf der Visualisierung, also die Entscheidungen, welche Objekte zu einem bestimmten Zeitpunkt an einem spezifischen Ort in welcher Form auftreten, gibt Bedarf zur Evaluation.

Das Konzept des „multiview editing“ (Musumbu 2014, S. 3) kombiniert im Hauptfenster des Programms eine Sicht auf die Szenerie, als auch auf den Quellcode des analysierten Texts. Thematisch angepasst kann analog zum Quellcode die Turing Maschinenprogrammierung in

⁵ Hundhausen zitiert nach Price, Baecker & Small 1993.

⁶ Hundhausen zitiert nach Stasko et al. 1993.

Form eines Zustandsdiagramms betrachtet werden. So kann eine graphische Entwicklung aufgezeigt werden.

Sprache

Das Benutzen eines Programms in der Schule kann eine Vielzahl von Probleme hervorrufen. Viele in Schulen einsetzbare Programme sind nur mit englischem Schriftbild verfügbar. Auch wenn die Erwartungen an die Schülerinnen und Schüler der Sekundarstufe II im Bereich der Anglistik hoch sind, können Schwierigkeiten in diesem Bereich auftreten. Programme, welche nicht in der Muttersprache des Nutzers geschrieben sind, würden in vielen Fällen nicht genutzt werden (Musumbu 2014, S. 4).

Umfang und Dokumentation

Die Dokumentation richtet sich vorrangig nicht an die SuS, sondern an den Lehrer, der in Erwägung zieht, eine Visualisierung im Rahmen seines Unterrichts einzusetzen. Es sei einer Lehrperson nicht zumutbar, in der Freizeit oder Unterrichtsvorbereitung eine intensive Analyse eines Programms auf seine Tauglichkeit durchzuführen (Lauterbach 1986).

Die Dokumentation solle Hinweise geben auf Korrektheit des Programms, notwendiges Hintergrundwissen, Möglichkeiten und Hinweise zur Nutzung der Visualisierung, sowie Beispiele (Staudacher 1985, S. 12).

Verfügbarkeit

Interessant für den Einsatz im Unterricht ist weiterhin die Verfügbarkeit der Visualisierung. Effizient sind Programme, die i) im Browser oder ii) als .exe ausführbar sind. Mit Abstrichen sollen Simulationen betrachtet werden, die individuell auf jedem Rechner installiert werden müssen, auch mindere dieser Prozess die Attraktivität des Simulators (Shaffer et al. 2007, S. 152). Interessant, auch wenn die Entwicklung wohl noch in seiner Anfangsphase steckt, sind mobile Applikationen für Smartphones oder Tablet-PCs. Genauere Informationen liegen hierzu jedoch nicht vor. Zudem muss die Programmierumgebung betrachtet werden: Läuft das Programm nur mithilfe älterer Toolkits? Ist es auf jedem zur Verfügung stehenden Rechner ausführbar? Um eine möglichst große Gruppe anzusprechen, sollte das Programm als Java-Applikation verfügbar sein (Rößling und Naps 2002, S. 1). Wie gestalten sich die Hardware und Software-Anforderungen? Zu letzterem muss jedoch festgehalten werden, dass diese Frage in der Regel keine aktuellen Probleme mehr darstellt. Dies ist zurückzuführen auf die beachtliche Entwicklung, die Computer und ihre Hardware genommen haben.

Interaktion vs. Animation

Wie in Teil 1.3 angedeutet, existieren deutliche Unterschiede zwischen einer Animation und einer Simulation. Während eine Animation auf die Veranschaulichung eines Prozesses abzielt und keine Möglichkeiten der Manipulation der vom Programmierer festgelegten Eingabe zulässt, enthalten die hier als Simulationen oder Visualisierungen bezeichneten Modelle explizit Mittel zum selbstständigen Arbeiten mit diesen. Testpersonen, die eine Simulation benutzen durften, zeigen ein signifikant besseres Verständnis für die behandelten Thematiken auf, als Personen der Testgruppe, welche mit reinen Animationen arbeiteten (Hundhausen et al. 2002, S. 13)⁷. Basierend auf dieser Interaktion finden sich unter dem Aspekt Freiheitsgrade Verzweigungen dieser Thematik.

Freiheitsgrade

Das Schulfach Informatik ist prädestiniert, Schülerinnen und Schülern gewisse Freiheitsgrade einzuräumen, denn nur selten kann von einer Singularität der Lösung gesprochen werden. Vielmehr existieren eine Vielzahl einfacher und komplexer Ergebnisse. Auch bei der Nutzung von Visualisierungen sollen diese Freiheitsgrade eingeräumt werden.

So sei das Erstellen eigener Eingaben von enormer Bedeutung für den Lernzuwachs eines Nutzers (Hundhausen et al. 2002, S. 13)⁸. Hierbei beachtet werden muss, dass dies auch zu einer Überforderung oder ungewollt zeitlichen Streckung führen könne (Rößling und Naps 2002, S. 1). Zudem können Aufgaben in Form von Vorhersagen oder Fragen (problematisch dabei: Erraten der Antwort) in den Ablauf des Programms eingewoben werden. So werden Fehler oder korrekte Antworten direkt durch den Simulationsablauf beantwortet und das Interesse der SuS aufrecht erhalten (Rößling und Naps 2002, S. 2). Auch in diesem Punkt sei ein deutlicher Lernzuwachs festgestellt (Hundhausen et al. 2002, S. 13)⁹.

Lenkung der Aufmerksamkeit

Elementare Prozesse seien textuell oder visuell hervorzuheben, um die Entwicklung der Thematik zu verdeutlichen. Das Aufzeigen konzeptioneller Schritte effektiviere den Wissenszuwachs (Hundhausen et al. 2002, S. 13)¹⁰.

⁷ Hundhausen nach zitiert Lawrence 1993.

⁸ Hundhausen nach zitiert Lawrence 1993.

⁹ Hundhausen zitiert nach Bryne, Catrambone & Stasko 1996.

¹⁰ Hundhausen zitiert nach Lawrence 1993.

Hieraus folgt, dass alle wichtigen Konzepte wiedergegeben und als solche wahrgenommen werden können. Zu beachten ist hier, dass farbige Hervorhebungen einen negativen Einfluss auf den Lernprozess haben können (Hundhausen et al. 2002, S. 15)¹¹.

Zeit

Die Norm einer Unterrichtsstunde beinhaltet eine Zeitspanne von 45 beziehungsweise 60 Minuten, im Rahmen einer Doppelstunde auch 90 bis 120 Minuten.

Somit sollte der Einsatz eines Mediums den Rahmen von circa 30 Minuten unter Betrachtung des Mehrphasensystems einer Unterrichtsstunde nicht überschreiten. Deduktiv hieraus ist zu schlussfolgern, dass die verwendeten Systeme eine schnelle Einarbeitung ermöglichen sollen, bestenfalls durch ein unkompliziertes Design gestaltet sind und als selbsterklärend titulierte werden können. Zudem sei die Vergeudung von Zeit durch „unnötige“ Animationen oder Einblendungen zu unterbinden (Staudacher 1985, 1985, S. 13).

Praktische Einsetzbarkeit

Ziel einer guten Visualisierung sollte die Verwendung durch Schüler, sowie durch Lehrer sein. Somit wird das Programm als Lern- und Lehrgegenstand eingesetzt (Musumbu 2014, S. 2). Aus Lehrersicht ist die Möglichkeit der Demonstration mithilfe eines Projektors oder Smart-Boards attraktiv, um möglichst alle SuS zu erreichen. Voraussetzung hierfür ist, dass die Simulation einen einfachen Verständnisaufbau ermöglicht. Da die Arbeit in einer Schule selten dem geplanten strikten Ablauf folge und von vielen Störvariablen beeinflusst werde, ist Flexibilität im Einsatz notwendig (Dietz und Jodl 1985, S. 19). So sind verschiedene Einsatzgebiete vorstellbar: Übungen, Projekte, Gruppenarbeiten bis hin zu Klausuraufgaben.

Weiterhin soll betrachtet werden, ob es sich bei dem System um ein „general-purpose System“ handle. Diese verkürzen die Einarbeitungszeit im Unterricht drastisch, da die Arbeit mit ihnen durch häufiges Nutzen zu verschiedenen Zwecken einstudiert wurde (Rößling und Naps 2002, S. 1). Alternativ müsse die Einarbeitungszeit in das Programm betrachtet werden, die notwendig ist, um einen „sicheren Umgang“ mit dem System zu gewährleisten (Latzina und Wedekind 1986, S. 39).

¹¹ Hundhausen zitiert nach Lawrence 1993.

Stabilität

Eingabesets und Steuerungen des Programms sind geeignete Mittel für Simulationsmodelle, erschweren aber die Arbeit des Programmierers. Um die Funktionalität des Programms zu gewährleisten, müssen Fehlersets und falsche Eingaben abgefangen und kommentiert werden (Dietz und Jodl 1985, 1985, S. 18). Es sollte auch beachtet werden, ob der gestartete Programmablauf anhaltbar und veränderbar ist.

Besonderheiten:

Dieser Punkt dient der Erwähnung hervorstechender interessanter Ideen, welche nicht von den bisherigen Kriterien erfasst werden.

Die folgende Tabelle zeigt eine beispielhafte Orientierung für die Erfüllung der Kriterien auf. Zudem werden pro Kriterium bis zu 5 Punkte vergeben, wobei 0 Punkte eine mangelhafte Bearbeitung der Kriterien und 5 Punkte eine sehr gute Bearbeitung ausdrücken. Die Kriterien „Verfügbarkeit, Sprache und Besonderheiten“ werden nicht einzeln bewertet, jedoch sollten die in Abschnitt 2.1 skizzierten Abwägungen getroffen werden.

Kriterium	Mögliche Einzelbetrachtungen	Wertung
Identifikation und Benennung aller relevanten Elemente und Ereignisse	-wichtige Elemente -wichtige Ereignisse -Erklärungen	5
Klarheit und Angemessenheit	-Struktur & Einheitlichkeit -Möglichkeiten & Grenzen -Konsistenz -Anforderungsniveau / Adressierung -Thematische Durchdringung -Umfang	5
Steuerung der Simulationsaktivität	-Schlüsselsteuerung -Zoomen -Justierung	5
Aufbau des Programms	-Design -„multiview editing“	5
Stabilität	-Fehler & Überlastungen -Erläuterungen	5
Umfang	-Wissensvermittlung -Beispiele	5
Praktische Einsetzbarkeit	-Demonstration -Arbeitswerkzeug -Flexibilität -bekanntes System	5
Interaktion vs. Animation	-Interaktion möglich	5
Freiheitsgrade	-Eigene Eingabesets -punktuelle Fragensets	5

Lenkung der Aufmerksamkeit	-Hervorheben konzeptioneller Schritte	5
Zeit	-Zeitraumen einhaltbar?	5
Verfügbarkeit	-Ausführbar -Browser -Installation nötig -Java-Applikation -Software + Hardware	
Sprache	-Sprachausgabe	
Besonderheiten	-Interessante, förderliche Aspekte	

Tabelle 2: Auflistung der als relevant erkannten Kriterien zuzüglich möglicher Einzelbetrachtungen.

Als Ergebnis der Analyse wird keine Gesamtpunktzahl ausgegeben. Dies hat zweierlei Gründe: Auf der einen Seite müsste eine Gewichtung der einzelnen Kategorien erfolgen, welche eine Wertung des Autors voraussetzt. Diese erzwungene Subjektivität erscheint nicht zielführend. Auf der anderen Seite sollte die Auswahl einer Simulation auf die Lern- und Lehrsituation abgestimmt werden. Der Verzicht auf die Gesamtwertung wird durch ein 5 Punkte-System ersetzt. So können Modellierungen kriterienorientiert ausgewählt und eingesetzt werden.

3.2 Betrachtung existierender Visualisierungen der Turing Maschine

Die Suche nach einer passenden Simulation ist begrenzt durch die Anzahl existierender Umsetzungen. Der Anspruch an diese Arbeit ist es, eine Vielzahl dieser Produkte zu identifizieren. Es muss trotzdem festgehalten werden, dass die Tabelle 2.3 keinen vollständigen Überblick bietet, sondern nur eine Momentaufnahme der aktuellen Suchergebnisse (Juni 2015) mithilfe von Internetsuchmaschinen darstellt.

Name	Version	Ursprung
Java Computability Toolkit	V1.2	http://www3.cs.stonybrook.edu/~algorithm/implement/jct/implement.shtml
Turingkara	Java 6	http://www.swisseduc.ch/informatik/karatojava/turingkara/
Turing-Simulator	V0.9	http://informatik.bildung-rp.de/werkzeuge-und-software/automaten-Turing_Maschinen.html
Tursi	V1.1	http://ais.informatik.uni-freiburg.de/tursi/
JFlap	V7.0	http://www.jflap.org/framebody.html
Turing machine simulator	kA	http://morphett.info/turing/turing.html
Turing machines implemented in JavaScript	kA	http://www.turing.org.uk/turing/scrapbook/tmjjava.html

Turingsimul	kA	http://www.iti.fh-flensburg.de/lang/theor/accept/turingsimul.htm
Turing Maschinen-Emulator	kA	http://wwwmath.uni-muenster.de/logik/Personen/Schulz/WS13/tm.html
Turing Maschinen-Simulator	kA	http://www.haberstroh.at/index.php?option=com_content&view=article&id=270:Turing_Maschinen-simulator&catid=39:programmierblog&Itemid=72
Einband Turing Maschine	kA	http://Turing_Maschine.klickagent.ch/einband/?lang=de#_-
xTuringMachine	kA	http://math.hws.edu/TMCM/java/index.html
TM2.0	V2.0	https://www2.informatik.hu-berlin.de/~verbuech/code/TuringMachine/applet/
Tmjava	k.A	http://www.turing.org.uk/book/update/tmjava.html

Tabelle 3: Liste verfügbarer Simulatoren, Seiten geprüft am: 14.06.2015.

Die Tabelle 3 listet alle gefundenen Simulationen auf, die i) online oder offline verfügbar sind und ii) funktionale Flash-Player verwenden. Einige Simulationen wurden bereits aussortiert, da sie diese Bedingungen nicht erfüllten. Andere Simulationen sind sehr ähnlich aufgebaut oder bieten zu wenig Inhalt für das oben aufgeführte Analyseschema. Aus diesem Grund erscheint eine Selektion bezüglich Polymorphie und Diversität zielführend.

Es werden die folgenden Modellierungen betrachtet:

Turingkara JFlap

Turing-Simulator <http://morphett.info/turing/turing.html>

Turing-Simulator, herausgegeben von Patrick Breuer, 30.03.2005, bereitgestellt über http://informatik.bildung-rp.de/werkzeuge-und-software/automaten-Turing_Maschinen.html

Die vorliegende Turing-Simulation ist eine 596 KB große .exe-Datei, welche in Kombination mit einer vierseitigen Kurzanleitung und sechs Beispielsprogrammen heruntergeladen werden kann. Das Gesamtpaket umfasst als .zip 371 KB.

Der internen Information ist zu entnehmen, dass das Programm zu unterrichtlichen Zwecken genutzt, kopiert und weitergegeben werden darf. Es liegt Version 0.9 des Programms vor.

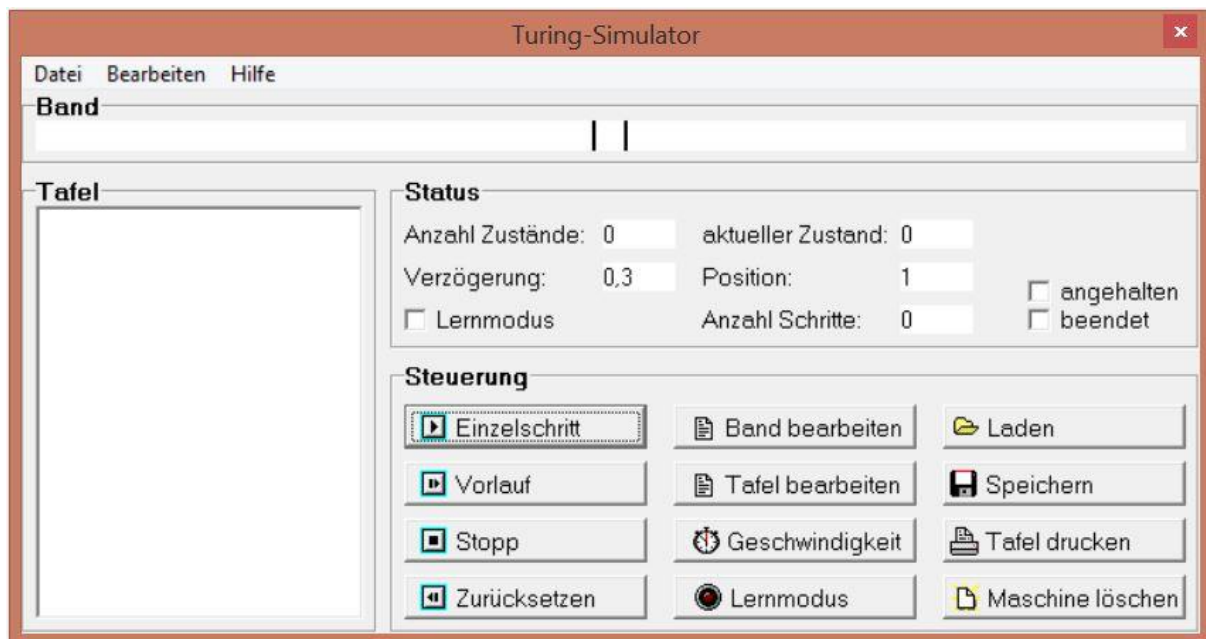


Abbildung 4: Turing-Simulator unmittelbar nach Start des Programms.

Kriterium	Begründung der Leistungsfähigkeit	0-5
Identifikation und Benennung aller relevanten Elemente und Ereignisse	Es werden alle relevanten Bestandteile aufgezeigt, das Zustandsdiagramm ist in Form einer Tafel festgehalten. Alle Zustände sind erkennbar, wünschenswert ist jedoch ein Aufzeigen der Start und Stoppzustände. Die Hilfeinstellungen sind noch nicht verfügbar, aufgrund des Datums der letzten Version (2005) ist von keiner Aktualisierung auszugehen.	3
Klarheit und Angemessenheit	Das Programm macht einen sehr geordneten Eindruck, alle verfügbaren Funktionen sind in dem Hauptmenü sortiert aufgeführt. Die Bedeutungen aller Knöpfe sind, abgesehen von dem Lernmodus, offensichtlich. Die Verwaltung der Eingaben folgt einer klaren Struktur, die Darstellung größerer Programme wird jedoch schnell unübersichtlich. Das Vorgehen zur Erstellung einer neuen Maschine erklärt sich durch Ausprobieren oder Lesen der Dokumentation. Negativ fällt auf, dass das „Befehlsdesign“, immerhin ein 5-Tupel-System, nicht intern erklärt wird. Nach kurzer Einarbeitungszeit und Erkennen der Struktur der Befehle sollte das Programm für Schüler handhabbar sein.	3
Steuerung der Simulationsaktivität	Die Steuerung der Simulator Aktivität erfolgt mithilfe von Start, Einzelschritt, Stopp und Wiederholung. Auch kann die Geschwindigkeit variiert werden. Es fehlt die Option, den vorherigen Schritt rückgängig zu machen. Aufgrund des gewählten starren Designs entfallen Optionen zum Zoomen, die Tafel kann nach Belieben justiert werden.	3

Aufbau des Programms	Das Design ist ansprechend, die einzelnen Elemente sind gruppiert und unabhängig. Die Fenstergröße ist festgelegt auf circa 850x450. Eine Vergrößerungsoption ist somit nicht vorgesehen. Es kann von einer Art „multiview editing“ gesprochen werden.	5
Stabilität	Das Programm läuft in der Regel stabil, ein gezielter Absturz konnte nicht herbeigeführt werden, jedoch trat ein solcher Fall mehrfach auf. Endlos-Schleifen werden nicht abgefangen, so wurde die Simulation nach circa 35.000 Schritten manuell beendet. ¹² Es wurde eine Fehlermeldung bei leerer Tafel (Tafel bearbeiten -> Befehl bearbeiten) nicht abgefangen.	2
Umfang	Als Dokumentation liegt eine Kurzanleitung zur fachgerechten Nutzung vor, welche eine Aufstellung der Funktionen beinhaltet. Eine theoretische Abhandlung ist nicht vorhanden. Mehrere Beispiele gehören dem Umfang des Programms an.	3
Praktische Einsetzbarkeit	Das vorliegende Programm kann zeitgleich mehrfach benutzt werden. Somit sind Quervergleiche möglich. Zudem können Programme gespeichert oder ausgedruckt werden, dies ermöglicht eine Verwendung in mehreren Unterrichtssituationen. Ein Kennenlernen des Programms ist zunächst nötig, da es sich um ein eigenständiges Programm ohne Vorgänger oder Analogsimulationen handelt.	3
Interaktion vs. Animation	Interaktion durch Bearbeitung der Tafel und des Bands möglich. Ein Bearbeiten der Parameter setzt das Programm zurück.	5
Freiheitsgrade	Die Erstellung eigener Eingaben ist als elementar für die Nutzung des Programms anzusehen. Der Nutzer erstellt eine eigene Eingabe für Band und Tafel. Zudem existiert Anschauungsmaterial, welches bearbeitet werden kann.	4
Lenkung der Aufmerksamkeit	Aktuelle Zustände werden im Layer Status angezeigt. Wünschenswert wäre ein Hervorheben der Zustandsgruppen oder des aktuellen Zustands. Diese werden als Ausgleich in einem eigenen Fenster angezeigt.	2
Zeit	Das selbstständige Programmieren gestaltet sich als strukturiert, jedoch resultiert aus einer großen Programmierung eine sehr lange Tafel. Durch das Laden von Programmen kann Zeit eingespart werden.	3
Verfügbarkeit	Download als .exe : Lokales Speichern auf jedem PC.	
Sprache	Deutsch.	
Besonderheiten	Lernmodus als kleiner „Assistent“ bei der Erstellung von Maschinen.	

Tabelle 4: Betrachtung des Turing-Simulators hinsichtlich der aufgestellten Kriterien

TuringKara, verfasst von Jürg Nievergelt, Werner Hartmann, Raimond Reichert, Markus Brändle, Tobias Schlatter, mit Beiträgen von Philipp Boksberger, Horst Gierhardt, Reto

¹² Probeprogramm schreibt nur das Zeichen „a“ auf das Band. Der Endloslauf stellt allerdings zunächst kein Problem dar, sondern ermöglicht neue Varianten von Programmen. Dennoch sollten Endlosschleifen zumindest durch Abfrage abgefangen werden, um eine Überlastung des Systems durch Unachtsamkeit zu vermeiden.

Lamprecht, Remo Meier, Horst Müller, Nicole Völki, Samuel Zürcher, 2003-2006, veröffentlicht über <http://www.swisseduc.ch/informatik/karatojava/turingkara/>

Kara ist eine, im Informatikunterricht, weit verbreitete Programmreihe zur Verständnisbildung bezüglich verschiedenster Thematiken. TuringKara kann als weiteres Produkt angesehen werden. Betrachtet wird die circa 3 MB große kara.jar-Datei. Diese setzt Java v.6 voraus. Zusätzlich steht ein Programm ohne Lösungen, sowie eine 200 KB große Anleitung zum Download bereit. Nach eigener Angabe werden Programme der Kara-Reihe bereits erfolgreich an Berufsbildenden Schulen, Gymnasien und Fachhochschulen eingesetzt.

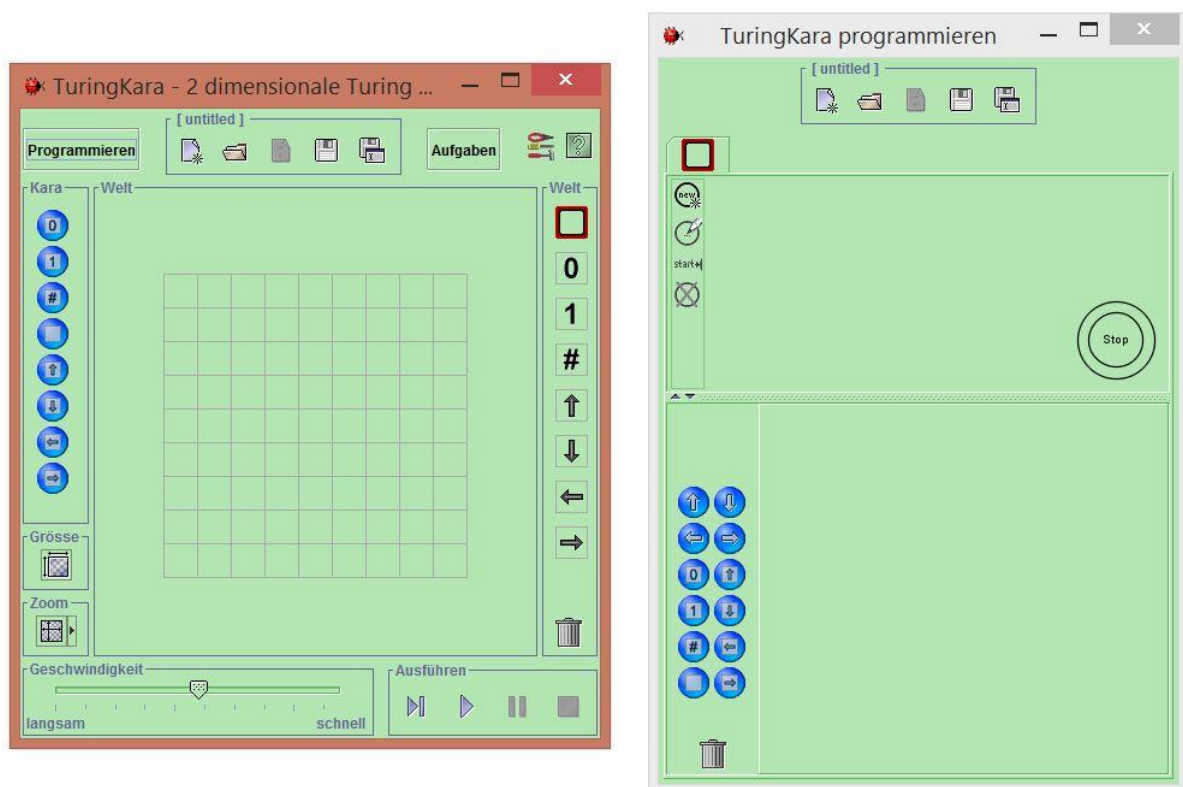


Abbildung 5: TuringKara unmittelbar nach Start des Programms.

Kriterium	Begründung der Leistungsfähigkeit	0-5
Identifikation und Benennung aller relevanten Elemente und Ereignisse	TuringKara erfasst alle relevanten Bestandteile der Turing Maschine. Wichtige Ereignisse, wie das Lesen, Schreiben oder die Angabe von Zuständen werden klar offengelegt. Die einzelnen Bestandteile sind mit Pop-up Fenstern unterlegt. Ein großer Makel existiert jedoch: Das Band mit n Feldern springt nach Betrachtung des Felds n und weiterführender Richtung auf Position 1 des Bands. Dies führt zu einer Verfälschung des Ergebnisses oder erfordert eine Justierung der Feldlänge.	3

Klarheit und Angemessenheit	Der Aufbau des Programms wirkt aufgeräumt, etwas ungewöhnlich erscheinen jedoch die zwei Layer „Kara“ und „Welt“, da beide ähnliche Symbole abbilden. Die Erstellung eines Programms wird vom Programm durch Hinweise und eine Hilfefunktion unterstützt. Das Programmieren-Layout funktioniert sehr eingängig und wird durch eine visuelle Darstellung unterlegt. TuringKara erlaubt eine große Anzahl einfacher und komplexer Aufgaben (UTM), das Anforderungsniveau kann demzufolge angepasst werden. Der Einsatz in Schulen erscheint möglich.	5
Steuerung der Simulationsaktivität	Steuern lässt sich TuringKara mittels Start, Einzelschritt, Pause und Zurücksetzen. Zudem gibt es die Möglichkeit zur Änderung der Geschwindigkeit. Eine Funktion, einen Schritt zurück zu gehen, existiert nicht. Die Größe des Felds kann vorgegeben werden, Funktionen zum Zoomen oder Verschieben des Felds existieren.	4
Aufbau des Programms	Das Design des Programms wirkt zunächst sehr schematisch. Dies fällt insbesondere bei der Vergrößerung des Programmfensters auf. Schön hingegen sind die Möglichkeiten zur Individualisierung von Kara, welche 3 Farbschema vorgeben. Viele Funktionen des Programms sind als eigenständige Pop-up Fenster konzipiert, hieraus resultiert ein Verlust an Kompaktheit. Das Konzept des „multiview editing“ wurde umgesetzt.	3
Stabilität	Das Programm läuft sehr stabil, auftretende Fehler werden erklärt und führen zu keinen Abstürzen. Endlosschleifen werden nicht abgefangen.	4
Umfang	Neben einem kurzen Benutzerhandbuch liegt eine theoretische Abhandlung von Katrin Moßbrucker vor. Zudem wird über die interne Hilfsfunktion das Arbeiten mit TuringKara verdeutlicht. Erwähnenswert ist der verfügbare Aufgabenbereich, welcher eine große Zahl an Exempeln inklusive Lösung bereithält.	5
Praktische Einsetzbarkeit	TuringKara ist polyvalent im Unterricht einsetzbar. Es können kleine Aufgaben und größere Projekte bearbeitet werden. Des Weiteren können Aufgaben direkt in das Programm geladen werden. Zudem kann Kara als „general-purpose“-System betrachtet werden, welches die Einarbeitungszeit der Schüler deutlich verkürzt.	5
Interaktion vs. Animation	Eine Interaktion wird durch Bearbeitung des Programms und des Bands möglich. Weiterhin kann direkt in das laufende Programm eingegriffen und Werte manipuliert werden. Ein Bearbeiten der anderen Parameter erfordert ein Zurücksetzen des Programms.	5
Freiheitsgrade	Die Nutzung von TuringKara kann auf vielen Ebenen erfolgen. So können neben Programmen und Eingaben auch existierende Aufgaben und Lösungen bearbeitet werden. Weiterhin kann zwischen einer Deterministischen und Nicht-Deterministischen TM gewählt werden.	4

Lenkung der Aufmerksamkeit	Der Ablauf des Programms wird nach Start auf zwei Ebenen aufgezeigt: Im Hauptfenster ist der Verlauf des Schreib- und Lesekopfs aufgezeigt, während das Programm markiert, welcher Zustand abgearbeitet wird und welche Option ausgeführt wird. Dieser Prozess wird farblich unterlegt.	5
Zeit	Das Arbeiten mit TuringKara variiert verständlicherweise von Aufgabe zu Aufgabe. Die Einarbeitungszeit ist hierbei jedoch nicht hoch, die Programmierung lässt sich angenehm und einfach vornehmen.	4
Verfügbarkeit	Download als .jar: Ausführung lokal notwendig. Es liegen ältere Versionen vor.	
Sprache	Deutsch und Englisch.	
Besonderheiten	Zwei Versionen: Aufgaben mit und ohne Lösung.	

Tabelle 5: Betrachtung von TuringKara hinsichtlich der aufgestellten Kriterien.

JFlap, unter Leitung von Susan Rodger seit 1990, veröffentlicht über <http://www.jflap.org/>

JFlap ist ein langfristig ausgelegtes Projekt, welches regelmäßig mit neuen Inhalten versorgt wird. Es handelt sich um ein graphisches Werkzeug, welches auch für die Abhandlung fundamentaler Inhalte der Automaten und Sprachentheorie dienlich ist. Betrachtet wird JFlap 7.0, welches kostenlos als .jar im Internet zur Verfügung gestellt wird. Die vorliegende Datei hat eine Größe von circa 11,4 MB. Auf der Website steht nebst anderer Informationen ein Tutorial für die Nutzung von JFlap zur Verfügung.

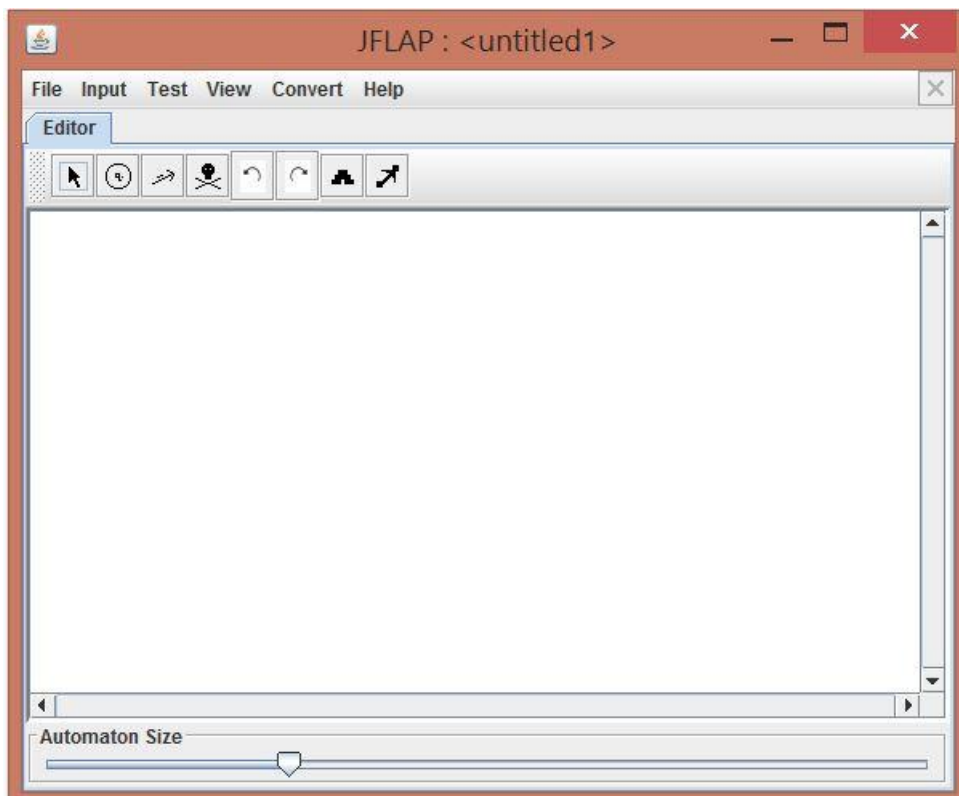


Abbildung 6: JFlap unmittelbar nach Start des Programms.

Kriterium	Begründung der Leistungsfähigkeit	0-5
Identifikation und Benennung aller relevanten Elemente und Ereignisse	Band, Schreib- und Lesekopf, Zustände und die Eingabe sind Elemente von JFlap. Start- und Stopp-Zustände sind markiert. Somit kann eine Turing Maschine vollständig abgebildet werden. Die Einführung der Bestandteile funktioniert schrittweise. Weiterhin sind alle wichtigen Ereignisse nachvollziehbar. Die Funktionalitäten sind mit Pop-up Fenstern hinterlegt.	4
Klarheit und Angemessenheit	JFlap wirkt wie ein Editor, der viele Freiheiten gewährt. Zu Beginn erscheinen die Funktionen und Aufgaben undurchsichtig. Viele Buttons sind trotz Hinterlegung mit Pop-ups wenig verständlich. Die Programmierung und Ausführung der Programme ist gut durchdacht und bietet viele Möglichkeiten. So können Turing Maschinen zu Blöcken zusammengefasst werden und hintereinander geschaltet werden. Dies unterstreicht die Mächtigkeit der Turing Maschine. Das Anforderungsniveau wird durch die vielen Funktionen (Multiple Run, Block Building, Combine Automata, etc.) gehoben.	3
Steuerung der Simulationsaktivität	JFlap erweitert die Funktionenvielfalt zur Steuerung: „Step, Reset, Focus, Defocus, Freeze, Thaw, Trace und Remove“ sind die Möglichkeiten für die einzelne Betrachtung einer Eingabe. Die Möglichkeit, einen Schritt rückgängig zu machen, ist auf etwas umständlichen Weg realisierbar. Separat werden Schnelltest-Möglichkeiten angeboten, im Gegenzug wurde auf eine Autoplay-Funktion verzichtet. Funktionen zum Zoomen oder Justieren sind vorhanden.	5
Aufbau des Programms	Das Design des Programms erinnert an einen Editor und wirkt funktionalorientiert aufgebaut. Die Symbole sind insbesondere im maximierten Bildfenster zu klein. Ein ähnlich geartetes Problem tritt auch bei der Simulation mit einer Eingabe bezüglich des Eingabebands auf. Negativ fällt die Möglichkeit auf, beliebig viele Zustände auf eine Position zu setzen, auf. „Multiview Editing“ wurde umgesetzt.	2
Stabilität	Das Programm läuft stabil, potenziell auftretende Fehler werden erklärt und führen zu keinen Abstürzen. Endlosschleifen erzeugen im „Fast Run“-Modus in Intervallen von 500, 1000, 2000 etc. Durchlaufen eine Ausgabe, ob das Programm weiter ausgeführt werden soll	5
Umfang	Nicht zuletzt aufgrund der sehr langen Lebensspanne von JFlap existiert eine große Sammelurium an Material: Tutorials, Geschichtliches, Bücher, Paper, Diskussionen, Workshops und Mustervorlagen, sowie ein eigenes Wiki. Es bleiben keine Wünsche offen.	5

Praktische Einsetzbarkeit	Die Möglichkeiten zum Einsatz von JFlap sind vielfältiger Natur. Beispielweise das Betrachten, Erstellen, Korrigieren oder Vergleichen einer bzw. mehrerer TM. Das Erzeugen großer Systeme dank der Block-Bildungsfunktion ist auch möglich. Aufgrund der vielen Möglichkeiten, die JFlap bietet (11 Simulationen in einem Programm) kann das Programm als „general purpose“ System in mehrere Unterrichtseinheiten eingesetzt werden.	5
Interaktion vs. Animation	Die Benutzung von JFlap setzt eine Interaktion mit dem Programm voraus. Nach Programmierung der TM kann entschieden werden, welche Art der Ausführung auf welcher Eingabe erfolgen soll. Einzelne Schritte können angehalten oder festgehalten werden. Es werden Möglichkeiten für Mehrband-TM offeriert.	5
Freiheitsgrade	JFlap steuert die Aktivitäten des Nutzers nicht, sämtliche Programme und Aufgaben müssen selbst überlegt und bearbeitet werden. Es existieren jedoch Beispiele, die zur Bearbeitung hochgeladen werden können. Das Programm bietet dem Nutzer einen beeindruckenden Spielraum.	5
Lenkung der Aufmerksamkeit	Nach Erstellen eines Zustandsdiagramms kann durch Auswahl des „Step“-Modus Schritt für Schritt die Bearbeitung der Programmierung betrachtet werden. Hierfür zoomt das Programm automatisch an die relevanten Stellen des Diagramms und hebt die aktuellen Zustände hervor.	5
Zeit	Die Einarbeitungszeit variiert von Aufgabe zu Aufgabe. Viele einfache, in der Schule relevante Probleme können sehr schnell geschrieben und ausgeführt werden. Das vollständige Ausschöpfen der Möglichkeiten gestaltet sich verständlicherweise aufgrund der großen Vielfalt als zeitaufwändiger.	4
Verfügbarkeit	Download als .jar: Ausführung lokal notwendig. Es liegen ältere Versionen vor.	
Sprache	Englisch (teils über Standardvokabular).	
Besonderheiten	Block-Bildung, Testoptionen, Layout-Optionen, Mehrband-TM, verschiedene Ausführloptionen.	

Tabelle 6: Betrachtung von JFlap hinsichtlich der aufgestellten Kriterien.

Webbasierte Turing Maschine, erstellt von Anthony Morphett, verfügbar über <http://morphett.info/turing/turing.html>.

Anthony Morphett ist Dozent für Mathematik an der Universität von Melbourne. Durch sein Interesse an Theoretischer Informatik entwickelte er den Simulator in Javascript. Als einzige betrachtete Visualisierung handelt es sich hier um eine im Internet gehostete Version. Der Quellcode ist über die Codehosting-Plattform GitHub einsehbar.

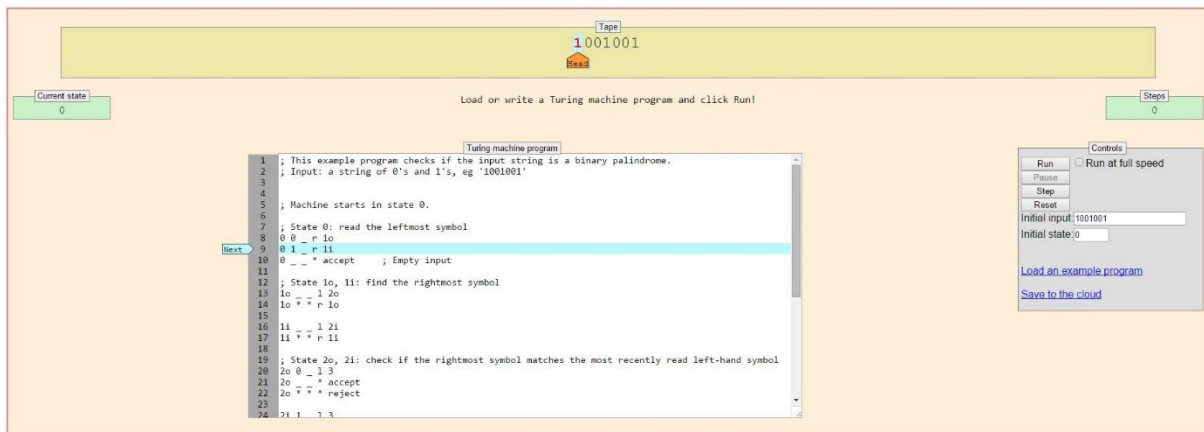


Abbildung 7: Ausschnitt von Morphetts Simulator unmittelbar nach Öffnen der Website.

Kriterium	Begründung der Leistungsfähigkeit	0-5
Identifikation und Benennung aller relevanten Elemente und Ereignisse	Die vorliegende Simulation erfasst alle als relevant betrachteten Bestandteile und Ereignisse zum Abbilden einer Turing Maschine. Erklärungen bezüglich des Ablaufs und der Syntax des Programms sind gegeben. Interessant im Vergleich zu anderen Visualisierungen ist das freie „Programmieren“ der TM in einem Textfeld. Hierbei sind Kommentare erlaubt.	5
Klarheit und Angemessenheit	Die Website offeriert von Beginn an alle verfügbaren Möglichkeiten. Aufgrund der verfügbaren Erklärungen wird schnell klar, wie das Programm aufgebaut wurde und funktioniert. Das Arbeiten mit der Visualisierung erfordert zunächst etwas Zeit, um die Syntax der Modellierungssprache zu erlernen, da die Eingabe über kein festes Schema erfolgt. Zudem steigt die Unübersichtlichkeit der Programme proportional zu seiner Größe. Aufgrund der oben genannten Eigenschaften ist der Einsatz in Schulen vermutlich eingeschränkt möglich.	2
Steuerung der Simulationsaktivität	Die Möglichkeiten zur Kontrolle des Programms sind nur „Run, Pause, Step und Reset“. Möglichkeiten, einen Schritt rückgängig zu machen, fehlen. Eine Justierung des Blickfelds ist bezüglich der Befehlsliste und dem Band gegeben. Eine Funktionalität betreffend Zoomens ist nicht notwendig.	2
Aufbau des Programms	Trotz der Konzentration sämtlichen Inhalts auf die Hauptseite der Website wirkt der farblich abgegrenzte Simulator sehr zentriert. Aufgrund fehlendem Funktionsumfang ist die Simulation sehr aufgeräumt. Die Anpassung der Frame-Größen könnte besser auf die Größenänderungen des Browsers angepasst sein. Da die Programmierung gleichzeitig den Code des Programms darstellt, kann von „multiview editing“ gesprochen werden.	3

Stabilität	Die Simulation läuft stabil, es kam zu keinen Abstürzen bei ihrer Nutzung. Fehler in der Programmierung der TM werden bei der Ausführung korrigiert, die TM stoppt und gibt eine Begründung an. Parallel läuft vor dem Start des Programms ein Test bezüglich der Verschiedenheit der Zustände. Endlosschleifen werden nicht abgefangen.	5
Umfang	Morphetts Turing Maschine enthält keine Dokumentation im eigentlichen Sinne. Über und unterhalb des Programms sind kurze Erklärungen bezüglich der Benutzung der Website angegeben. Fester Bestandteil der Simulation ist die Möglichkeit, fertige Programme zu laden und zu speichern. Der Speicherprozess findet im Internet statt, es werden keine lokalen Daten erzeugt, aber ein Link generiert. Der Quellcode wird frei zugänglich über GitHub angeboten.	3
Praktische Einsetzbarkeit	Die Möglichkeiten für den Einsatz im Unterricht sind ähnlich wie die der anderen Simulationen. Beachtet werden muss hier jedoch, dass ein Herunterladen der .html-Datei der Seite Funktionen wie das Laden der Beispielprogramme unterbindet. Die Ausführung ist weiterhin möglich, jedoch ist eine stetige Internetverbindung von Vorteil. Positiv ist die Kommentarfunktion innerhalb des Codes.	3
Interaktion vs. Animation	Die Interaktion mit Morphetts Simulator erfolgt über direkten Eingriff in den Programmtext der Maschine. Dies kann auch während eines laufenden Programms erfolgen. So kann eine Maschine nach Belieben manipuliert und Überarbeitet werden.	5
Freiheitsgrade	Alle Programme und Aufgaben müssen selbst überlegt und bearbeitet werden. Die vorliegenden Beispielprogramme können als Grundlage von Arbeiten genutzt werden.	5
Lenkung der Aufmerksamkeit	Der Simulator weist den Nutzer auf mehreren Ebenen auf den aktuellen Ablauf des Programms hin. So ist die Bewegung des Schreib- und Lesekopfs, der aktuelle Zustand und die Bearbeitung innerhalb des Programms aufgezeigt. Verbesserungswürdig ist hierbei jedoch, dass die Programme oft sehr groß sind und mit einer Bildlaufleiste ausgestattet sind, welche nicht automatisch mit dem Verlauf des Programms verschoben wird.	4
Zeit	Die Arbeit mit dem Programm ist aufgrund des recht hohen Schreibaufwands durch Verfassung sämtlicher Zustände inklusive zugehöriger Übergänge aufwändiger. Trotzdem ist die Syntax des Turing-Codes sehr einfach gehalten und erlernbar. Die Einarbeitung kann nichtsdestotrotz eine gewisse Zeitspanne beanspruchen.	3
Verfügbarkeit	Website, ältere Version verfügbar, .html-Datei herunterladbar, Quellcode einsehbar.	
Sprache	Englisch.	
Besonderheiten	Keine Besonderheiten	

Tabelle 7: Betrachtung des webbasierten Turing Maschinen Simulators hinsichtlich der aufgestellten Kriterien.

4. Der Entwicklung einer schulisch wertvollen Turing Maschine

In den Abschnitten 1 und 2 der Bachelorarbeit wurden i) Literatur der Fachpresse und Forschung und ii) Umsetzungen einer Modellierung betrachtet. Es soll nun versucht werden, den Aufbau eines für die Schule interessanten Simulators der Turing Maschine zu konstruieren. Es könne an dieser Stelle auf die Kriterien-Liste hingewiesen werden, welche viele wichtige Faktoren für Simulatoren umfasst. Anstelle dieser Vorgehensweise wird versucht, auf Basis der Kriterien sowie neuer Erkenntnisse ausgehend von der vorherigen Betrachtung und Analyse diverser Simulationen Schwierigkeiten und Möglichkeiten aufzuzeigen.

Um eine Vergleichbarkeit zu den existierenden Simulationen zu schaffen, dienen die Kriterien der Orientierung. Es findet eine Gruppierung in vier Teilbereiche statt:

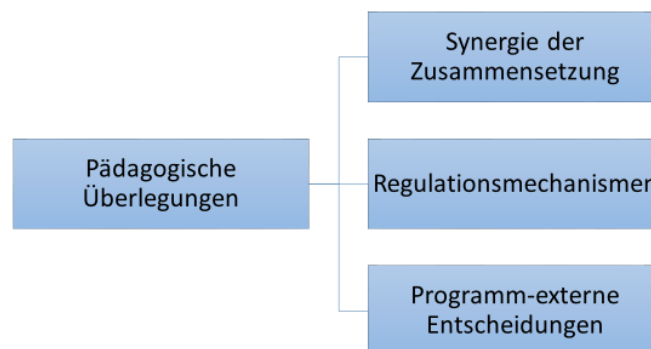


Abbildung 8: Gliederung der Teilgebiete

Synergie der Zusammensetzung	Identifikation und Benennung aller relevanten Elemente und Ereignisse Klarheit und Angemessenheit Aufbau des Programms Lenkung der Aufmerksamkeit
Regulationsmechanismen	Steuerung der Simulationsaktivität Interaktion Stabilität
Programm-externe Entscheidungen	Sprache Umfang Verfügbarkeit
Pädagogische Überlegungen	Zeit Praktische Einsetzbarkeit Freiheitsgrade (Besonderheiten)

Tabelle 8: Sortierte Kriterien Liste hinsichtlich verschiedener Orientierungen

Synergie der Zusammensetzung:

Sie umfasst sämtliche Entscheidungen bezüglich der Zusammenstellung visueller Elemente innerhalb des Programms. Einige Entscheidungen sind einfach, beispielsweise die Identifikation der relevanten Elemente und Ereignisse.¹³ Insbesondere hier können Fehler gravierende Auswirkungen auf den späteren Lernerfolg haben, da fehlerhafte Grundbausteine zu einer konzeptionell falsch ausgeprägten Sichtweise führen. Hin und wieder in Vergessenheit geraten Tooltips, welche aber für das Verständnis verfügbarer Optionen von großer Bedeutung sind. Aufgrund des sehr geringen Aufwands zur Erstellung dieser sollten sie Bestandteile jedes Programms sein. Des Weiteren wurde die Klarheit und Angemessenheit der Simulationen betrachtet. Hier konnten nicht alle Programme vollständig überzeugen. Probleme konnten hier insbesondere bei der Klarheit lokalisiert werden. Auffallend oft war nicht ersichtlich, welche Funktionalität durch welchen Knopf präsentiert wird, Erkenntnis lieferte hier erst die (vorhandene) Dokumentation. Weiterhin fiel in mehreren Programmen die Eingabeverwaltung negativ auf. Generell sollten die Funktionen insbesondere im Bereich der Nutzerschnittstellen ausreichend erklärt und dokumentiert werden. Die Betrachtung des Designs erbrachte die Erkenntnis, dass das Programm bestenfalls vollständig in einem Frame untergebracht werden sollte. Pop-Ups stellen grundsätzlich kein Problem dar, jedoch sollte hierbei der quantitative Maßstab ausschlaggebend sein. So sollte das Programm in jeglicher Hinsicht nicht überladen wirken. Eine etwas unwichtiger anmutende Entwurfsentscheidung betrifft die Skalierung einzelner Bereiche insbesondere bei der Maximierung des Simulators. Einige Simulatoren verloren hier stark an Kompaktheit und Ordnung. Ein sehr wichtiger Faktor für die Konstruktion einer guten Simulation ist die Hervorhebung elementarer Prozesse. Dies wurde in den Programmen in der Regel gut gelöst. Die Lösung der Simulatoren TuringKara und JFlap erscheinen hier als bestens geeignet, da diese sowohl Programmierung als auch Verlauf synchron verarbeiten.

Regulationsmechanismen:

Dieser Punkt umfasst die Nutzerinteraktion mit der Simulation über die bereitgestellten Funktionen. Hier können verschiedene Ebenen existieren. Zunächst wird betrachtet, wie die Steuerung der Simulation innerhalb des Programms erfolgt. Hier greifen alle Simulationen auf ein Basis Set, bestehend aus „Schritt, Durchlauf, Pause und Zurücksetzen“ zurück. JFlap erweitert dieses Set um weitere Funktionen: So können wichtige Konfigurationen angewählt

¹³ Aussagen bezüglich der Komplexität einer Operation werden basierend auf der erreichten Punktzahl bzw. begangener Fehleinschätzungen, welche in 2.2 identifiziert wurden, getätigt.

und festgehalten werden. Die Wichtigkeit einer „Schritt zurück“-Funktion wurde in 2.2 behandelt. JFlap kommt dieser mit dem erweiterten Steuerungs-Sets recht nahe, bei den anderen Simulationen fehlt diese vollständig. Das Zoomen und Justieren innerhalb des abgebildeten Modells dient der individuellen Verständnisbildung. Sofern nötig, waren diese Funktionen in ausreichendem Maße verfügbar. Die Einsetzbarkeit einer Simulation wird zudem stark davon beeinflusst, wie stabil das Programm läuft. Erfahrungsgemäß ist die Stabilität dann gefährdet, wenn ein Nutzer mit diesem interagiert. Dies geschieht trivialerweise über die zur Verfügung gestellten Schnittstellen. Insbesondere an diesen Stellen sollte der Fokus liegen. Weiterhin sollten Überlegungen hinsichtlich der Handhabung von Endlosschleifen gemacht werden. Eine Abfrage, ob das Weiterführen der Schleife beabsichtigt ist, wie in JFlap, erscheint vorteilhaft.

Programm-externe Entscheidungen:

Zu Beginn der Programmierung eines Simulators müssen bestimmte Vorüberlegungen stattfinden hinsichtlich der Sprache und Verfügbarkeit. An späterer Stelle können Überlegungen bezüglich des Umfangs erfolgen. Die Entscheidung für eine Sprache sollte auf Abwägung der Zielgruppe stattfinden. Soll das Programm in deutscher Sprache verfasst sein, erstreckt sich die Zielgruppe zunächst nur auf den deutschsprachigen Raum. Fällt die Entscheidung zugunsten des Englischen, sollte auf komplexe Formulierungen verzichtet werden. Dies stellt einen direkten Kritikpunkt an JFlap dar. Ideal für die Bereitstellung der Simulation wäre die Verfügbarkeit sowohl on- als auch offline. Da eine derartige Koexistenz in sämtlichen betrachteten Simulationen nicht existiert, scheint auch hier eine Entscheidung getroffen werden zu müssen. Der Verfasser bezieht hier Position bezüglich lokal verfügbarer Simulationen, da diese bedeutend professioneller erscheinen, auch wenn ein höherer Vorbereitungsaufwand mit diesen einhergeht. Der zur Verfügung gestellte Umfang sollte mit dem Programm skalieren. Hierzu ist es erforderlich, dass ein gutes Mittelmaß gefunden wird. Während Morphett keinerlei Dokumentation, dafür aber einige Beispiele kombiniert, verfügt JFlap über eine gewaltige Anzahl an Dokumentation, aber keinerlei intern zur Verfügung gestellter Beispiele. TuringKara enthält eine gute Koexistenz beider Daten.

Pädagogische Entscheidungen:

Als vierter Teilbereich treten getroffene pädagogische Entscheidungen in Erscheinung. Hierbei muss angemerkt werden, dass pädagogische Überlegungen selbstverständlich auch in die anderen Teilbereiche einfließen. An dieser Stelle werden solche aufgeführt, deren Zuordnung schwieriger fiel. Zu Beginn sollte das Hauptaugenmerk auf den Zeitaufwand gelegt werden,

den das Programm den Nutzern zur Einarbeitung oder Bearbeitung abverlangt. Eine Einführung in das Programm spart im Nachhinein viel Zeit. Dies war auch einer der Gründe für den Ausschluss der JCT. Weiterhin sollten unnötige Einblendungen und Wartezeiten vermieden werden, diese untergraben den Anteil effektiver Lernzeit. Diese zeitliche Betrachtung fortführend, fallen Programme wie Morphet oder der Turing-Simulator durch einen erheblich höheren Zeitaufwand bezüglich des Programmschreibens auf. Eine graphische Oberfläche dient neben der Zeitersparnis auch der Simplifizierung komplexerer Maschinen. Eine gewisse Schnittmenge zwischen der Zeit und der praktischen Einsetzbarkeit ist nicht von der Hand zu weisen. Dennoch ist die thematische Durchdringung der Einsetzbarkeit größer. Bestmöglich könnte das Programm auf verschiedene Unterrichtssituationen ausgelegt sein. Hierunter fallen Einzel, Gruppen und Projektarbeiten, aber auch viele kurze Betätigungen. Umsetzungen hierfür wurden in den Programmen aufgezeigt: Erstellen von Maschinen, Kombination mehrerer Maschinen (JFlap), Laden existierender TM-Programme oder Bänder, verfügbare Aufgaben (TuringKara) oder auch Betrachtungen mehrerer zeitgleicher Eingaben (JFlap). Hinzukommen Erweiterungen des Modells auf Mehrdimensionale Turing Maschinen oder die Betrachtung von Grammatiken wie in JFlap.

5. Abschließende Diskussion

Ausgehend von den vorausgegangenen Betrachtungen kann festgehalten werden, dass die Erstellung einer schulisch einsetzbaren und pädagogisch wertvollen Visualisierung ein anspruchsvoller Prozess ist. Sollte eine Visualisierung erstellt werden, bildet Abschnitt 2.1 eine möglichst allgemeingültige Taxonomie ab. Im vorgenannten Abschnitt wurden Kriterien basierend auf der einschlägigen Literatur herausgearbeitet und bilden somit eine beispielhafte Orientierung für den Modellbildungsprozess ab. Mögliche Fehlerquellen bei der Erstellung einer Simulation für Turing Maschinen wurden in Abschnitt 3.1 herausgestellt. Viele dieser Erkenntnisse sind jedoch für andere Modelle übertragbar.

Weiterhin kann die Bachelorarbeit Lehrkörpern helfen, eine geeignete Visualisierung zur Verwendung im Unterricht auszuwählen. Hierfür dient das in 2.2 herausgearbeitete Schema, welches die Kriterien exemplarisch für vier verschiedenartige Umsetzungen anwendet. Hierbei wird noch einmal dringend darauf hingewiesen, dass eine Gesamtwertung nicht zielführend ist. Vielmehr sollte die Auswahl basierend auf Vorwissen und Leistungsstärke, sowie Umfang der Themeneinheit angepasst werden. So ist für eine umfassende Behandlung von Thematiken wie beispielsweise „Modelle der theoretischen Informatik“ oder „Grenzen der Berechenbarkeit“ ein Rückgriff auf JFlap (insbesondere bei der beabsichtigten Behandlung weiterer Themen wie:

Endliche Automaten, Mealy & Moore Maschinen und Grammatiken) oder Turing-Kara empfehlenswert. Als Einführung in Turing Maschinen erscheint der Turing-Simulator ein probates Mittel abzubilden. Abschließend möchte der Autor darauf aufmerksam machen, dass der Einsatz der Turing Maschine in der Schule in einem deutlich zu geringem Ausmaß (sofern sie überhaupt thematisiert wurde) durchgeführt wird. Das Konzept der Turing Maschine erfasst viele thematische Bereiche und kann insbesondere im Rahmen der Verständnisbildung eine essentielle Rolle einnehmen. So behandelt es Zustandsmaschinen, Formalismus, Grammatiken und vieles mehr. Die Bedeutung der Turing Maschine für die Berechenbarkeit und Entscheidbarkeit führt prozessartig über interessante Thematiken wie der Gödelnummer. So kann eine große Fülle informatischer Inhalte bearbeitet werden. Kurz gesagt: Die Turing Maschine ist ein bemerkenswertes Modell und bildet eine wichtige Grundlage der informatischen Bildung.

Literaturverzeichnis

Berliner Landesinstitut (2006): Rahmenlehrplan für die gymnasiale Oberstufe für Schule und Medium et al. Hg. v. Senatsverwaltung für Bildung, Jugend und Sport Berlin, Ministerium für Bildung, Jugend und Sport des Landes Brandenburg und Ministerium für Bildung, Wissenschaft und Kultur Mecklenburg-Vorpommern. Online verfügbar unter https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1429785405&file=sek2_informatik.pdf, zuletzt geprüft am 15.06.2015.

Dietz, B.; Jodl, J. (1985): Didaktische Analyse von (Physik-)Unterrichtsprogrammen. In: *LOG IN* (2), S. 15–22.

Hilbert, D.; Ackermann, W. (2013): Grundzüge der theoretischen Logik: Springer-Verlag (27).

Hundhausen, C. D.; Douglas, S. A.; Stasko, J. T. (2002): A meta-study of algorithm visualization effectiveness. In: *Journal of Visual Languages & Computing* 13 (3), S. 259–290.

Kirchner, E. (1983): Bemerkungen zur Modellbildung und zum Modellbegriff. In: *LOG IN* (1), S. 12–15.

Klafki, W. (1964): Das pädagogische Problem des Elementaren und die Theorie der kategorialen Bildung: Beltz Weinheim.

Latzina, M.; Wedekind, J. (1986): Simulationsprogramme: Systematische Beschreibung und Bewertung. In: *LOG IN* (5/6), S. 35–41.

Lauterbach, R. (1986): Bewertung pädagogischer Software. In: *LOG IN* (5/6), S. 25–33.

Ministerium für Bildung, Familie, Frauen und Kultur Saarland (2008): Lehrplan Informatik. Hg. v. Ministerium für Bildung, Familie, Frauen und Kultur Saarland. Online verfügbar unter http://www.saarland.de/dokumente/thema_bildung/IN-GOS-2-Feb2008.pdf, zuletzt geprüft am 15.06.2015.

Modrow, E. (1991): Zur Didaktik des Informatik-Unterrichts. 1. Ziele und Inhalte. Anfangsunterricht. Beispiele und Anwendungen: Dümmler.

Modrow, E. (1992): Zur Didaktik des Informatik-Unterrichts. 2. Gesellschaftliche Auswirkungen. Fachunterricht. Abitur: Dümmler.

Musumbu, K. (2014): Algorithms Visualization Tool for Students and Lectures in Computer Science. In: *arXiv preprint arXiv:1403.4423*, S. 1–5. Online verfügbar unter <http://arxiv.org/pdf/1403.4423.pdf>, zuletzt geprüft am 18.06.2015.

Naumer, H.; Martin, S. (1983): Computersimulationen auch im Chemieunterricht? In: *LOG IN* (1), S. 26–31.

Niedersächsisches Kultusministerium (2012): Abitur 2015 – Thematische Schwerpunkte für das Fach Informatik. Hg. v. Niedersächsischer Bildungsserver. Online verfügbar unter http://www.nibis.de/nli1/gohrgs/13_zentralabitur/zentralabitur_2015/18InformatikTS2015.pdf, zuletzt geprüft am 15.06.2015.

Rößling, G.; Naps, T. L. (2002): A testbed for pedagogical requirements in algorithm visualizations. In: *ACM SIGCSE Bulletin*, Bd. 34. ACM, S. 96–100.

Rowland, T.: Church-Turing Thesis. Hg. v. Eric W. Weisstein. Mathworld--A Wolfram Web Resource. Online verfügbar unter <http://mathworld.wolfram.com/Church-TuringThesis.html>, zuletzt geprüft am 14.06.2015.

Shaffer, C. A.; Cooper, M.; Edwards, S. H. (2007): Algorithm visualization: a report on the state of the field. In: *ACM SIGCSE Bulletin*, Bd. 39. ACM, S. 150–154.

Shaffer, C. A.; Cooper, M. L.; Alon, A. J. D.; Akbar, M.; Stewart, M.; Ponce, S.; Edwards, S. H. (2010): Algorithm visualization: The state of the field. In: *ACM Transactions on Computing Education (TOCE)* 10 (3), S. 9.

Stachowiak, H. (1973): Allgemeine Modelltheorie.

Staudacher, L. (1985): Unterrichts-Software: Unlust an der Phantasie. In: *LOG IN* (2), S. 8–15.

Treitz, N. (1983): Gesichtspunkte zu den didaktischen Funktionen der Simulation auf dem Bildschirm-computer im Physikunterricht. In: *LOG IN* (1), S. 31–39.

Turing, A. M. (1936): On computable numbers, with an application to the Entscheidungsproblem. In: *J. of Math* 58 (345-363), S. 5.

Glossar

M-O Relation: Beziehung zwischen Modell und Objekt

M-S Relation: Beziehung zwischen Modell und Subjekt

Bohr'sches Atommodell: veraltetes Konzept bezüglich des Aufbaus von Atomen nach Niels Bohr.

MINT-Fächer: Bezeichnung für Fächer der Bereiche Mathe, Informatik, Naturwissenschaften und Technik.

Mehrphasensystem: Unterrichtsstunden-Aufbau basierend auf einzelnen charakteristischen Phasen

Frame: Teilbereich eines Programms

Button: Umsetzung der Funktionalität eines Knopfes innerhalb eines Programms

general purpose system: Ein System, welches für mehrere Begebenheiten eingesetzt werden kann.

multiview editing: Vereinigung von Quellcode und Animation im Hauptfenster des Programms.

Abkürzungsverzeichnis

TM: Turing Maschine

SuS: Schülerinnen und Schüler

Anhang

Begründung der Kriterien-Bewertung (Seite 5)

Inhalte der CD:

- Die vorliegende Arbeit im PDF-Format
- Referenzierte Programme (sofern digital vorhanden)

Anhang 1: Begründung der Kriterien-Bewertung (Seite 5)

1. Die Schüler können selbstständig in diesem Gebiet arbeiten.

Das Unterrichtsthema Turing Maschine bietet unter anderem durch Simulatoren Möglichkeiten, selbstständig tätig zu werden.

(Positiv)

2. Das Thema eignet sich für arbeitsteiligen Unterricht und/oder Projektarbeit.

Mit einer Turing Maschine können alle intuitiv berechenbare Funktionen abgebildet werden. Die Programmierung einer umfangreicheren Arbeit wäre eine Möglichkeit für arbeitsteiligen Unterricht oder Projektarbeiten.

(Positiv)

3. Das Thema ermöglicht längere selbstbestimmte Arbeitsphasen der Schüler/innen. Der Lehrer muss nicht kleinschrittig den Unterricht führen.

Bei Nutzung eines Simulators können die SuS über einen längeren Zeitraum selbstständig arbeiten.

4. Es gibt nicht nur eine Standardlösung für das Problem. Die Schülerinnen und Schüler (SuS) können das Problem auf unterschiedlichen Wegen lösen.

Die Programmierbarkeit einer TM ermöglicht eine Vielzahl von Lösungen eines Problems.

(Positiv)

5. Das Thema gestattet Spezialisierungen und Erweiterungen, so dass auf unterschiedlichem Niveau gearbeitet werden kann. Es ermöglicht durch die unterschiedlich komplexen Lösungen Binnendifferenzierung.

Das Modell der Turing Maschine bietet unter anderem Erweiterungen wie die Mehrband-TM oder Einseitig-beschränkte TM, welche verschiedene Formen der Problemlösung erfordern.

(Positiv)

6. Die Relevanz des Themas ergibt sich nicht nur aus seiner Bedeutung für die derzeitige Technologie.

Die Turing Maschine ist als theoretisches, nicht real umsetzbares Modell anzusehen. Eine Verbindung zwischen Modell und einer Technologie existiert nicht.

(Positiv)

7. Die Relevanz des Themas ergibt sich nicht nur aus Prognosen über seine zukünftige Bedeutung.

Siehe 6. (Positiv)

8. Aussagen über gesellschaftliche Folgen der Datenverarbeitung sind möglich, sie können sachlich fundiert behandelt werden.

Es wird kein Zusammenhang zwischen der TM und aktuellen gesellschaftlichen Folgen gesehen.

(Negativ)

9. Das Thema spricht prinzipielle Fragen an. Es beschränkt sich nicht nur auf eine aktuelle Mode.

Anhand der TM können unter anderem die Grenzen der Berechenbarkeit definiert werden. Die TM als Begleiterscheinung einer Mode zu sehen, ist falsch.

(Positiv)

10. Der Zeitbedarf zum Erlernen von Benutzerwissen steht in einem angemessenen Verhältnis zum angestrebten Lernerfolg (mit Einschränkung)

Diese Wertung ist abhängig von dem Lehrplan des jeweiligen Bundeslandes.

(Mit Einschränkung)

11. Die Ausgangsproblemstellung kann so einfach gehalten werden, dass die ohne großes Spezialwissen verstanden und bearbeitet werden kann. Der Zugang zum Thema ist also „naiv“ möglich. (mit Einschränkung)

Zur Arbeit mit TM muss zunächst eine Vorstellung des Modells vermittelt werden, welches ein gewisses Maß an Abstraktionsdenken voraussetzt.

(Mit Einschränkung)

12. Fachliche Inhalte können elementar behandelt werden. Spezialkenntnisse und -methoden sind aus dem Problem ableitbar; ihre Kenntnis nicht Voraussetzung zu seiner Bearbeitung.

Ja, es können unter anderem Grammatiken und Zustandsdiagramme behandelt werden.

(Positiv)

13. Die entwickelte Fachsprache wird entwickelt und benutzt. Formale Begriffe werden nicht nur eingeführt, sondern auch angewandt. Die Schüler können die benutzten Begriffe erklären.

Die Voraussetzungen zur Erfüllung der Bedingungen sind gegeben.

(Positiv)

14. Die benutzen Modelle sind als solche erkennbar. Alternative Modellvorstellungen sind möglich. Der Modellbildungsprozess wird deutlich.

Da es sich bei der TM um ein Modell handelt, können die Bedingungen erfüllt werden.

(Positiv)

15. Die bei der Bearbeitung des Themas gewonnen Erfahrungen können verallgemeinert werden. Einblicke in informatikspezifische Themen sind möglich.

Siehe 9 bzw. 12

(Positiv)

16. Das Thema deckt mehrere Ziele des Informatikunterrichts ab, ist multifunktional einzusetzen.

Angeführt werden können unter anderem Hintergrundwissen und Verständnis, Behandlung wichtiger Thematiken oder selbstständiges Arbeiten.

(Positiv)

17. Das Thema steht nicht isoliert, sondern bereitet auch spätere Fragestellungen vor.

Es sind Übergänge zu weiterführenden Themen möglich, beispielsweise der Berechenbarkeit.

(Positiv)

Plagiatserklärung der / des Studierenden

Hiermit versichere ich, dass die vorliegende Arbeit über den „Vergleich bildlich-grafischer Repräsentationen der Turing Maschine“ selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

(Datum, Unterschrift)

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

(Datum, Unterschrift)