Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik

Petrinetze in der Schule:

Ein Spiralcurriculum für die Sekundarstufen I und II

Masterarbeit

zur Erlangung des akademischen Grades
Master of Education (M. Ed.)

Eingereicht von: Michael Rücker
geboren am: 12. August 1988
in: Berlin

Gutachter: Prof. Dr. Niels Pinkwart
Prof. Dr. Wolfgang Reisig

eingereicht am: .....

verteidigt am: .....

Vielen Dank, Katja!

Inhaltsverzeichnis

1 Einleitung **1**

2 Didaktische Vorüberlegungen **5**

2.1 Sachanalyse und Legitimation 5

2.2 Lern- und Curriculumtheorie 11

3 Ein Spiralcurriculum zu Petrinetzen **15**

3.1 Petrinetze im Unterricht 15

3.2 Themenbausteine 18

3.2.1 Eingebettete Systeme 20

3.2.2 Netzwerkkommunikation 27

3.2.3 Nebenläufige Prozesse 37

3.2.4 Formale Sprachen 43

3.2.5 Softwareprojekt 48

4 Erprobung und qualitative Evaluation **53**

4.1 Leitfragen 53

4.2 Theorie und Methodik 54

4.3 Auswertung 63

4.3.1 Petrinetz-Konzepte 63

4.3.2 Modellierte Systeme 67

4.3.3 Versys 72

4.4 Diskussion und Kritik 75

5 Fazit und Ausblick **81**

Literaturverzeichnis **85**

Eigenständigkeitserklärung **88**

Anhang **89**

Abbildungsverzeichnis

1 Elementares Petrinetz 5
2 Nach dem Schalten von t_0 5
3 Farbmischung als gefärbtes Petrinetz 6
4 Oberfläche von *Versys* 16
5 Aufbau des Spiralcurriculums 19
6 Elektronisches Schließsystem 25
7 Licht/Lüfter-System 26
8 Ampelsteuerung 27
9 E-Mail-Kommunikation 30
10 Asymmetrische Verschlüsselung 31
11 BitTorrent-Protokoll mit zwei Peers 32
12 Alternating-Bit-Protokoll als elementares Petrinetz 33
13 Drei speisende Philosophen mit möglicher Verklemmung 40
14 Erzeuger/Verbraucher-System 41
15 Produktionskette 41
16 Petrinetz mit drei sequentiellen Abläufen 45
17 Petrinetz mit nicht-kontextfreier Sprache 46
18 Petrinetz-Sprachen im Verhältnis zur Chomsky-Hierarchie 46

Tabellenverzeichnis

1 Bezug des Themenbausteins *Eingebettete Systeme* zum RLP . . . 22
2 Bezug des Themenbausteins *Netzwerkkommunikation* zum RLP . 29
3 Bezug des Themenbausteins *Nebenläufige Prozesse* zum RLP . . . 38
4 Bezug des Themenbausteins *Formale Sprachen* zum RLP 44
5 Bezug des Themenbausteins *Softwareprojekt* zum RLP 50
6 Deduktive Kategorien zur Kodierung der erhobenen Daten 62

1 Einleitung

Seit ihrer Entstehung kämpft die Schulinformatik um ihre Legitimation als allgemeinbildendes Schulfach. Muss ein Mensch heutzutage programmieren können? Muss man über binäre Rechenwerke, Datenformate oder Netzwerkprotokolle Bescheid wissen, nur um einen internetfähigen Rechner zu benutzen? Muss man die Theorie asymmetrischer Kryptographie verstehen, nur, weil man E-Mails verschlüsseln möchte? Vermutlich nicht. Ein allgemeinbildendes Schulfach muss sich dem Anspruch stellen, Kompetenzen zu fördern, die für jeden Menschen nützlich sind, egal, welche berufliche Laufbahn er einmal einschlagen wird. Dies lässt sich für die oben genannten Punkte nur schwer vertreten.

In seiner Dissertation argumentiert Thomas (2002) stattdessen für den hohen allgemeinbildenden Wert informatischer Modellierung: „Informatische Modelle sind ein Bildungsgut, das für die kulturell-tradierte Tätigkeit des Modellierens von Modellen im Allgemeinen steht“ (S. 82). Dieser Auffassung ist grundlegend zuzustimmen. Die Kognitionspsychologie geht mittlerweile davon aus, dass Menschen ihre Umwelt mithilfe mentaler Modelle verstehen, sodass „Aufgaben und Probleme stellvertretend und mental an diesem inneren Objekt gelöst werden können“ (Moser 2003, S. 185). Sich bewusst mit verschiedenen Modellierungsansätzen auseinanderzusetzen, einen Sachverhalt zu abstrahieren, sich auf das Wesentliche zu beschränken und alles Unwichtige zunächst unberücksichtigt zu lassen sind somit allgemeine kognitive Kompetenzen, die jeder Mensch in verschiedenen Situationen des Lebens immer wieder benötigt.

Thomas (2002) hält zudem fest, dass diese Fähigkeit, die eigene Umwelt mental zu modellieren, „durch Übung gefördert werden“ kann (S. 35). Hinsichtlich der Weiterentwicklung des Schulfaches Informatik argumentiert er daher für eine „Leitlinie ‚Informatische Modellbildung‘, die sich systematisch an der Viel-

falt informatischer Modelle orientiert“ (S. 80). Mittlerweile ist die informatische Modellbildung auch weitgehend als eine zentrale Kompetenz etabliert, die es im Informatikunterricht zu fördern gilt. Sowohl die von der Gesellschaft für Informatik e. V. (GI) (2008) herausgegebenen Bildungsstandards Informatik für die Sekundarstufe I (vgl. S. 45-47) als auch der Berliner Rahmenlehrplan Informatik für die Sekundarstufe II (vgl. LISUM 2006a, S. 9) erkennen dies an.

Die von Thomas (2002) hervorgehobene Vielfalt der Modelltypen scheint dabei jedoch auf der Strecke geblieben zu sein. Die Schulinformatik und auch der fachdidaktische Diskurs werden von statischen Strukturdiagrammen wie Klassendiagrammen und ER-Modellen dominiert. Selbst die Darstellung von Abläufen beschränkt sich in der Regel auf statische Darstellungen wie Pseudo-Codes, Programmablaufpläne oder Struktogramme (vgl. GI 2008, S. 30-34), welche nur geringfügig von einer Programmiersprache abstrahiert sind und sich auch nicht für die Simulation des von ihnen modellierten Verhaltens eignen. Vor dem Hintergrund der oben erwähnten mentalen Modelle muss dies als Defizit angesehen werden. Eine zentrale Funktion mentaler Modelle ist nämlich die Repräsentation von Prozessen, welche anschließend mit ihrer Hilfe „vor dem ‚geistigen Auge‘ simuliert werden“ können (Moser 2003, S. 186).

Zustandsautomaten bieten diese Möglichkeit zur grafischen Simulation von Abläufen zumindest ansatzweise und sind auch etwas weiter abstrahiert. Sie eignen sich jedoch, wie auch die anderen genannten Modelltypen, nicht zur Darstellung verteilter Systeme. In der Schulinformatik scheint diesbezüglich das Dogma zu herrschen, Prozesse, insbesondere informatische Prozesse, wären in jedem Fall sequentiell und terminierend. Eine solche Auffasung kann angesichts eingebetteter und interaktiver Systeme, globaler Rechnernetze und Mehrkernprozessoren heute nicht mehr aufrecht erhalten werden. Betrachtet man noch etwas allgemeinere gesellschaftliche Systeme, wie Verkehrssysteme oder industrielle Produktionspro-

zesse, wird schnell offensichtlich wie absurd die Vorstellung einer sequentiellen Abfolge von globalen Zuständen ist. Die Welt hat keinen globalen Takt. Die Welt ist ein dynamisches, verteiltes System. Ein allgemeinbildender Informatikunterricht, der den Anspruch vertritt, Schülerinnen und Schüler dazu zu befähigen, diese Welt mithilfe von Modellen zu begreifen, kann und darf sich nicht ausschließlich auf sequentielle Prozesse reduzieren.

Ziel dieser Arbeit ist es daher, Petrinetze – eine grafische und simluationsfähige Modellierungstechnik für dynamische, verteilte Systeme – für den schulischen Informatikunterricht didaktisch aufzuarbeiten. Dazu soll ein entsprechendes curriculares Konzept entwickelt, exemplarisch erprobt und qualitativ evaluiert werden.

In Abs. 2 werden zunächst einige didaktische Vorüberlegungen angestellt. Nach einer kurzen Sachanalyse wird anhand verschiedener didaktischer Kriterien argumentiert, dass Petrinetze zunächst ein angemessener Kandidat für den schulischen Informatikunterricht sind. Sie sind keineswegs nur eine Modeerscheinung, sondern seit vielen Jahrzehnten in der Informatik etabliert und finden auch in anderen Gebieten breite Verwendung. Auch die einschlägige fachdidaktische Literatur erwähnt Petrinetze an verschiedenen Stellen – jedoch immer nur sehr oberflächlich. Tiefergehende didaktische Betrachtungen oder gar praktische Unterrichtserfahrungen zu Petrinetzen existieren kaum. Es müssen daher inbesondere die Fragen nach der Vermittelbarkeit von Petrinetzen in den verschiedenen Jahrgangsstufen sowie nach einer sinnvollen curricularen Einflechtung weiter erörtert werden. Diese Arbeit soll einen Beitrag dazu leisten. Eine curriculare Einflechtung kann meiner Meinung nach nur in Form eines Spiralcurriculums erfolgen, sodass Petrinetze in den verschiedenen Jahrgangstufen und in verschiedenen Anwendungskontexten immer wieder thematisiert werden. Eine einmalige Behandlung in Form eines gesonderten Themenblocks würde weder dem Prozesscharakter noch der Zweckbindung informatischer Modellierung gerecht werden.

Ein Curriculum kann sich jedoch nicht nur auf inhaltiche Fragen beschränken. Daher werden Petrinetze in Abs. 3 zunächst aus unterichtsmethodischer Sicht betrachtet. Ich argumentiere, dass eine breite Verwendung von Petrinetzen im Unterricht ein geeignetes Software-Werkzeug zur Modellierung und Simulation voraussetzt. In Ermangelung eines solchen Werkzeuges, das meinen Ansprüchen gerecht wird, habe ich mit *Versys* selbst einen didaktischen Petrinetz-Editor entwickelt. Im Anschluss wird das eigentliche Spiralcurriculum vorgestellt, welches aus insgesamt fünf verschiedenen Themenbausteinen besteht. Für jeden Baustein wird argumentiert, wie Petrinetze sinnvoll darin integriert werden können. Es werden Beispielmodelle und mögliche Lernziele diskutiert. Die Bausteine werden an den jeweiligen Rahmenlehrplänen legitimiert. Sie bauen aufeinander auf und werden von Petrinetzen spiralförmig durchzogen.

Abschließend werden in Abs. 4 zwei der fünf Themenbausteine exemplarisch durch je einen Unterrichtsblock á 90 Minuten erprobt und qualitativ evaluiert. Es wird untersucht, inwiefern die jeweils eingeführten Petrinetz-Konzepte für die Lernenden fasslich sind und ob sich die verwendeten Modelle tatsächlich dazu eignen, die jeweils modellierten Systeme zu veranschaulichen. Auch der von mir geschriebene Editor *Versys* wird evaluiert.

Die Ergebnisse der Evaluation deuten darauf hin, dass Petrinetze bereits in der Sekundarstufe I erfolgreich eingesetzt werden können, um verschiedene informatische Systeme anschaulich zu modellieren und ihr Verhalten grafisch zu simulieren. In der Sekundarstufe II eignen sie sich scheinbar auch gut, um verschiedene theoretische Konzepte, insbesondere im Rahmen nebenläufiger Prozesse, zu analysieren und zu diskutieren. Petrinetze haben somit das Potential, die Vielfalt der Modelle im Informatikunterricht zu bereichern und seinen allgemeinbildenden Wert zu steigern.

2 Didaktische Vorüberlegungen

2.1 Sachanalyse und Legitimation

Ein elementares Petrinetz, die zunächst einfachste Variante von Petrinetzen, wird beschrieben durch ein Tupel $N = (P, T, F, M_0)$, bestehend aus einer Menge P von Plätzen, einer Menge T von Transitionen, sodass $P \cap T = \emptyset$, einer Menge $F \subseteq (P \times T) \cup (T \times P)$ von gerichteten Kanten (auch Flussrelation genannt) und einer Anfangsmarkierung $M_0 : P \to \mathbb{N}$, die jedem Platz eine Anzahl von Marken zuweist. In einer grafischen Darstellung werden Plätze in der Regel durch Kreise, Transitionen durch Rechtecke und Kanten durch Pfeile dargestellt. Die Marken werden als schwarze Punkte direkt in die Plätze gezeichnet. Abbildung 1 zeigt ein einfaches Beispiel.

Wenn eine Transition aktiviert ist, kann sie schalten und damit die aktuelle Markierung des Netzes ändern. Eine Transition t ist aktiviert, wenn sich auf jedem Platz ihres Vorbereichs ${}^\bullet t =_{def} \{p \in P \mid (p,t) \in F\}$ mindestens eine Marke befindet. Schaltet eine Transition t, so verliert jeder Platz ihres Vorbereichs eine Marke und jeder Platz ihres Nachbereichs $t^\bullet =_{def} \{p \in P \mid (t,p) \in F\}$ erhält eine Marke. In Abb. 1 ist die Transition $\mathsf{t_0}$ aktiviert, die Transition $\mathsf{t_1}$ nicht. Abbildung 2 zeigt das Netz nach dem Schalten von $\mathsf{t_0}$. Hier sind nun beide Transitionen aktiviert. Welche als nächstes schaltet, ist jedoch nicht festgelegt: das Netz ist nichtdeterministisch. Sollte $\mathsf{t_1}$ schalten, so verschwindet auch die

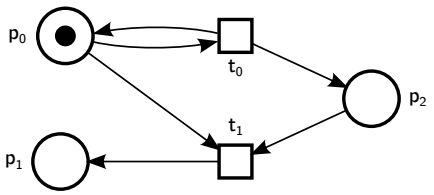


Abb. 1: Ein elementares Petrinetz

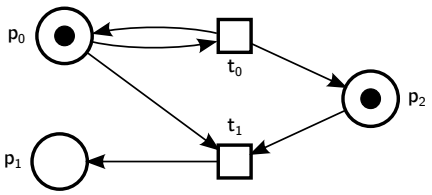
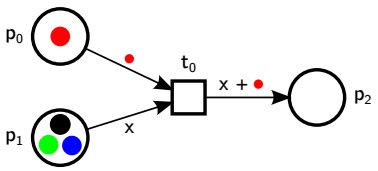


Abb. 2: Nach dem Schalten von $\mathsf{t_0}$

Marke von p_0 und keine Transition wäre mehr aktiviert. Die beiden Transitionen stehen hier also im Konflikt.

Diese elementare Variante von Petrinetzen kann nun durch verschiedene zusätzliche Ausdrucksmittel erweitert werden. Dazu zählen z.B. Kantengewichte, die es erlauben, dass beim Schalten einer Transition mehre Marken „durch eine Kante fließen;“ Platzkapazitäten, d.h. Beschränkungen für die maximale Markenzahl eines Platzes; oder Inhibitor-Kanten, die dazu führen, dass eine Transition aktiviert ist, wenn gerade keine Marken auf einem ihrer Vorplätze liegen. Andere Erweiterungen, die typischerweise die Schaltregel betreffen, führen z.B. zu zeitbewerteten oder stochastischen Netzen, die es erlauben, zu analysieren wann, wie häufig bzw. mit welcher Wahrscheinlichkeit eine bestimmte Transition schaltet. Eine sehr häufig verwendete Erweiterung sind die sogenannten gefärbten Netze, welche im Folgenden kurz vorgestellt werden sollen, da sie für den weiteren Verlauf dieser Arbeit relevant sind.

In einem gefärbten Petrinetz können die Marken unterschiedliche Farben haben. Das heißt, eine Markierung ist nicht länger eine Abbildung in die natürlichen Zahlen, sondern eine Abbildung $M : P \to \mathcal{M}(\Sigma)$ in die Menge aller Multimengen über einem beliebigen Alphabet von Symbolen, den „Farben.“ Intuitiv ist eine Multimenge schlicht eine Menge, in der gleiche Elemente mehrfach vorkommen können. Auf einem Platz könnten also z.B. drei rote und eine grüne Marke liegen. Solche Markierungen erfordern natürlich entsprechende Kantengewichte, die angeben, welche Farben dazu führen, dass eine bestimmte Transition akti-



Sei Σ die Menge aller Farben und $+ : \Sigma \times \Sigma \to \Sigma$ die additive Farbmischung.

Abb. 3: Farbmischung als gefärbtes Petrinetz

viert ist. Die Gewichtungen sind nun auch nicht mehr auf einfache Konstanten beschränkt, sondern können auch Variablen oder ganze Terme enthalten. Abbildung 3 zeigt ein gefärbtes Petrinetz, in dem durch das Schalten der Transition t_0 die rote Farbe in p_0 mit einer Farbe aus p_1 „gemischt“ wird. Welche Farbe aus p_1 dabei genommen wird, d.h. in welchem Modus die Transition eintritt, ist wiederum nichtdeterministisch.

Eine gut lesbare Einführung in elementare und gefärbte (dort: „allgemeine“) Petrinetze findet sich bei Reisig (2010). Eine ausführlichere Betrachtung gefärbter Netze findet sich bei Jensen und Kristensen (2009). Bause und Kritzinger (2002) betrachten stochastische Netze. Einen ausführlichen Überblick über unterschiedliche Netzklassen sowie verschiedene Hybridformen liefern David und Alla (2005).

Petrinetze eignen sich hervorragend zur grafischen Modellierung und Simulation von vernetzten und verteilten dynamischen Systemen. Doch sind sie auch für den schulischen Informatikunterricht geeignet? Hubwieser (2007) beschreibt vier grundlegende didaktische Kriterien, für die Auswahl informatischer Lerninhalte: allgemeine Bedeutung, Lebensdauer, Vermittelbarkeit sowie exemplarische Auswahl und Einflechtung (vgl. S. 83-85). Diese sollen im Folgenden kurz für Petrinetze diskutiert werden.

Hubwieser (ebd.) zufolge hat ein Lerngegenstand dann eine allgemeine Bedeutung, wenn er „einen möglichst breiten Anwendungsbereich“ hat (S. 83). Er unterscheidet dazu vier Allgemeinheitsklassen, wobei in die allgemeinste Klasse Gegenstände fallen, für die „eine Anwendung auch außerhalb des Bereichs der EDV möglich“ ist (S. 83). Abgesehen davon, dass Hubwieser (ebd.) an dieser Stelle bereits „Modellierungstechniken“ als Beispiel nennt (S. 83), trifft dieses Kriterium auf Petrinetze uneingeschränkt zu. Viele der oben genannten Erweiterungen und Spezifizierungen für Petrinetze wurden vorrangig aus ihren vielen unterschiedlichen Anwendungsgebieten heraus motivert und entwickelt. Dazu zählen nicht

nur die Software-Entwicklung, sondern auch Gebiete außerhalb der Informatik wie die Systembiologie (vgl. Koch et al. 2011) oder die Soziologie (vgl. Köhler et al. 2001). Carl Adam Petri, der Namenspatron der Petrinetze, wurde bei seinen ursprünglichen Überlegungen maßgeblich durch die Physik inspiriert (vgl. Brauer und Reisig 2009, S. 132-133) und auch dort finden Petrinetze Verwendung (vgl. Abramsky 2008).

Ein ähnlich positives Fazit kann für das Kriterium der Lebensdauer gezogen werden. Hubwieser (2007) fordert, dass jedem Lerngegenstand „auch dann noch eine ausreichende Bedeutung [zukommt], wenn die Schüler ins Berufsleben eintreten“ (S. 84). Ob sich Petrinetze auch in zehn Jahren noch einer derart breiten Anwendung erfreuen wie heute, lässt sich natürlich nicht mit absoluter Sicherheit sagen. Sie haben jedoch, seit ihrer Entstehung in den 1960er Jahren, eine erstaunliche Erfolgsgeschichte hinter sich gebracht:

> Petrinetze gehören zu den ältesten Systemmodellen der Informatik. [...] Petrinetze sind nie, wie einige andere Methoden, eine Weile bevorzugt und dann vergessen worden, sondern haben über die Jahrzehnte ihren Platz in der Methodenvielfalt behalten. (Reisig 2010, S. 1)

Vernetzte und verteilte Systeme gewinnen in unserer heutigen Gesellschaft zudem immer mehr an Relevanz (Stichwort: Cloud-Computing, *ubiquitous computing* oder „Internet der Dinge“), sodass durchaus vermutet werden kann, dass Petrinetze, als eine zentrale Modellierungstechnik für derartige Systeme, auch noch in Zukunft eine breite Verwendung finden werden.

„Der interessanteste Lernstoff hat in der Schule natürlich nichts verloren, wenn eine altersgemäße Vermittlung nicht möglich ist“ (Hubwieser 2007, S. 84). Diese Forderung ist einerseits offensichtlich, andererseits sehr komplex. Eine unterrichtliche Vermittlung eines Lernstoffs erfordert in der Regel zunächst eine „Reduktion der inhaltlichen Komplexität“ auf ein für die Lernenden fassliches

Maß (Lehner 2012, S. 118). Welches Maß für die Lernenden noch fasslich ist, ist natürlich je nach Altersgruppe verschieden und kann letztlich nur im Rahmen konkreter unterrichtlicher Umsetzungen validiert werden. Die Frage nach der Vermittelbarkeit eines Lerngegenstandes muss somit immer auch praktische und methodische Überlegungen umfassen (vgl. Meyer 2009, S. 71-108).

Nach meinem Wissen wurden diese Fragen für die Altersgruppen der Sekundarstufen I und II für Petrinetze bisher gar nicht bzw. kaum erörtert. Petrinetze werden zwar sowohl von Schubert und Schwill (2011, S. 265-274) als auch von Hubwieser (2007, S. 248-249) in ihren jeweiligen Monographien zur „Didaktik der Informatik“ erwähnt, die Betrachtungen gehen jedoch kaum über eine grundlegende Einführung und das Fazit, Petrinetze würden sich gut zur Darstellung nebenläufiger Prozesse eignen, hinaus. Hubwieser et al. (2010) erwähnen Petrinetze sogar in ihrem Informatik-Lehrwerk für Gymnasien, doch auch dort werden nur ein kurzer historischer Abriss und zwei Einführungsbeispiele gegeben (vgl. S. 86-87). Konkrete Aufgabenstellungen, ein Hauptaspekt schulischer Lehrwerke, fehlen. Auch die Ausführungen von Wikarski (1997a; 1997b; 1997c) zu kommunizierenden Netzen, Geschäftsprozessen und Analyse-Methoden, die immerhin in einer fachdidaktischen Zeitschrift publiziert wurden, bleiben rein fachwissenschaftlich. Der einzige mir bekannte Praxisbericht zu Petrinetzen in der Schule stammt von Rotering (2010), welcher sie in einer elften Klasse einsetzte um zunächst nebenläufige Prozesse und im Anschluss Software für Steuerungssysteme zu modellieren. Die von ihm eingesetzten elementaren Petrinetze waren für seine Schülerinnen und Schüler dabei scheinbar „leicht [zu] erlernen“ (S. 30). Für die Sekundarstufe I sind mir keinerlei Berichte zur praktischen Umsetzung von Petrinetzen im Unterricht bekannt.

Es muss daher festgehalten werden, dass sich die Frage nach der Vermittelbarkeit von Petrinetzen in den Sekundarstufen I und II an dieser Stelle nicht

befriedigend klären lässt. Die meisten didaktischen Betrachtungen bleiben schlicht zu fachwissenschaftlich, um für die Unterrichtspraxis hilfreich zu sein. Der Unterrichtsversuch von Rotering (2010) kann bestenfalls Anhaltspunkte für die Sekundarstufe II liefern, lässt jedoch keine Aussagen über jüngere Altersgruppen zu. Die Frage der Vermittelbarkeit zu klären, ist jedoch entscheidend, wenn Petrinetze Teil des Informatikunterrichts werden sollen. Es ist daher ein zentrales Ziel dieser Arbeit, einen Beitrag zur Klärung dieser Frage zu leisten.

Das letzte von Hubwieser (2007) genannte Kriterium ist das der exemplarischen Auswahl und Einflechtung. Das exemplarische Prinzip meint dabei die Behandlung einer ganzen „Klasse von Konzepten“ anhand eines „einzelne[n] repräsentative[n] Vertreter[s]“ (S. 85). In diesem Kontext können zunächst Petrinetze selbst als ein exemplarischer Vertreter einer grafischen Modellierungstechnik für dynamische und verteilte Systeme gesehen werden (vgl. Thomas 2002, S. 68). Wie bereits erwähnt, ist eine solche Modellklasse meines Wissens nach im Informatikunterricht bisher kaum oder gar nicht vertreten, sodass eine Betrachtung von Petrinetzen hier tatsächlich einen echten Mehrwert schaffen kann. Die Erarbeitung konkreter, für den Unterricht geeigneter Petrinetz-Modelle, anhand derer wiederum verschiedene informatische Systeme exemplarisch betrachtet werden können, ist ein zentrales Anliegen dieser Arbeit und wird in Abs. 3 ausführlicher erörtert. Für das Prinzip der Einflechtung hält Hubwieser (2007) fest, dass es für einige Lerngegestände angemessener wäre, „sie immer wieder an verschiedenen Stellen anhand des gerade betrachteten Systems anzusprechen, anstatt sie allein in den Mittelpunkt des Unterrichts zu stellen“ (S. 85). Dieses Prinzip ergänzt sich hervorragend mit dem des Spiralcurriculums. Warum solch ein spiralcurricularer Ansatz meiner Meinung nach für Petrinetze unbedingt verfolgt werden sollte, wird im folgenden Abschnitt diskutiert.

2.2 Lern- und Curriculumtheorie

Lehner (2009) unterscheidet zwei grundlegende Formen von Curricula: das gestufte Curriculum und das spiralförmige Curriculum. Ein gestuftes Curriculum besteht aus verschiedenen „in sich abgeschlossene[n] Themen [...], wobei die thematische Komplexität [mit jedem neuen Thema] zunimmt“ (S. 124). Jedes Thema taucht also nur ein einziges Mal im Curriculum auf, wobei es an dieser Stelle möglichst vollständig und umfassend behandelt wird. Danach folgt ein neues, anspruchsvolleres Thema. Für Petrinetze würde dies bedeuten, dass sie einmalig im Rahmen eines gesonderten Themenblocks möglichst umfangreich behandelt und danach nie wieder eine Rolle spielen würden.

Daraus würden sich jedoch verschiedene Probleme ergeben. Zunächst müsste für solch einen gesonderten Block natürlich geklärt werden, wo genau er im Curriculum verortet werden soll. Dies allein dürfte keine einfache Aufgabe sein. In jedem Fall entsteht aber unmittelbar ein Konflikt zwischen dem neuen Thema und dem alten, welches ursprünglich an dieser Stelle im Curriculum behandelt werden sollte. Entscheidet man sich, das neue Thema einfach zusätzlich noch zu behandeln, erhöht sich der inhaltliche Gesamtumfang des Curriculums und es entsteht ein Stoffmengenproblem (vgl. Lehner 2012, S. 72-80). Entscheidet man sich alternativ dazu, einen (oder Teile eines) bisherigen Themenblock(s) für den neuen zu opfern, entsteht ein Kanonisierungsproblem, denn es müsste begründet werden, warum das neue Thema besser oder wertvoller ist als das bisherige. Dies ist jedoch selten objektiv und pauschal möglich.

Ein gesonderter Themenblock zu Petrinetzen würde aber vor allem auch den von der GI (2008) herausgegebenen Bildungsstandards widersprechen, in denen die informatische Modellierung klar als prozessbezogene Kompetenz ausgewiesen ist, also als eine „Art, wie mit den Inhalten umgegangen wird“ (S. 45). Dies impli-

ziert, dass Modellierungstechniken nicht um ihrer selbst Willen, sondern immer nur bezogen auf die jeweils betrachteten Systeme unterrichtet werden sollen. Ein gesonderter Themenblock, der Petrinetze selbst in den Mittelpunkt des Unterrichts stellt, würde dem offensichtlich widersprechen.

Ein Spiralcurriculum verfolgt einen grundlegend anderen Ansatz:

> Ein Thema wird über einen größeren Zeitraum so verteilt, dass die Komplexität des Themas in den aufeinander folgenden Abschnitten zunimmt. Bestimmte thematische Begriffe, Strukturen und Konzepte werden wiederholt aufgegriffen und weiterentwickelt. (Lehner 2009, S. 123-124)

Das Konzept des Spiralcurriculums geht auf den Entwicklungspsychologen Jerome Bruner (1980) zurück und basiert im Kern auf der Annahme, dass „jedem Kind in jedem Stadium seiner Entwicklung jeder Lehrgegenstand in einer intellektuell ehrlichen Form beigebracht werden kann“ (S. 61). Er begründet diese Annahme vorwiegend kognitivistisch mit der „Denkweise des heranwachsenden Kindes“ (S. 61) und dem Einordnen von neuen Informationen in vorhandene Wissensstrukturen (vgl. S. 60). Das Konzept des Vorwissens hat also eine zentrale Bedeutung für das Spiralcurriculum. Jede erneute Iteration des Themas baut auf den vorherigen auf. Vorhandene Kenntnisse werden ausgebaut, vertieft und in neue Kontexte eingebettet.

Erst solch ein spiraliges Vorgehen erlaubt es, Petrinetze bzw. Modellierungstechniken im Allgemeinen im Sinne einer prozessbezogenen Kompetenz im Kontext verschiedener Inhaltsbereiche wiederholt zu thematisieren und in verschiedenen Anwendungssituationen weiterzuentwickeln. Nur durch solche konkreten Anwendungssituationen kann den Lernenden wiederum die „Zweckbindung von Modellen“ (GI 2008, S. 45) nahe gebracht werden. Modelle existieren nicht um ihrer selbst Willen, sondern weil mit ihrer Hilfe konkrete Probleme gelöst werden sollen. Es können somit auch konstruktivistische Argumente für einen spiralcur-

ricularen Ansatz für Petrinetze gefunden werden: die Lernenden sollen informatische Modelle möglichst handlungsorientiert einsetzen, d.h. sie im Sinne eines Handlungsproduktes selbst *herstellen* und anschließend auch *verwenden* um ein konkretes Problem zu lösen (vgl. Gudjons 2008, S. 124-126).

Natürlich wäre es naiv, zu glauben, Petrinetze könnten im Unterricht ausschließlich im Sinne einer zweckgebundenen Modellierung eingesetzt werden, ohne dass sie zuvor selbst einmal als Lerngegenstand im Mittelpunkt standen. Die bereits erwähnten Probleme der Stoffmenge und der Kanonisierung können durch einen spiralcurricularen Ansatz daher nicht vollständig beseitigt werden. Es ist jedoch davon auszugehen, dass sie bei Weitem nicht so stark auftreten. Lediglich das Einführen neuer Modellierungskonzepte verlangt eine explizite Thematisierung. In einer anschließenden zweckgebundenen und handlungsorientierten Anwendung stellen Petrinetze lediglich Teilkompetenzen dar, die dem eigentlichen Handlungsziel untergeordnet sind und quasi „nebenbei“ geübt werden (vgl. ebd., S. 48-51). Sie stehen also, vor dem Hintergrund eines Kanonisierungsproblems, nicht im Konflikt mit dem anderen Thema, sondern nehmen vielmehr eine unterstützende Funktion ein, indem sie die Lerninhalte anschaulich modellieren. Es kann daher sogar vermutet werden, dass sich an einigen Stellen Synergieeffekte ergeben, d.h. die Lernenden die behandelten Themen mithilfe von Petrinetzen sogar besser und effizienter verstehen und bearbeiten können als mit anderen Methoden. Diese Vermutung ist einer der Hauptgründe, aus denen ich Petrinetze für den Informatikunterricht vorschlage.

Abschließend muss an dieser Stelle noch geklärt werden, nach welcher Methodik ein solches Spiralcurriculum konstruiert werden soll. Wie in Abs. 2.1 erwähnt, erfordert die Vermittelbarkeit eines Lernstoffes zunächst eine Reduktion seiner Komplexität. Auch die Forderung von Bruner (1980) nach einer für das Kind „intellektuell ehrlichen Form“ (S. 61) bedeutet in der Konsequenz nichts anderes,

als dass der entsprechende Gegenstand didaktisch reduziert werden muss. Eine solche Reduktion „von oben herab“ birgt jedoch die Gefahr einer Abbild-Didaktik, weshalb Jank und Meyer (2002) sie kategorisch ablehnen:

> Abbild-Didaktiken arbeiten mit der irrigen Annahme, die bildenden Strukturen seien schon in der Fachwissenschaft eingeschlossen. Sie müssten nur von Fachleuten hereausgefiltert werden. [...] Eine etwas anspruchsvollere, vor fünfzig Jahren vertretene Position lautete, die Unterrichtsinhalte würden durch eine „didaktische Reduktion“ aus den Fachwissenschaftstrukturen herausgefiltert. Auch diese Position ist überholt. Sie betont zwar den eigenständigen didaktischen Gedankengang, aber sie ist einseitig wissenschafts- und inhaltsorientiert. Sie blendet – ebenso wie die Abbild-Didaktik – die Methodenfrage und damit der Lernerperspektive aus. (S. 32)

Dem gegenüber stellt Lehner (2012) die Forderung nach „einer Ergänzung der derzeit vorherrschenden Methoden-, Medien- und Lernergebnisorientierung durch eine stärker inhaltliche Orientierung, die dann auch Fragen der didaktischen Reduktion einschließen würde“ (S. 31).

Es lässt sich nicht abstreiten, dass das Thema dieser Arbeit maßgeblich durch den Inhalt, Petrinetze, bestimmt wird. Um der Gefahr einer zu starken Orientierung an der Fachwissenschaft zu begegnen, soll diese deshalb nicht als Ausgangspunkt einer didaktischen Reduktion dienen. Stattdessen werden, im Sinne eines *Bottom-Up*-Ansatzes, verschiedene etablierte und auch methodisch gut untersuchte Themengebiete des Informatikunterrichts betrachtet und anschließend diskutiert, wie Petrinetze sinnvoll in das jeweilige Thema integriert werden können. Dies entspricht exakt der von Hubwieser (2007) beschriebenen Einflechtung (vgl. S. 85). Die daraus hervorgehende, reduzierte Inhaltsstruktur zu Petrinetzen wird somit nicht durch die Fachwissenschaft, sondern durch konkrete unterrichtliche Anwendungen legitimiert. Die verschiedenen Themen bilden schließlich die „Bausteine“ des Spiralcurriclums, d.h. Petrinetze setzen sich spiralig durch die Themen fort (vgl. Brühne 2009, S. 75).

3 Ein Spiralcurriculum zu Petrinetzen

In diesem Abschnitt soll nun nach der in Abs. 2.2 beschriebenen Vorghensweise eine mögliche spiralcurriculare Einflechtung zu Petrinetzen vorgestellt werden. Wie die Betrachtungen in Abs. 2 gezeigt haben, können sich die Fragen nach der Vermittelbarkeit und der curricularen Einflechtung eines Themas nicht nur auf inhaltliche Aspekte beschränken. Daher sollen Petrinetze zunächst kurz aus unterrichtsmethodischer Sicht betrachtet werden.

3.1 Petrinetze im Unterricht

Petrinetze lassen sich am einfachsten grafisch darstellen (vgl. Abs. 2.1). Das bedeutet, sie können im Unterricht ohne Weiteres an die Tafel oder auf Papier gezeichnet werden. Auch das dynamische Verhalten eines Petrinetzes kann so bereits simuliert werden, indem z.B. Magnete, Reißzwecken oder Spielsteine als Marken verwendet und über die Tafel bzw. das Papier geschoben werden (vgl. Schwill 1993, S. 51).

Wie bereits erwähnt, sollen die Modelle im Unterricht möglichst handlungsorientiert eingesetzt, d.h. von den Lernenden selbstständig und vor allem zweckorientiert erstellt werden. Die verschiedenen Modellierungsansätze können dabei durchaus komplexer werden. Vor allem muss aber beachtet werden, dass die Lernenden bei der Modellierung Fehler machen werden, sogar machen sollen (vgl. Gudjons 2008, S. 57), welche sie im Anschluss möglichst eigenständig finden und korrigieren sollen. Dies erfordert es wiederum, die Modelle kontinuierlich, eventuell sogar experimentell, zu verändern und anzupassen. Es ist offensichtlich, dass die oben beschriebene „analoge“ Methode dabei sehr schnell an ihre Grenzen stößt. Hinzu kommt, dass ein Arbeiten mit Stift und Papier im Informatikunterricht von den Lernenden häufig als trocken und theoretisch angesehen wird.

Eine breitere Verwendung von Petrinetzen im Unterricht erfordert daher ein Software-Werkzeug, das es erlaubt, sie auch am Rechner zu modellieren und zu simulieren. Es existiert auch eine nahezu unüberschaubare Vielfalt an Modellierungs- and Analysewerkzeugen für Petrinetze. Die von der TGI-Gruppe der Universität Hamburg (2013) verwaltete *Petri Nets Tool Database* verschafft einen Überblick. Die dort aufgeführten Werkzeuge sind jedoch vorrangig für den professionellen oder wissenschaftlichen Gebrauch entwickelt. Viele sind proprietär, laufen nur auf bestimmten Plattformen oder besitzen nicht einmal eine grafische Oberfläche und sind somit für den schulischen Einsatz ungeeignet. Rotering (2010) kommt zu einem sehr ähnlichen Schluss und hält schließlich „die leider nur in englischer Sprache verfügbare Software *Pipe2*“ für die beste Alternative, mit der seine Lerngruppe in der Sekundarstufe II scheinbar auch gut zurecht kam (S. 11). In jüngeren Jahrgangsstufen könnte jedoch nicht nur die englische Oberfläche von *Pipe2* Probleme bereiten. Viele Funktionen des Programms sind meiner Meinung nach für einen schulischen Kontext schlicht überflüssig und somit im Zweifel für die Lernenden eher verwirrend und ablenkend. Mir ist aber leider auch kein anderer Editor bekannt, der besser für die Schule geeignet wäre.

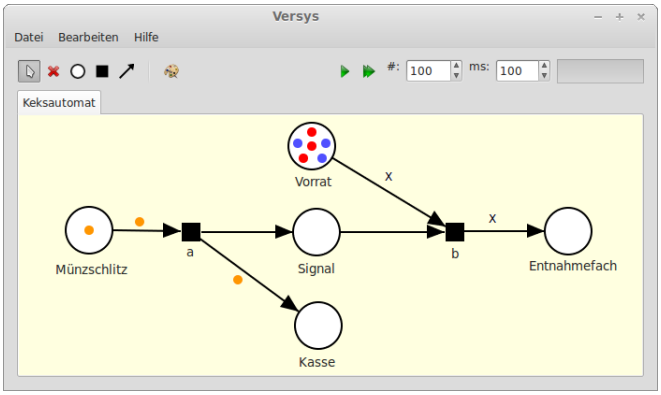


Abb. 4: Die Oberfläche von *Versys* mit einer Variante des Keksautomaten von Reisig (2010, S. 11)

Dies hat mich veranlasst, selbst einen Petrinetz-Editor zu entwickeln, der speziell für den schulischen Einsatz konzipiert ist. *Versys* ist in Java geschrieben, quelloffen und portabel. Das Programm, die Quelltexte und eine vollständige Dokumentation stehen unter einer Creative Commons Lizenz (CC BY-SA 4.0) und können unter `http://www.informatik.hu-berlin.de/~ruecker/` heruntergeladen werden.

Der Funktionsumfang von *Versys* ist didaktisch motiviert. Er beschränkt sich auf genau die Konzepte, die im folgenden Abschnitt in den verschiedenen Themenbausteinen verwendet werden. Alle dort vorgestellten Petrinetz-Modelle stehen als *Versys*-Dateien zur Verfügung und können ebenfalls unter der obigen Adresse heruntergeladen werden.

Die Oberfläche von *Versys* ist möglichst reduziert. Sie ist in Abb. 4 dargestellt. Die Bedienung soll so intuitiv und selbsterklärend wie möglich sein. Zeit, die im Unterricht damit verbracht wird, die verwendete Lernsoftware zu erklären, ist meiner Meinung nach schlicht vergeudet. Inwiefern *Versys* diesem Anspruch gerecht wird, wird in Abs. 4 genauer untersucht.

Das Programm bietet grundlegend drei verschiedene Arbeitsmodi, zwischen denen jederzeit frei gewechselt werden kann: Modellierung, manuelle und automatische Simulation. Während einer manuellen Simulation können die Lernenden die aktivierten Transitionen eines Netzes durch Anklicken selbst schalten lassen. In einer automatischen Simulation übernimmt dies das Programm, in einer vorher festgelegten Geschwindigkeit und Anzahl.

Dieser freie Wechsel zwischen Modellierung und Simulation erfordert, dass sich stets ein zumindest syntaktisch korrektes Netz auf der Arbeitsfläche befindet. *Versys* erlaubt daher währen der Modellierung keine syntaktisch falschen Operationen, zum Beispiel das Erstellen einer Kante zwischen zwei gleichen Elementen. Die Überlegungen ähneln an dieser Stelle denen zu didaktischen Programmier-

umgebungen. Auch dort herrscht zunehmend ein Konsens, dass die Syntax bei der Programmierung eine eher untergeordnete Rolle spielen sollte, sodass sich die Lernenden weniger auf die Programmiersprache selbst und stärker auf die darunter liegenden algorithmischen Konzepte konzentrieren können (vgl. z.B. Strecker 2009, S. 46; Resnick et al. 2009, S. 63; Powers et al. 2007, S. 214). Analog ließe sich für Petrinetze argumentieren, dass die Lernenden sich in einem handlungs- und zweckorientierten Unterricht in erster Linie auf das zu lösende Problem, d.h. das zu modellierende System, konzentrieren sollen und nicht auf die Syntax der verwendeten Modellierungssprache. *Versys* vertritt den Anspruch, eine solche handlungs- und zweckorientierte Anwendung von Petrinetzen zu unterstützen. Auch dieser Aspekt wird in Abs. 4 einer Evaluation unterzogen.

3.2 Themenbausteine

Wie in Abs. 2.2 dargestellt, ist das Ziel des hier vorgestellten Curriculums *nicht*, Petrinetze als gesonderten Themenblock zu behandeln. Vielmehr sollen sie in vorhandene und etablierte Themenfelder des Informatikunterrichts integriert werden. Zu allen vorgestellten Themen existieren bereits verschiedene didaktische Betrachtungen, Unterrichtsmaterial oder sogar vollständig geplante und erprobte Sequenzen. Alle Themen lassen sich zumindest an den Berliner Rahmenlehrplänen legitimieren, was jeweils auch getan wird. Ziel dieses Abschnittes ist es, aufzuzeigen, dass die Integration von Petrinetzen in die folgenden Themen prinzipiell möglich und vor allem sinnvoll wäre, da sie an vielen Stellen eine veranschaulichende Funktion erfüllen oder sogar eine völlig neue Perspektive auf den jeweiligen Unterrichtsgegenstand eröffnen können.

Abbildung 5 zeigt den Aufbau des Curriculums. Auf der linken Seite sind die fünf Themenbausteine dargestellt, aus denen das Curriculum besteht. Der Pfeil auf der rechten Seite beinhaltet die entsprechenden Petrinetz-Konzepte, die

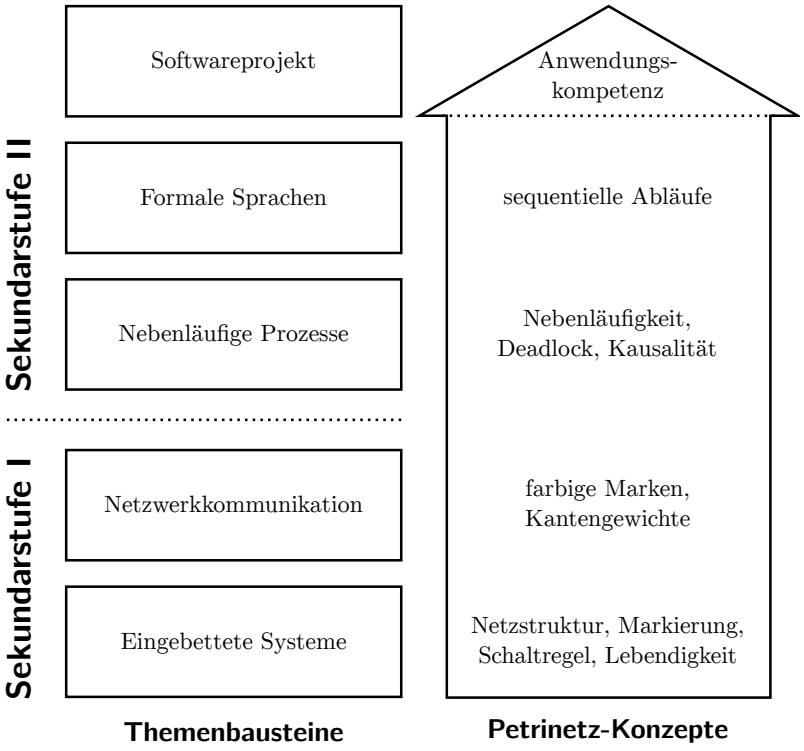


Abb. 5: Aufbau des Spiralcurriculums

für die Modellierung und Analyse der jeweils betrachteten Systeme verwendet werden. Im Rahmen der Sekundarstufe I sollen in den ersten beiden Themenbausteinen zu eingebetteten Systemen und zur Netzwerkkommunikation vor allem die grundlegenden Modellierungskonzepte für elementare und gefärbte Netze erarbeitet werden. Die Lernenden sollen mit den neuen Modellierungstechniken vertraut gemacht werden und erfahren, in welchen Kontexten Petrinetze sinnvoll zur Modellierung eingesetzt werden können. In der Sekundarstufe II liegt der Fokus dagegen stärker auf der Analyse theoretischer Konzepte wie Nebenläufigkeit und formale Sprachen. Die Lernenden sollen nun auch stärker über Petrinetze reflektieren und auch Kontexte kennenlernen, in denen sie sich gerade nicht besonders gut zur Modellierung eignen. Im Rahmen des abschließenden Softwareprojekts sollen die Lernenden schließlich selbst entscheiden, ob und zu welchem Zweck sie

Petrinetze als Modellierungstechnik einsetzen. Sie sind nun lediglich eine weitere neben verschiedenen anderen Modellierungstechniken, die ihnen im Unterricht zur Verfügung stehen.

Die einzelnen Themenbausteine werden im Folgenden detaillierter vorgestellt. Nach einer kurzen Sachanalyse zum jeweiligen Thema folgt eine didaktische Betrachtung, mit dem Ziel, Petrinetze sinnvoll in das Thema zu integrieren. Es werden methodische Ansätze und vor allem Beispiele für Petrinetz-Modelle genannt, die in einer entsprechenden Unterrichtseinheit behandelt werden könnten. Die Netze werden dabei in erster Linie didaktisch-methodisch diskutiert. Ihre Funktionsweise wird nicht jedes Mal im Detail erläutert. Zum fachlichen Verständnis reichen die in Abs. 2.1 vorgestellten Konzepte. Alle Netze stehen zudem, wie bereits erwähnt, auch als *Versys*-Dateien zur Verfügung. Die Leserin oder der Leser ist somit an dieser Stelle eingeladen, die Modelle mithilfe des Programms selbst auszuprobieren und ihre Funktionsweise interaktiv nachzuvollziehen.

3.2.1 Eingebettete Systeme

Klassische Desktop-Rechner sind heutzutage die Ausnahme. Die große Mehrheit aller Mikroprozessoren wird mittlerweile für sogenannte eingebettete Systeme verwandt:

> Ein eingebettetes System [...] ist eine Software-/Hardware-Einheit, die über Sensoren und Aktuatoren mit einem Gesamtsystem verbunden ist und darin Überwachungs-, Steuerungs- beziehungsweise Regelungsaufgaben übernimmt. In der Regel handelt es sich bei eingebetteten Systeme [sic] um reaktive, häufig auch um hybride verteilte Systeme mit Echtzeitanforderungen. (Broy zitiert nach Marwedel 2000, S. 16)

Eingebettete bzw. reaktive Systeme stehen somit ständig in Interaktion mit ihrer Umwelt, die sie über ihre Sensoren wahrnehmen und in die sie über ihre Aktuatoren zurückwirken. Typische Beispiele wären elektronische Schließsysteme,

Aufzugs- und Ampelsteuerungen oder Alarmanlagen.

Inbesondere von einem algorithmischen Standpunkt aus betrachtet, „unterscheidet sich [ein eingebettetes System] grundsätzlich von einem Rechner, dessen Programm eine Funktion berechnet“ (Marwedel 2000, S. 16). Es stellt vielmehr eine Art Service-System dar, dessen Dienst den Nutzern zur Laufzeit zur Verfügung steht. Die Software, die auf einem solchen System läuft, lässt sich daher eher als Betriebssystem verstehen, das in einer Endlosschleife läuft: „input is continually arriving; output is continually being produced“ (Stein 2006, S. 470). Eine solche Art der kontinuierlichen Interaktion mit der Umwelt lässt sich nicht mit den klassischen, sequentiellen algorithmischen Modellen wie dem EVA-Prinzip oder dem Begriff der Terminierung beschreiben:

> In a conventional computation, the end is the moment of success. When an interactive system – an operating system, a network, a robot – stops, it is usually because something has gone wrong. (ebd., S. 470)

Damit die Lernenden dieses Prinzip in einer entsprechenden Unterrichtseinheit möglichst selbst erfahren können, sollten sie meiner Meinung nach im Rahmen eines Projektes ein eigenes eingebettetes System konstruieren, konfigurieren und programmieren. Eine so gestaltete Einheit eignet sich, um verschiedene Aspekte der im Rahmenlehrplan genannten thematischen Module zu behandeln. Tabelle 1 zeigt die für diesen Themenbaustein relevanten Teile des Berliner Rahmenlehrplans der Sekundarstufe I.

Mit dem Lego-Mindstorms-System, dem PicoBoard und dem Arduino existieren auch bereits verschiedene Plattformen, die sich sehr gut für einen handlungsorientierten Umgang mit eingebetten Systemen im Informatikunterricht eignen. Diese Umgebungen sollen hier nicht weiter im Detail betrachtet werden, da für sie bereits verschiedene didaktische Analysen und Unterrichtsentwürfe für die Sekundarstufe I publiziert worden sind. Diese sind in der Regel sehr handlungsorientiert;

	Die Schülerinnen und Schüler:	**Inhalte und mögliche Kontexte:**
Aufbau und Wirkungsweise von Informatiksystemen (S. 25-26)	• stellen Algorithmen grafisch dar • erarbeiten ein Softwareprodukt nach der Grundstruktur informatischer Projekte • entwickeln selbstständig algorithmische Lösungen	• Betriebssysteme
Automatische und technische Systeme (S. 29)	• dokumentieren Einsatzzweck und Arbeitsweise eines automatischen Systems • programmieren und testen ein reduziertes Modell	• Mögliche Untersuchungsgegenstände: Mobiltelefone, Haushaltsgeräte, Steuerung von Haus- und Klimatechnik, Fahrgast-Informationssysteme im öffentlichen Personennahverkehr, Personenkraftwagen

Tabelle 1: Bezug des Themenbausteins *Eingebettete Systeme* zum Berliner Rahmenlehrplan der Sekundarstufe I (LISUM 2006b)

die dabei entwickelten Systeme sind fast immer interaktiv oder eingebettet (vgl. z.B. Modrow und Strecker 2011; Baumann 2011; Hirsch et al. 2000).

Das Thema *eingebettete Systeme* für den Informatikunterricht der Sekundarstufe I ist also keinesfalls neu und unerforscht. Bei einigen der didaktischen Betrachtungen wird jedoch deutlich, dass die Autoren immer wieder versuchen, die Funktionsweise interaktiver Systeme mit dem EVA-Prinzip zu erklären. Dieses ist grundlegend geprägt von der „klassischen“ theoretischen Informatik und den

Fragen der Berechen- und Entscheidbarkeit (vgl. Hubwieser et al. 2013, S. 15). So hält Baumann (2011) zunächst fest: „Der Arduino funktioniert nach dem EVA-Prinzip“ (S. 53), kommt jedoch später zu der widersprüchlichen Erkenntnis, dass sich die „Endlosschleife des Arduino-Programms [...] nicht unter den bisherigen Algorithmusbegriff (Terminierung!) subsumieren“ lässt (S. 60). Auch Strecker (2009) definiert in ihrer Dissertation zunächst einen rein sequentiellen Algorithmusbegriff im Sinne einer mathematischen „*Funktion* [...], die Eingabezustände auf Ausgabezustände abbildet“ (Hervorhebung wie im Orig., S. 18). Die Systeme, die ihre Schülerinnen und Schüler jedoch später entwerfen, sind Steuerungssysteme für Puppenhäuser, d.h. für Markisen, Türen und Beleuchtungen (vgl. S. 89-90), bei denen Fragen nach dem Terminierungsverhalten oder gar der berechneten Funktion geradezu absurd wirken. Solche Systeme arbeiten in der Regel verteilt und auch Streckers Lernende begannen wie selbstverständlich, Parallelität in ihren Modellen zu verwenden (vgl. S. 43). Petrinetze als Modellierungstechnik lehnt sie jedoch explizit ab:

> [E]in Petri-Netz [spiegelt] nicht die Realität eines zugrunde liegenden Systems wieder. Wenn die Systemkonfiguration technischer Systeme durch allgemeinbildende Aspekte wie „Weltorientierung“ legitimiert werden soll, dann sollten auch zugrunde liegende Rechenmodelle der Realität verwendet werden. Und das sind nach wie vor nur scheinbar parallele, intern jedoch sequentielle Verarbeitungssysteme. (ebd., S. 44)

Wie absurd die Vorstellung einer sequentiellen Welt mit einem globalen Taktgeber ist, habe ich bereits in der Einleitung dargestellt. Selbst bei einem Einkernrechner, der nur mit einem Peripheriegerät wie einer Festplatte oder einem Drucker kommuniziert, lässt sich diese Auffassung nur schwer aufrecht erhalten. Im Zeitalter von Mehrkernprozessoren, globalen Netzwerken und dem „Internet der Dinge“ verliert sie jegliche Legitimation. Die obige Aussage muss meiner Meinung nach daher ins exakte Gegenteil verkehrt werden: eine „Weltorientierung“ verlangt ge-

rade *keine* ausschließliche Orientierung an sequentiellen Verarbeitungssystemen, sondern muss auch verteilte und interaktive Systeme berücksichtigen.

Meiner Meinung nach steht das obige Zitat von Strecker exemplarisch für eine grundlegende Fehlannahme vieler Informatik-Didaktiker, nämlich dass Algorithmen sequentiell arbeiten und terminieren müssen. Das EVA-Prinzip scheint in den Köpfen vieler zum Dogma geworden zu sein und wird somit auch auf Kontexte und Systeme angewandt, deren Funktionsweisen es nicht oder nur unzureichend zu erklären vermag. Welche Ausgabe erzeugt ein Schließsystem? In welchem Zustand befindet sich das Internet? Wann ist eine Ampel damit fertig, den Verkehr zu regeln, und terminiert?

Die im Rahmenlehrplan genannte „Grundstruktur informatischer Projekte“ (LISUM 2006b, S. 25), verlangt von den Schülerinnen und Schülern in der Regel, dass sie das zu erstellende System zunächst modellieren. Ein informatisches Modell ist nur dann sinnvoll, wenn es möglich ist, mit seiner Hilfe über die zentralen Funktionsweisen und Eigenschaften des modellierten Systems zu diskutieren. Das EVA-Prinzip und vom ihm geprägte Modelle wie Struktogramme oder endliche Automaten leisten dies für eingebettete Systeme nicht. Hier bedarf es einer Modellierungstechnik, die nicht auf der Annahme einer sequentiellen und terminierenden Verarbeitung basiert. Marwedel (2000) schlägt dazu StateCharts vor (vgl. S. 17). Zum Modellieren eingebetteter Systeme eignen sie sich gut. Petrinetze haben jedoch eine etwas einfachere Syntax und sind meiner Meinung nach im Rahmen des Informatikunterrichts vielseitiger einsetzbar (vgl. Abs. 3.2.2–3.2.5). Wenn die Schülerinnen und Schüler schon eine neue Modellierungstechnik lernen sollen, dann sollte diese auch eine möglichst breite Anwendung in der weiteren schulischen Laufbahn finden.

Nach dem hier vorgestellten Spiralcurriculum werden die Schülerinnen und Schüler im Rahmen dieses Themenbausteins zum ersten Mal mit Petrinetzen kon-

frontiert. Es ist daher eine entsprechende Einführung notwendig. Es reichen hier die grundlegenden Konzepte für elementare Netze: Plätze, Transitionen, Kanten, (schwarze) Marken und die Schaltregel. All diese Begriffe können rein informell und grafisch eingeführt werden. Eine formale Tupel-Definition halte ich nicht für sinnvoll, da sie weder eine veranschaulichende Funktion erfüllt, noch sinnvoll mit ihr im Unterricht gearbeitet werden kann. Meiner Meinung nach ist es vor dem Hintergrund der „Endlosschleife“ eines Betriebssystems auch möglich, das Konzept der Lebendigkeit zumindest intuitiv einzuführen. Dabei handelt es sich streng genommen um ein eher fortgeschrittenes Konzept, das auf dem Problem der Erreichbarkeit basiert (vgl. Reisig 2010, S. 151-152). Bei den im Folgenden vorgestellten Beispielen überzeugt man sich jedoch sehr schnell davon, dass die Modelle stets in ihre Anfangsmarkierung zurückkehren und somit endlos laufen können. Wie oben bereits erwähnt, enthalten viele interaktive Systeme zudem Nebenläufigkeit. Auch bei den folgenden Beispielen ist dies der Fall. Meiner Meinung nach muss dieses Konzept an dieser Stelle aber nicht explizit besprochen werden. Nebenläufigkeit ist ein relativ intuitives Konzept (vgl. Abs. 3.2.3), das den Lernenden hier schlicht als der Normalfall präsentiert wird.

Abbildungen 6, 7 und 8 zeigen nun Beispiele für Petrinetz-Modelle eingebetteter Systeme, die sich meiner Meinung nach für eine Behandlung im Unterricht

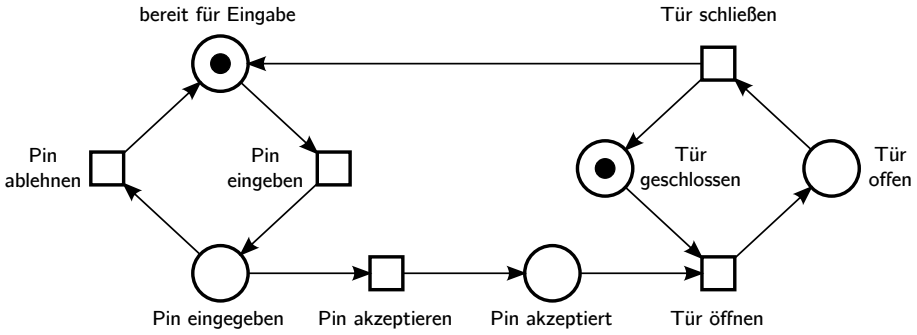


Abb. 6: Ein elektronisches Schließsystem

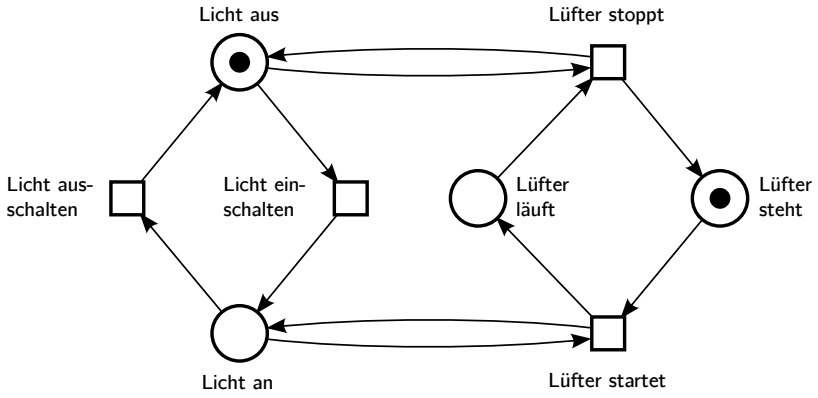


Abb. 7: Das Licht/Lüfter-System von Reisig (2010, S. 82)

eignen. Die Modelle veranschaulichen das grundlegende Verhalten der jeweils modellierten Systeme. In allen Modellen zeigt die Verteilung der Marken auf die Plätze, d.h. die aktuelle Markierung, dabei den aktuellen Zustand des Systems an. Den Modellen liegt also ein zustandsorientierter Ansatz zugrunde. Im Gegensatz zu endlichen Automaten erlauben sie jedoch Nebenläufigkeit und besitzen keinen Endzustand. Wie oben erwähnt, kehren sie stets in ihre Anfangsmarkierung zurück und laufen somit in einer „Endlosschleife." Außerdem sind die betrachteten Zustände nicht zwangsläufig global. In der Regel ist es intuitiver, von lokalen Zuständen der einzelnen Teilsysteme zu sprechen: beim Schließsystem von dem des elektronischen Schlosses und dem der Tür; beim Licht/Lüfter-System von dem des Lichtes und dem des Lüfters. Im Modell der Ampelsteuerung (Abb. 8) werden die lokalen Zustände nicht unmittelbar deutlich. Die Nebenläufigkeit der Transitionen anfordern und KFZ grün schalten macht aber dennoch deutlich, dass hier zwei Teilsysteme miteinander interagieren. Alle gezeigten Netze sind 1-beschränkt, d.h. auf jedem Platz befindet sich maximal eine Marke. Die Plätze können somit als Bedingungen interpretiert werden, die entweder erfüllt oder nicht erfüllt sind. Im Schließsystem kann beispielsweise die Tür nur dann geöffnet werden, wenn die die Tür geschlossen ist und die Pin akzeptiert wurde. Das Identifizieren solcher

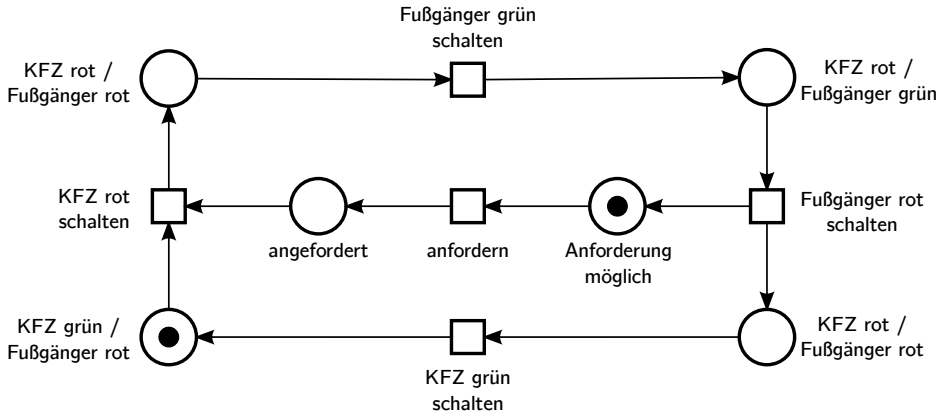


Abb. 8: Eine Variante der Ampelsteuerung von Rotering (2010, S. 23) ohne Kantengewichte

Bedingungen kann für eine spätere Implementation sehr hilfreich sein.

Das Licht/Lüfter-System eignet sich meiner Meinung nach gut für eine Implementation in Lego-Mindstorms. Ein Tastsensor kann als Lichtschalter benutzt werden. Das Licht selbst kann über eine LED realisiert werden, der Lüfter über einen einfachen Servomotor. Die Umsetzung des Schließsystems ist ein wenig aufwändiger, da hier ein Sensor benötigt wird, der das Öffnen und Schließen der Tür überwacht. Mit einem Ulltraschallsensor sollte dies aber prinzipiell realisierbar sein. Baumann (2011) beschreibt eine Umsetzung einer Ampelsteuerung wie aus Abb. 8 mittels Arduino (vgl. S. 56-58).

3.2.2 Netzwerkkommunikation

Informatiksysteme sind heutzutage in aller Regel vernetzt. Sie kommunizieren kabelgebunden oder mittels Funk über verschiedene lokale oder globale Netzwerke. Das bekannteste und zugleich komplexeste globale Netzwerk ist wohl das Internet mit seinen unzähligen Diensten und Nutzungsarten. Menschen nutzen es, um im Web zu surfen, zu chatten oder E-Mails zu verschicken. Dabei kommt eine Vielzahl von Kommunikationsprotokollen zum Einsatz: allen voran TCP/IP als Basis

aller Internet-Kommunikation und, darauf aufbauend, HTTP für Web-Adressen, SMTP, IMAP und POP3 zum Versenden und Empfangen von E-Mail, oder FTP und BitTorrent für die Dateiübertragung, um nur die gängigsten zu nennen.

Die Kommunikation in Netzwerken ist dabei prinzipiell asynchron: das Versenden und Empfangen einer Nachricht geschieht nicht zeitgleich. Der Absender kann nicht ohne Weiteres feststellen, wann bzw. ob seine Nachricht beim Empfänger eintrifft. Die Nachricht könnte ja auch verloren gegangen sein. Daraus folgt ein grundlegendes Problem der Kommunikation in asynchronen Netzwerken: Verlässlichkeit. Mit dem Sliding-Window-Protokoll bzw. seinem Spezialfall, dem Alternating-Bit-Protokoll, exisitiert jedoch ein Algorithmus, der eine verlässliche Datenübertragung auch bei Kanälen mit hohem Paketverlust sicherstellt (vgl. Reisig 1998, S. 107-112).

Nachrichten können jedoch nicht nur verloren gehen, sondern auch abgefangen, mitgelesen oder verändert werden. Ein weiterer kritischer Aspekt der Netzwerk-Kommunikation ist somit die Sicherheit und Vertraulichkeit der verwendeten Kanäle. Mittels einer Ende-zu-Ende Verschlüsselunge, z.B. via RSA, kann nicht nur die Vertraulichkeit, sondern auch die Integrität und Authentizität einer Nachricht sichergestellt werden.

Die von Gramm et al. (2012) für die Sekundarstufe I entwickelte Unterrichtsreihe *E-Mail (nur?) für Dich* behandelt viele der oben genannten Aspekte und Probleme. Die Reihe kann als Grundlage für den hier vorgestellten Themenbaustein dienen. Tabelle 2 fasst die relevanten Teile des Rahmenlehrplans der Sekundarstufe I zusammen.

Petrinetze können in einer Unterrichtsreihe zu Netzwerkkommunikation meiner Meinung nach zwei grundlegende Funktionen erfüllen. Die erste ist die Veranschaulichung der verschiedenen Netzwerk-Topologien. Abbildung 9 zeigt ein Modell zur E-Mail-Kommunikation, mit dem sich der Weg einer E-Mail vom Sen-

	Die Schülerinnen und Schüler:	**Inhalte und mögliche Kontexte:**
Leben mit vernetzten Systemen (S. 27)	• beschreiben Protokolle im Alltagsleben • erforschen im Software-Experiment die Struktur eines Standard-Internetdienstes • bewerten Nutzen und Risiken der Internetbenutzung	• Protokolle als Vereinbarungen zwischen Kommunikationspartnern [...], beispielhaft ein Internet-Dienst (z.B. E-Mail, Chat, Foren)
Automatische und technische Systeme (S. 29)	• dokumentieren Einsatzzweck und Arbeitsweise eines automatischen Systems • definieren das untersuchte System als Computer(-Netzwerk)	In diesem Zusammenhang können Inhalte aus den Bereichen der Kommunikation, der Vernetzung oder Modellierung sinnvoll integriert werden [...].

Tabelle 2: Bezug des Themenbausteins *Netzwerkkommunikation* zum Berliner Rahmenlehrplan der Sekundarstufe I (LISUM 2006b)

der zum Empfänger simulieren lässt. Entsprechend lassen sich im Modell viele der oben genannten Protokolle (TCP/IP, SMTP und POP3 bzw. IMAP) sowie unsichere Kanäle (Mail-Server, Internet) verorten. Das Netz könnte somit ohne Weiteres als Advance Organizer für die von Gramm et al. (2012) vorgestellte Unterrichtsreihe verwendet werden. Es ließe sich auch problemlos zu einer allgemeineneren Client-Server-Struktur (mit nur einem Server) reduzieren.

Selbst wenn das Modell den Lernenden „nur“ als interaktive Veranschaulichungen zur Verfügung gestellt wird, leistet es einen Beitrag dazu, möglichst viele Lerntypen im Unterricht anzusprechen. Es kann und sollte aber auch von

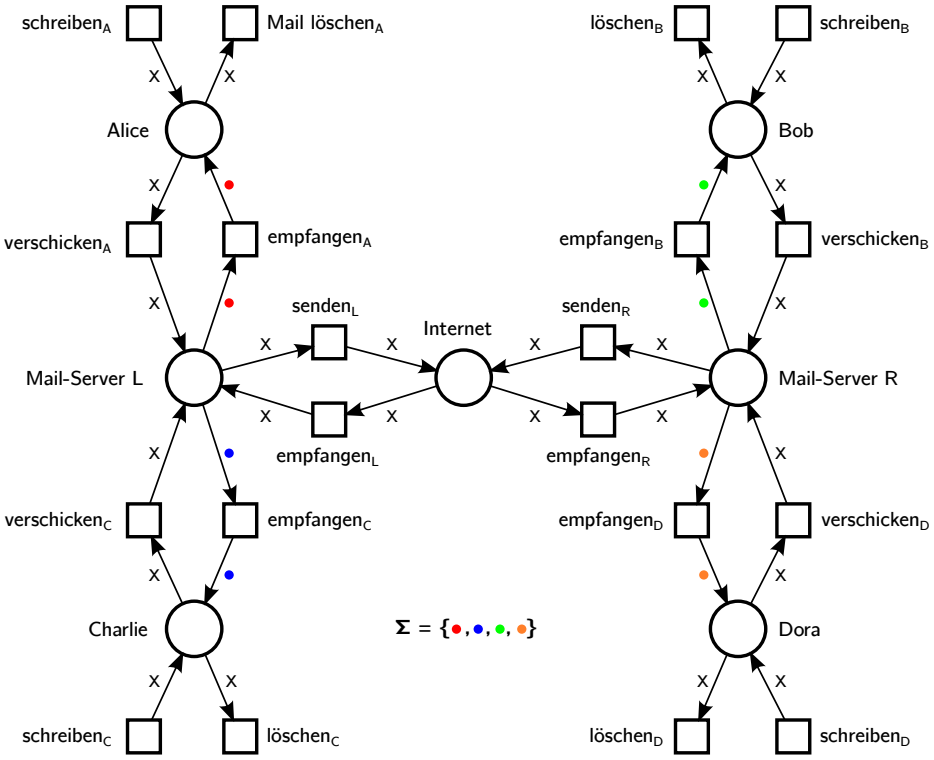


Abb. 9: E-Mail-Kommunikation

den Lernenden selbst erstellt oder zumindest bearbeitet werden, z.B. indem sie einen weiteren Kommunikationspartner hinzufügen – oder einen Angreifer, der versucht, die Kommunikation zu kompromittieren. An welchem Punkt des Netzwerkes könnte ein solcher Angreifer ansetzen?

Abbildung 10 zeigt ein Petrinetz-Modell für eine asymmetrische Verschlüsselung. Analog zu Abb. 9 lassen sich auch darin viele zentrale Begriffe, wie ver- und entschlüsseln, öffentliche und private Schlüssel sowie unsichere Kanäle, identifizieren. Es kann daher ebenso als Advance Organizer für eine entsprechende Unterrichtseinheit dienen. Einige interessante Effekte der asymmetrischen Verschlüsselung können auch bereits direkt am Modell diskutiert werden: z.B., dass Alice eine Nachricht, die sie mit Bobs öffentlichem Schlüssel kodiert hat, danach auch selbst nicht wieder entschlüsseln kann. Hat sie keine weitere unverschlüsselte

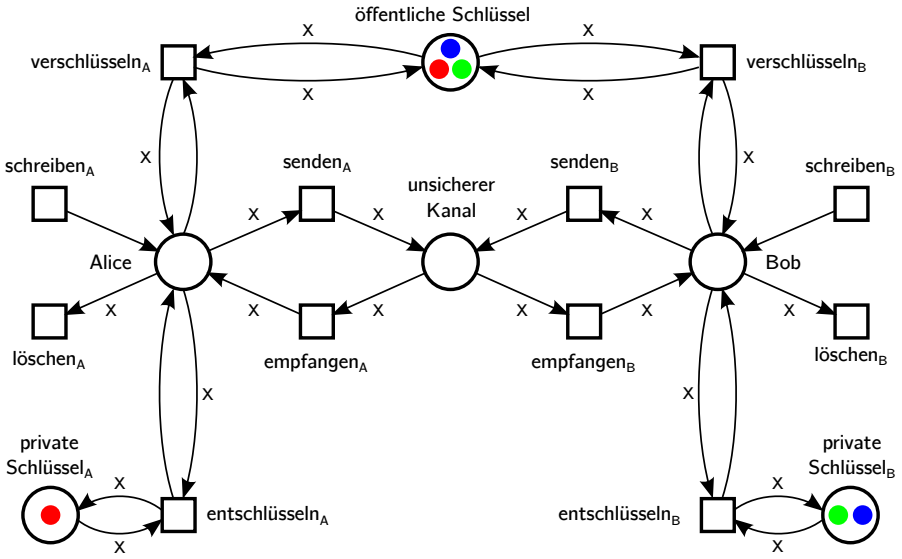


Abb. 10: Asymmetrische Verschlüsselung

Kopie der Nachricht, kann sie sie nicht mehr lesen. Es ist auch nicht möglich eine Nachricht zu verschlüsseln, die im Nachhinein von mehr als einer anderen Person entschlüsselt werden kann. Eine Nachricht an mehrere Personen muss daher für jeden Empfänger individuell verschlüsselt werden.

Abbildung 10 ist bereits ein Beispiel für die zweite Funktion, die Petrinetze im Rahmen einer Unterrichtseinheit zu Netzwerken übernehmen können: die Modellierung von Protokollen. In Anlehnung an die in Abs. 3.2.1 dargestellte Dogmatik des EVA-Prinzips, wird dabei häufig übersehen, dass es sich auch bei einem Kommunikationsprotokoll um einen Algorithmus handelt (vgl. Reisig 1998, S. 107), nämlich um eine Handlungsvorschrift, die die jeweiligen Partner zu befolgen haben, wenn sie erfolgreich kommunizieren wollen.

Ein weiteres Beispiel, das sich mit Hilfe von Petrinetzen hervorragend veranschaulichen lässt, ist das oben erwähnte BitTorrent-Protokoll. Es wird vorrangig verwendet, um große Dateien zwischen den Knoten eines Netzwerks zu übertragen. Jeder Knoten erhält dabei benötigte Teile der Datei von den anderen und

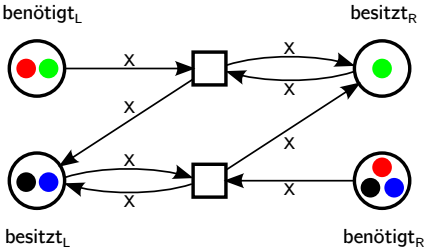


Abb. 11: BitTorrent-Protokoll mit zwei Peers

stellt selbst die Teile, die er bereits besitzt, zur Verfügung. Es handelt sich also um ein Peer-to-Peer-Protokoll. Abbildung 11 zeigt eine Verbindung zwischen zwei Peers des Netzwerks. Die farbigen Marken repräsentieren die verschiedenen Teile der zu ladenden Datei. Dabei gilt für jeden Knoten des Netzwerks und jedes Datei-Fragment, dass der Knoten das Fragment entweder bereits besitzt oder noch benötigt. Die beiden Knoten können nun die von ihnen besessenen Teile zwischen sich verteilen, wobei in Abb. 11 beide das rote Fragment nicht besitzen. Es ist also keine vollständige Kopie der Datei im Netzwerk vorhanden. Abb. 11 kann nun beliebig um Knoten erweitert werden. Auch die Anfangsmarkierung kann entsprechend variiert werden, um verschiedene Fragment-Verteilungen zu simulieren. Alle Abläufe eines so entstehenden Netzes enden immer in der selben Markierung: alle Knoten besitzen alle im Netzwerk vorhanden Datei-Fragmente, im Idealfall die vollständige Datei. Da jeder Knoten des Netzwerkes eigenständig Daten verteilt, arbeitet das Protokoll hochgradig verteilt. Im Petrinetz-Modell wird dies dadurch deutlich, dass in der Regel sehr viele Transitionen zugleich aktiviert sind und unabhängig voneinander schalten können. Genau dieser hohe Grad an Verteiltheit macht das Protokoll überhaupt erst so effektiv. Es muss wohl auch nicht weiter erläutert werden, dass es sich hervorragend eignet, um verschiedene gesellschafltiche und urheberrechtliche Kontexte zu motivieren.

Ein etwas anspruchsvolleres, aber ebenso interessantes Protokoll ist das oben erwähnte Alternating-Bit-Protokoll, welches eine verlässliche Kommunikation auch bei unzuverlässigen Kanälen sicherstellt. Seine Erarbeitung kann problemorien-

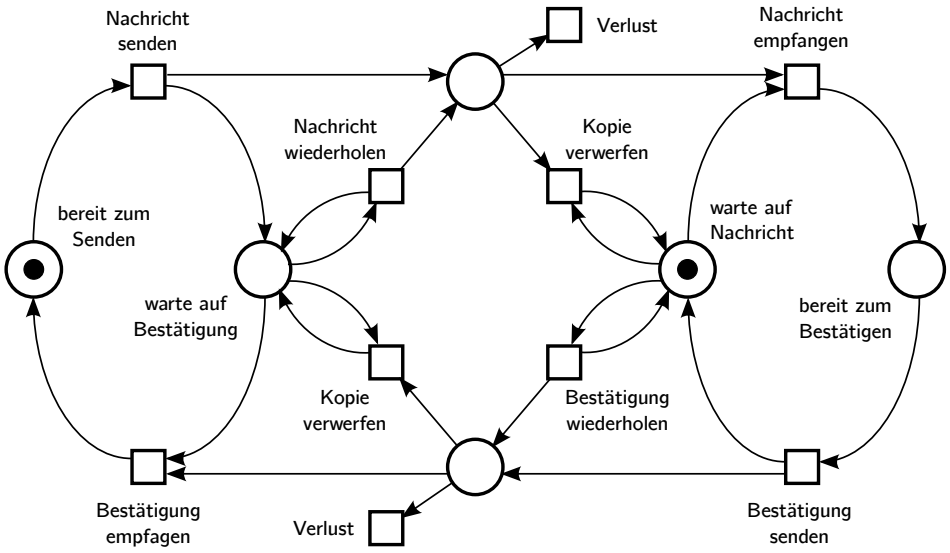


Abb. 12: Das Alternating-Bit-Protokoll von Reisig (ebd., S. 111) als elementares Petrinetz

tiert erfolgen. Die Ausgangssituation bildet der unzuverlässige Kanal: Nachrichten können abgefangen werden oder verloren gehen. Der Sender kann daher seine Nachrichten wiederholen und verlangt vom Empfänger, dass er eine Bestätigung schickt. Natürlich kann diese genauso abgefangen werden oder verloren gehen. Abbildung 12 zeigt ein didaktisch reduziertes Modell des Protokolls als elementares Petrinetz. Es veranschaulicht das prinzipielle Verhalten der Kommunikationspartner. Formal betrachtet ist das Modell jedoch nicht ganz korrekt:

> The receiver is unable to distinguish a new message from copies of old messages, and the sender may entirely ignore the arrival of receipts, thus forever repeating copies of a message instead of eventually receiving its receipt. (Reisig 1998, S. 108)

Beide Probleme lassen sich nicht trivial im Modell lösen. Zur Unterscheidung der verschiedenen Nachrichten braucht der fertige Algorithmus bei Reisig (ebd.) eine weitere Annahme: nämlich, dass sich der Kommunikationskanal wie eine FIFO-Warteschlange verhält, Nachrichen sich also nicht überholen können (vgl. S. 110). Ein solches Verhalten explizit im Netz zu modellieren ist aufwändig und würde

es erheblich komplexer machen. Stattdessen sollte mit den Lernenden intuitiv erarbeitet werden, dass die Nachrichten in der Realität natürlich Identifikationsnummern brauchen, um voneinander unterschieden werden zu können. Abschließend dürften die Frage, warum der Algorithmus denn „Alternating Bit“ heißt, und die Erkenntnis, dass bereits zwei verschiedene ID-Nummern zur Unterscheidung der Nachrichten ausreichen (bei der Annahme eines FIFO-Kanals), bei einigen Lernenden zu Erstaunen führen.

Das zweite Problem, das der ignorierten Bestätigungen, ist ein Fairness-Problem (vgl. Reisig 1998, S. 110). Analog kann natürlich auch der Empfänger die ursprünglichen Nachrichten ignorieren. Fairness ist ein nicht implementierbares Konzept, das sich mit der Menge aller Abläufe eines Systems beschäftigt und bestimmte unendliche Abläufe davon ausschließt (vgl. Reisig 2010, S. 200-201). Ich halte es für zu anspruchsvoll, um es im Informatikunterricht der Sekundarstufe I zu behandeln. Intuitiv sollte den Lernenden aber sofort ersichtlich sein, dass ein Agent irgendwann auch einmal nachschaut, ob eine Bestätigung bzw. eine Nachricht für ihn vorliegt. Ein weiteres Fairness-Problem ist der unsichere Kanal. Auch hier sollte schnell klar sein, dass natürlich kein Protokoll der Welt eine verlässliche Kommunikation gewährleisten kann, wenn immer alle Nachrichten verloren gehen oder abgefangen werden. Der Algorithmus basiert also auf der Annahme, dass irgendwann einmal eine Nachricht auch tatsächlich ankommt.

Die Systeme in den Abb. 9–11 verwenden farbige Marken und Kantengewichte mit einfachen Variablen als neue Modellierungskonzepte. Die farbigen Marken verhalten sich dabei intuitiv genauso wie die schwarzen. Schaltet eine Transition, an deren Kanten farbige Marken stehen, werden diese in den entsprechenden Plätzen erzeugt bzw. aus ihnen entfernt. Natürlich müssen in den Plätzen vor der Transition entsprechende Marken vorhanden sein, damit die Transition aktiviert ist. Auch die variablen Kantengewichte sind in der Regel nicht sonderlich kompli-

ziert. Kommt an den Kanten einer Transition nur eine einzige Variable vor (wie z.B. die Variable X an den Kanten der Transition senden_L in Abb. 9), so bedeutet dies lediglich, dass die Transition Marken beliebiger Farbe weiterleitet. Die Variable wird beim Schalten der Transition durch eine beliebige Markenfarbe ersetzt, vorausgesetzt diese kommt in den entsprechenden Plätzen vor der Transition vor. Hat die Transition keine eingehenden Kanten mit variablen Gewichten (wie z.B. Transition $\text{schreiben}_\text{A}$ in Abb. 9), so bedeutet dies intuitiv, dass die Transition eine Marke mit einer zufälligen Farbe produziert. Etwas komplizierter wird es lediglich, wenn verschiedene Variablen an den Kanten einer Transition vorkommen. Auch hier wird beim Schalten jede Variable durch eine Konstante ersetzt, wobei verschiedene Variablen durchaus durch die selbe Konstante ersetzt werden können. Dies macht es nicht immer einfach, sofort zu „sehen“ ob eine Transition aktiviert ist oder nicht. Dieser Fall kommt jedoch in keinem der hier betrachteten Netze vor.

Sowohl farbige Marken als auch Kantengewichte mit einfachen Variablen erhöhen die theoretische Ausdrucksstärke der Netze nicht, solange die Anzahl der Farben begrenzt ist (vgl. David und Alla 2005, S. 12). Sie erlauben es lediglich, übersichtlichere und somit intuitivere Netze zu erstellen. Ohne variable Kantengewichte müsste z.B. die Transition senden_L in Abb. 9 in vier Transition geteilt werden: eine für jede Markenfarbe. Ein Platz, auf dem sich vier verschiedene Markenfarben befinden, kann wiederum durch vier Plätze mit jeweils nur schwarzen Marken ersetzt werden.

Abgesehen von den neuen Modellierungskonzepten, müssen die Netze in den Abb. 9–12 auch auf eine andere Weise interpretiert werden als die Modelle der eingebetteten Systeme in Abs. 3.2.1. Die Marken auf den Plätzen signalisieren hier nicht mehr die Zustände des Systems. Stattdessen repräsentieren sie die Nachrichten, die durch das Netzwerk fließen. Es ist auch nicht mehr unmittelbar möglich,

die Plätze als Bedingungen zu interpretieren, die entweder erfüllt sind oder nicht. In vielen Fällen sind die Plätze eher Speicher, in denen sich Daten bzw. Nachrichten in Form der Marken befinden. Dieser Unterschied muss den Lernenden meiner Meinung nach unbedingt deutlich gemacht werden. Es ist schlicht nicht sinnvoll von einem Netzwerk zu reden, als würde es sich gerade in einem Zustand befinden, in dem dieser oder jener Knoten gerade eine Nachricht vorhält. Sinnvoller ist es, einzelne Nachrichten bei ihrem Weg durch das Netzwerk zu fokussieren.

Die hier vorgestellten Modelle eignen sich meiner Meinung nach auch gut um zwei grundlegende Konzepte der informatischen Modellbildung zu thematisieren: Abstraktion und das Treffen bestimmter Annahmen. Wie bereits erwähnt, entsprechen die Marken in den Netzen oft Nachrichten oder Daten, die im Netzwerk verschickt werden. Welche Informationen diese enthalten, z.B. welchen Inhalt eine E-Mail hat oder welche Datei via BitTorren verteilt wird, ist jedoch nicht in den Modellen dargstellt. Die Nachrichten wurden abstrahiert, da ihre Inhalte für die Funktionsweisen der Systeme keine Rolle spielen und die Modelle daher nur unnötig komplex machen würden. Ebenso setzen die Modelle an einigen Stellen bestimmte Annahmen über das jeweils modellierte System voraus: z.B. dass eine E-Mail auch irgendwann beim Empfänger eintrifft und nicht endlos im Netzwerk umherwandert, oder dass der Kanal beim Alternating-Bit-Protokoll auch irgendwann einmal eine Nachricht durchlässt. Die Fähigkeiten zur Abstraktion, d.h. zur Konzentration auf das Wesentliche, und zum bewussten Treffen bestimmter Annahmen über das zu modellierende System sind essenziell für jegliche Art der Modellbildung, insbesondere in der Informatik. Sie können anhand der hier vorgestellten Beispiele exemplarisch thematisiert und geübt werden.

3.2.3 Nebenläufige Prozesse

Wir leben in einer verteilten Welt. Die Vorstellung einer zentralen Kontrolleinheit, die den globalen „Takt“ der Gesellschaft, der Weltwirtschaft oder des Internet vorgibt, ist schlicht absurd. Die eingebetteten und vernetzten Systeme in den letzten beiden Abschnitten haben gezeigt, dass Nebenläufigkeit auch bei informatischen Systemen heutzutage eher den Normalfall als die Ausnahme darstellt. Solange die verschiedenen Prozesse dabei unabhängig voneinander laufen, entstehen daraus auch keine weiteren Probleme. Interessant wird es jedoch, wenn zwei Prozesse auf dieselbe Ressource zugreifen, also voneinander abhängig sind und sich synchronisieren müssen. Diese Abhängigkeiten können unterschiedliche Formen annehmen. Beispielsweise kann die Arbeit eines Prozesses auf ein Ergebnis oder ein Produkt eines anderen angewiesen sein, z. B. ein Mechaniker, der auf die Produktion und Lieferung verschiedener Bauteile angewiesen ist. Eine andere Möglichkeit ist, dass zwei Prozesse um eine gemeinsame knappe Ressource konkurrieren, z.B. die Kopierer in einer Bibliothek. Sind alle Kopierer in Benutzung, muss man warten, bis wieder einer frei wird. Das wohl berühmteste Beispiel für solch einen wechselseitigen Ausschluss sind die speisenden Philosphen von E.W. Dijkstra. Im schlimmsten Fall kommt es hier zu einer Verklemmung, nämlich dann, wenn jeder Philosoph genau eine Gabel in der Hand hält und darauf wartet, dass sein Nachbar die jeweils andere wieder freigibt.

Aus didaktischer Sicht lässt sich festhalten, dass Nebenläufigkeit zunächst ein sehr intuitives Konzept zu sein scheint. Wie oben bereits erwähnt, begannen mehrere von Streckers (2009) Lernenden „im Struktogramm eine nicht erlaubte Parallelität“ darzustellen (S. 43). Weigend (2007) macht bei einer Schülerin ähnliche Beobachtungen und kommentiert: „Die Parallelisierung verkürzt die Ablaufzeit, macht das Modell kompakter und verbessert den Gestaltcharakter“ (S. 94). Umso erstaunlicher ist es, dass sich keines der von ihm diskutierten „intuitiven

Abschlussorientierte Standards:	**Kompetenz- und Inhaltsbezug:**
Informatisches Modellieren (S. 14): • entwickeln, implementieren, testen und validieren einfache Modelle Informatiksysteme verstehen (S. 15): • diskutieren Funktionalität, Leistungsfähigkeit und Zuverlässigkeit von Informatiksystemen	Softwareentwicklung (S. 21): • Algorithmen und Datenstrukturen • Darstellung von Algorithmen in grafischer Form • Simulation (dynamische Systeme, Automaten)

Tabelle 3: Bezug des Themenbausteins *Nebenläufige Prozesse* zum Berliner Rahmenlehrplan der Sekundarstufe II (LISUM 2006a)

Modelle der Informatik“ explizit mit Nebenläufigkeit befasst. Auch im Berliner Rahmenlehrplan der Sekundarstufe II werden nebenläufige Prozesse und die daraus resultierenden Probleme an keiner Stelle explizit erwähnt. Tabelle 3 zeigt diejenigen Ausschnitte, unter denen sich das Thema verorten lässt. Es könnte demnach im ersten oder zweiten Kurshalbjahr im Rahmen des Themenkomplexes „Softwareentwicklung“ unterrichtet werden. Ein zentraler Bestandteil dieses Themas ist das Modellieren und Analysieren von Algorithmen.

Die Tatsache, dass der Rahmenlehrplan nicht auch Nebenläufigkeit in diesem Kontext explizit nennt, mag daran liegen, dass sie methodisch für zu anspruchsvoll gehalten wird. Nur wenige didaktische Arbeiten beschäftigen sich mit dem Thema und kaum eine schafft es dabei, auf die Ebene der Programmierung zu verzichten. Letzteres liegt vermutlich daran, dass die verteilten Prozesse für die Lernenden interaktiv erfahrbar gemacht werden sollen. Schildknecht (2003) beschreibt das daraus entstehende Dilemma sehr treffend:

> Versucht man, die Begriffe und Verfahren [nebenläufiger Prozesse] unabhängig von irgendeiner Programmiersprache zu entwickeln, gerät der Unterricht möglicherweise zu theoretisch. Denn erst die Realisierung in einem lauffähigen Programm gestattet es den Schülern, zu beurteilen, ob die entwickelten Konzepte praktisch brauchbar und trag-

> fähig sind. Geht man dagegen von der Programmiersprache aus, läuft man Gefahr, deren Sprachkonstrukte in den Vordergrund zu stellen und damit dem Odium des Programmier(sprachen)kurses zu verfallen. (S. 17).

Hinzu kommt, dass das Programmieren von Nebenläufigkeit keineswegs trivial ist. Threads müssen korrekt erzeugt, gegebenenfalls synchronisiert und terminiert werden. Will man wie Schildknecht (ebd., S. 19) oder Hermes (2003, S. 29-32) ihre Abläufe zudem noch grafisch veranschaulichen, ist die Gefahr groß, dass am Ende mehr Zeit für die Klärung programmiertechnischer Fragen aufgewendet wird als für die eigentliche Betrachtung der Probleme nebenläufiger Prozesse.

Im Gegensatz zu den ersten beiden Themenbausteinen ist die Behandlung nebenläufiger Prozesse im Informatikunterricht also alles andere als methodisch ausgereift. Petrinetze können hier einen wertvollen Beitrag leisten, indem sie selbst die Funktion der Methode bzw. des Lernmediums übernehmen. Sie erlauben es, nebenläufige Prozesse grafisch zu modellieren und vor allem auch ihre Abläufe interaktiv zu simulieren. Sie ermöglichen es somit, die oben genannten, typischen Probleme verteilter Systeme zu diskutieren und auch interaktiv erfahrbar zu machen, ohne dabei sofort auch über das Programmieren von Threads nachdenken zu müssen. Ich bin sogar der Meinung, dass auf die abschließende Software-Implementation im Zweifel verzichtet werden kann. Ich stimme an dieser Stelle mit Strecker (2009) überein:

> Erfahrungen in der Schule zeigen, dass ein Großteil der Zeit bei der Codierung in textbasierten Sprachen dafür aufgewendet werden muss, Syntaxfehler zu finden und zu korrigieren, während die Erstellung oder das Verständnis für den zugrunde liegenden Algorithmus bereits verinnerlicht ist. (S. 8)

Eine Software-Implementation mit Threads und grafischen Oberflächen sollte daher zumindest nicht ohne wirklich guten Grund geschehen.

Abbildung 13 zeigt eine Variante der speisenden Philosophen, mit dem sich die Probleme des wechselseitigen Ausschlusses und der Verklemmung sehr gut

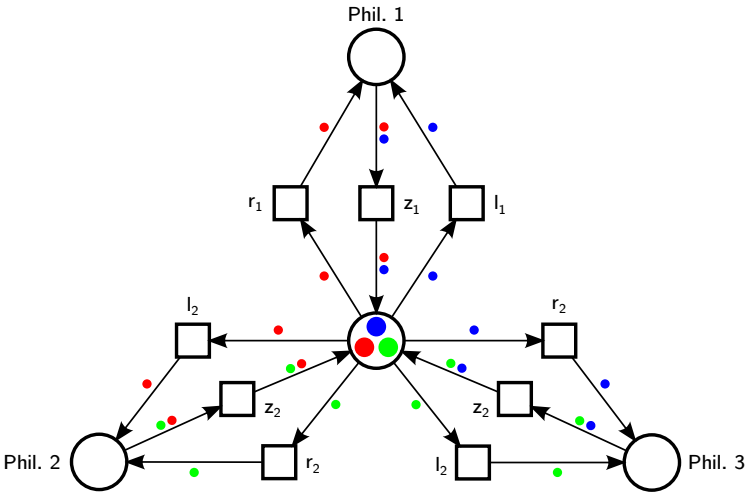


Abb. 13: Drei speisende Philosophen mit möglicher Verklemmung

veranschaulichen lassen. Die farbigen Marken repräsentieren die Gabeln, welche von den jeweiligen Philosophen nacheinander aufgenommen werden. Nimmt ein Philosoph seine rechte bzw. linke Gabel, steht diese für seinen rechten bzw. linken Nachbarn nicht mehr zur Verfügung. Die Reihenfolge, in der ein Philosoph seine Gabeln nimmt, ist in diesem Modell nicht festgelegt. Es kann also in zwei verschiedene Verklemmungen geraten, in denen jeder Philosoph entweder seine linke oder seine rechte Gabel in der Hand hält. Dieses Problem kann in Abb. 13 sehr einfach gelöst werden, indem die Transitionen l und r der Philosophen zu je einer Transition vereinigt werden. Jeder Philosoph würde somit seine beiden Gabeln gleichzeitig aufnehmen. Die Aktion wäre nun also atomar. Weitere Unterrichtsbeispiele zu wechselseitigen Ausschlüssen und die entsprechenden Petrinetz-Modelle liefern Schwill (1993, S. 48-49) und Rotering (2010, S. 14-19). Reisig (2010) diskutiert anhand einer sehr elementaren Variante auch das Problem der Fairness (vgl. S. 197-201), welches hier nicht weiter berücksichtigt wurde.

Prozesse müssen jedoch nicht immer miteinander konkurrieren. Sie können auch gemeinsam in Kooperation eine übergeordnete Aufgabe lösen, wobei ihre

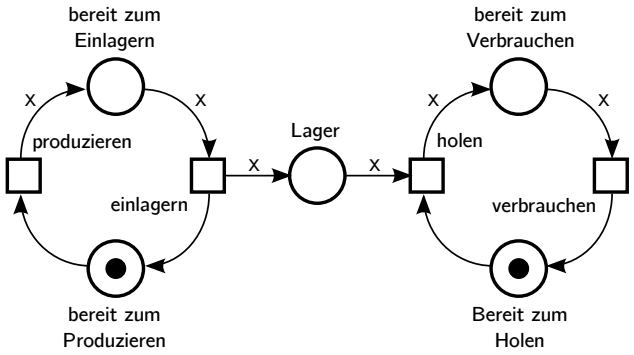


Abb. 14: Erzeuger/Verbraucher-System, basierend auf Reisig (1998, S. 230)

Aktionen mitunter voneinander abhängen. Das wohl einfachste Beispiel einer solchen Abhängigkeit ist die eines Verbrauchers zu einem Erzeuger. Abbildung 14 zeigt ein einfaches Beispiel eines solchen Systems. Der Verbraucher kann hier offensichtlich nur so lange arbeiten wie der Erzeuger auch Ressourcen produziert. Er ist vom Erzeuger abhängig. Im Gegensatz kann der Erzeuger jedoch problemlos Ressourcen produzieren, ohne dabei darüber nachdenken zu müssen, ob diese auch irgendwann konsumiert werden.

Das Modell in Abb. 14 ließe sich nun sehr einfach erweitern, sodass auch der Verbraucher wiederum ein Produkt erzeugt, die Ressource also eigentlich nicht verbraucht, sondern weiterverarbeitet. Nach diesem Prinzip lassen sich nun Produktionsketten bilden, in denen jeder Prozess einen Verarbeitungsschritt auf

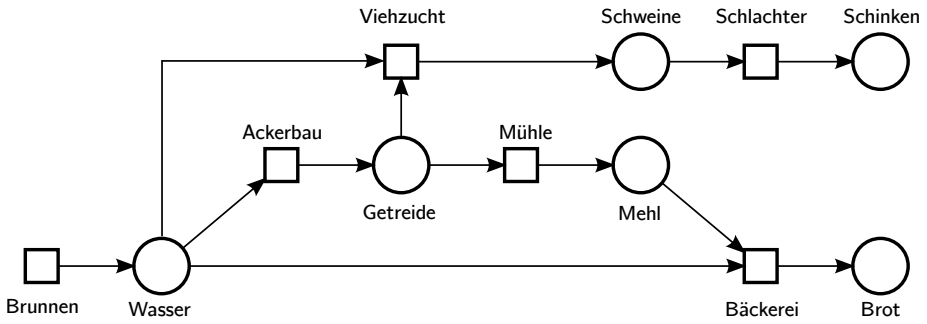


Abb. 15: Eine idealtypische Produktionskette

dem Weg zum fertigen Produkt darstellt. Abbildung 15 zeigt ein idealtypisches Beispiel aus der Landwirtschaft. Die Prozesse aus Abb. 14 wurden hier zu einfachen Transitionen abstrahiert. Solche Produktionsketten sind im Allgemeinen azyklisch: keine Ressource durchläuft zweimal denselben Verarbeitungsschritt. Sie eignen sich daher, ähnlich wie verteilte Abläufe, sehr gut um kausale Abhängigkeiten zu veranschaulichen (vgl. Reisig 2010, S. 58-60). In Abb. 15 ist ein Produktionsschritt jeweil abhängig von seinen jeweiligen Rohstofflieferanten: z.B. der Schlachter von der Viehzucht oder die Mühle vom Ackerbau. Ebenso wird das Konzept der Nebenläufigkeit bzw. kausalen Unabhängigkeit deutlich, welche im Gegensatz zur Abhängigkeit nicht transitiv ist: die Bäckerei arbeitet unabhängig vom Schlachter und der Schlachter unabhängig von der Mühle. Die Bäckerei ist jedoch abhängig von der Mühle. Ebenso lassen sich Konfliktsituationen finden: Viehzucht und Mühle stehen im Konflikt über das produzierte Getreide. Das Netz in Abb. 15 könnte im Rahmen einer Rechercheaufgabe auch um realistische Kantengewichte erweitert werden: Wie viele Liter Wasser und wie viele Kilo Futter(getreide) werden eigentlich im Schnitt benötigt, um ein Schwein aufzuziehen? Das Modell kann somit auch benutzt werden um verschiedene gesellschaftliche Fragen zu motivieren.

Kausale Ordnungen sind, im Gegensatz zu zeitlichen, nicht zwangsläufig total. Dies wird zum Beispiel deutlich, wenn man versucht, im Quelltext eines sequentiellen Programms diejenigen Befehle zu identifizieren, die tatsächlich voneinander abhängen, deren Reihenfolge also nicht geändert werden darf. Das Erkennen kausaler Abhängigkeiten kann somit eventuell auch einen Beitrag zur Programmierkompetenz leisten. In jedem Fall bin ich der Ansicht, dass das Erkennen und Diskutieren kausaler Zusammenhänge im Alltag einen enormen Beitrag zum allgemeinbildenden Wert des Informatikunterrichts leisten kann. Den Anwendungsgebieten sind dabei kaum Grenzen gesetzt: möglich wären Produktions- und

Verarbeitungsketten, oder sogar kausale Zusammenhänge historischer Ereignisse. Wikarski (1997a) diskutiert Geschäftsprozesse und Organisationsstrukturen (vgl. S. 78-79), die vor allem an beruflichen Gymnasien und Oberstufenzentren interessant sein dürften.

Weitere Anwendungsbeispiele und Aufgabenstellungen zu konkurrierenden und kooperierenden nebenläufigen Prozessen finden sich bei Hubwieser et al. (2010, S. 51-55). Die Systeme lassen sich ebenfalls alle mithilfe von Petrinetzen modellieren, simulieren und analysieren.

Soll auf das anschließende Programmieren nebenläufiger Prozesse dennoch nicht verzichtet werden, schlage ich vor, dabei auf das Erzeugen von Threads zu verzichten und stattdessen ein „echtes“ verteiltes System über das Netzwerk aufzusetzen. Jede Schülerin und jeder Schüler implementiert dabei einen Agenten des Systems (z. B. einen Verarbeitungsschritt in einer Produktionskette), der nur über das lokale Netzwerk mit den anderen Agenten kommunizieren kann. Natürlich muss dafür ein entsprechendes Kommunikationsprotokoll definiert werden (vgl. Abs. 3.2.2). Am Ende hängt es somit auch von der Fehlerfreiheit jedes einzelnen Agenten ab, ob das Gesamtsystem korrekt funktioniert – ein weiterer kritischer Aspekt verteilter Systeme. Alternativ existieren mit *Scratch* oder *MultiKara* auch didaktische Programmierumgebungen, in denen Nebenläufigkeit sehr einfach implementiert werden kann.

3.2.4 Formale Sprachen

Eine formale Sprache ist zunächst nichts weiter als eine Menge aus Wörtern über einem gegebenen Alphabet. Die Theorie der formalen Sprachen beschäftigt sich nun grundlegend mit verschiedenen Fragen, die man an solche Mengen stellen kann: Nach welchen Regeln können sie erzeugt werden? Ist für ein gegebenes Wort entscheidbar, ob es in einer gegebenen Sprache liegt oder nicht? Lassen

Abschlussorientierte Standards:	**Kompetenz- und Inhaltsbezug:**
Informatiksysteme verstehen (S. 15): • vergleichen formale und natürliche Sprachen	Sprachen und Automaten (S. 22): • Vergleich natürlicher und formaler Sprachen
Problemlösen (S. 16): • beachten Grenzen des Problemlösens mit Informatiksystemen	• Syntax und Semantik • zustandsorientierte Modellierung • endliche Automaten • Kellerautomat • Chomsky-Hierarchie

Tabelle 4: Bezug des Themenbausteins *Formale Sprachen* zum Berliner Rahmenlehrplan der Sekundarstufe II (LISUM 2006a)

sich Sprachen entsprechend in Klassen einteilen? Zur Bearbeitung dieser und vieler anderer Fragen bedient sich die theoretische Informatik verschiedener Modelle, wie Grammatiken, Zustandsautomaten und, nicht zuletzt, Turing-Maschinen. Die wohl berühmteste Klassifizierung formaler Sprachen ist die Chomsky-Hierarchie. All diese Konzepte sind vermutlich ein verbindlicher Teil eines jeden Informatikstudiums und sollen hier daher nicht weiter erörtert werden. Eine gut lesbare Einführung findet sich z.B. bei Schöning (2008).

Auch in der Informatik-Didaktik herrscht weitgehend ein Konsens darüber, dass formale Sprachen ein wichtiger Teil der Informatik sind. Schubert und Schwill (2011) zählen sie sogar zu den fundamentalen Ideen der Informatik (vgl. S. 69-70). Auch im Berliner Rahmenlehrplan der Sekundarstufe II ist das Thema fest verankert. Tabelle 4 zeigt die relevanten Ausschnitte. Eine entsprechende Unterrichtseinheit könnte beispielsweise problemorientiert durch den Compilerbau motiviert werden (vgl. Brandes 2009). Die Sprachen der Gleitkommazahlen und die der zulässigen Variablennamen sind regulär. Die Sprache der arithmetischen Ausdrücke

und auch Programmiersprachen sind kontextfrei. Hubwieser et al. (2010) nennen verschiedene weitere Anwendungsbeispiele und Aufgabenstellungen (vgl. S. 10-43), aber vermutlich hat jede hinreichend erfahrene Informatiklehrkraft mittlerweile eine eigene Vorstellung davon, wie sich formale Sprachen und Automatentheorie im Unterricht umsetzen lassen.

Petrinetze können die Betrachtung formaler Sprachen nun auf eine interessante Weise bereichern, denn auch sie können dazu benutzt werden, bestimmte Sprachen zu erkennen. Ein Wort gehört dabei genau dann zu der Sprache eines (elementaren) Petrinetzes, wenn es einem sequentiellen Ablauf des Netzes entspricht. Dabei sind sequentielle Abläufe das einzig neue Konzept, das an dieser Stelle eingeführt werden muss. Das Netz in Abb. 16 hat beispielsweise genau drei sequentielle Abläufe und erkennt entsprechend die Sprache $L_1 = \{a, b, bc\}$. Die Menge der zulässigen Abläufe kann nun auf verschiedene Weisen eingeschränkt werden, was zu unterschiedlichen Sprachklassen führt. Auch die Beschriftungen der Transitionen können eingeschränkt werden, indem doppelte Anschriften oder leere Anschriften nicht zugelassen werden. Eine ausführliche formale Definition findet sich bei Priese und Wimmel (2008, S. 189-191).

Für die schulische Behandlung kann die Menge der Abläufe auf genau diejenigen eingeschränkt werden, die in der leeren Markierung enden, also keine Marken im Netz zurücklassen und somit insbesondere keine weiteren Transitionen aktivieren. Eine Transitionsbeschriftung darf in einem Netz mehrfach vorkommen. Leere Anschriften sollen jedoch nicht zugelassen sein. Formal handelt es sich damit um

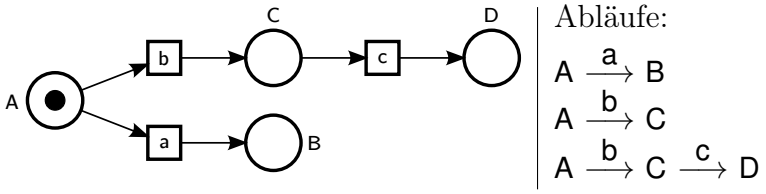


Abb. 16: Petrinetz mit drei sequentiellen Abläufen

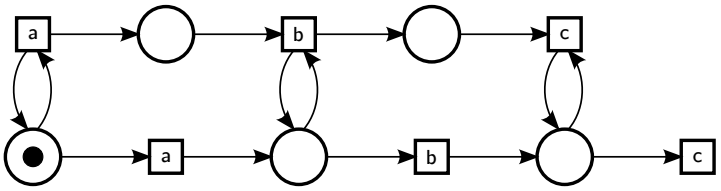


Abb. 17: Petrinetz mit nicht-kontextfreier Sprache nach Priese und Wimmel (2008, S. 191)

eine Teilklasse der „terminalen Sprachen von τ-freien Petri-Netzen“ (Priese und Wimmel 2008, S. 190). Nach dieser Definition zeigt Abb. 17 nun ein Netz, das die Sprache $L_2 = \{a^n b^n c^n \,|\, n \geq 1\}$ erkennt. Diese ist bekanntlich nicht kontextfrei, kann also nicht von einem Kellerautomaten erkannt werden. Im Gegenzug gehören quasi die „Haussprachen“ des Kellerautomaten, die Palindromsprachen $PAL = \{ww^R \,|\, w \in \Sigma^*\}$, zu keiner Petrinetz-Sprachklasse (vgl. Haddad und Poitrenaud 1999, S. 229), also insbesondere auch nicht zu der hier betrachteten. Abbildung 18 veranschaulicht das Verhältnis der Petrinetz-Sprachen zu Teilen der Chomsky-Hierarchie.

Im Rahmen einer Unterrichtseinheit zu formalen Sprachen kann ein kurzer Exkurs zu Petrinetz-Sprachen dazu benutzt werden, um eine sehr grundlegende Frage zu motivieren. Die beiden oben genannten Beispiele der Sprachen $\{a^n b^n c^n \,|\, n \geq 1\}$ und der Palindrome reichen aus, um zu veranschaulichen, dass sich Petrinetz-Sprachen offensichtlich nicht einfach in die Klassen der Chomsky-

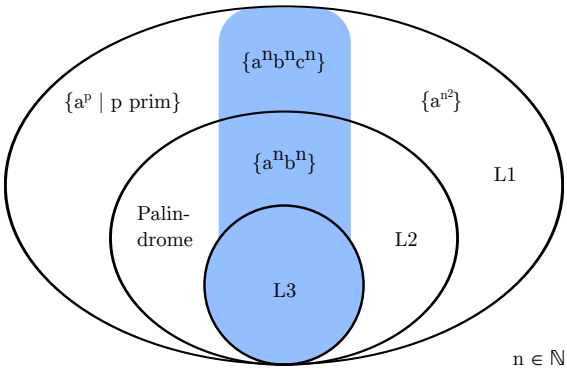


Abb. 18: Petrinetz-Sprachen (blau) im Verhältnis zu den Sprachklassen L1–L3 der Chomsky-Hierarchie

Hierarchie einordnen lassen. Es stellt sich also die Frage, warum diese eigentlich aus genau diesen Klassen besteht. Warum spielen ausgerechnet endliche Automaten, Kellerautomaten und Turingmaschinen eine derart dominante Rolle bei der Klassifizierung formaler Sprachen? Offenbar könnte man ja auch ganz andere Klassen betrachten.

Die Antwort auf diese Frage ist sicherlich vielschichtig, kann den Lernenden aber mithilfe zweier Perspektiven nahe gebracht werden: die historische und die praktische. Die klassischen Automatenmodelle arbeiten alle nach dem EVA-Prinzip. Sie lesen eine Eingabe, verarbeiten diese sequentiell und terminieren (hoffentlich!) irgendwann mit einer Ausgabe, was exakt der Arbeitsweise historischer Mainframe-Rechner entsprach. Dies änderte sich erst in den 1960er Jahren mit dem Aufkommen des Time-Sharing und der Geburtsstunde des Betriebssystems. Zwar existieren auch Petrinetze bereits seit den 60er Jahren, allerdings ist, wie in dieser Arbeit bereits mehrfach angedeutet, ein einmal etabliertes Paradigma nur schwer wieder aus den Köpfen zu vertreiben: „Mit seinen Vorschlägen zur Modellierung diskreter, asynchroner Systeme kam Petri in den 1960er Jahren für die Praxis zu früh und für die Theorie mit einem ‚falschen' Thema" (Reisig 2010, S. 233).

In praktischer Hinsicht sind vor allem Erkenntnisse über endliche Automaten und Kellerautomaten bzw. die von ihnen erkannten Sprachen interessant. Mit ihrer Hilfe lassen sich effiziente Suchalgorithmen und zuverlässige Compiler bauen. Auch Petrinetze können zur Analyse und Klassifikation formaler Sprachen eingesetzt werden. Die daraus gewonnen Erkenntnisse sind jedoch selten praktisch anwendbar. Tatsächlich ist mir keine Quelle bekannt, die ein solches Anwendungsgebiet beschreibt. Reisig (ebd.) beschreibt Petrinetz-Sprachen sogar als „am Kern der Sache vorbei" (S. 233). In einem persönlichen Gespräch teilte mir Reisig (2013) mit, dass für viele Probleme, die mittels Petrinetz-Sprachen ge-

löst werden könnten, oft effizientere und elegantere Lösungen exitierten, die ohne sie auskämen.

Die Beantwortung der Frage ist somit fundamental modelltheoretisch: Modelle werden dort eingesetzt, wo sie sinnvoll sind, wo sie das zu modellierende System akkurat abbilden und praktisch relevante Erkenntnisse darüber liefern können. Im Kontext formaler Sprachen gilt dies für Petrinetze, wenn überhaupt, nur sehr eingeschränkt. Die ausschließliche Betrachtung sequentieller Abläufe unterminiert eines der Kernkonzepte von Petrinetzen: Nebenläufigkeit. Genau aus diesem Grund sind sie hier jedoch dazu geeignet, um den Lernenden die Zweckbindung informatischer Modelle nahezubringen. Das perfekte Modell existiert nicht. Petrinetze eignen sich hervorragend zur Modellierung von nebenläufigen Prozessen, was die Lernenden auch breits im vergangenen Themenbaustein erfahren haben. Zur Beschreibung formaler Sprachen und sequentiellen Verhaltens greift man jedoch lieber auf andere Modelle zurück.

3.2.5 Softwareprojekt

Kaum ein anderes Gebiet hat in der Informatik eine derart zentrale und allgegenwärtige Stellung wie die Softwareentwicklung. Der englische Begriff *software engineering* macht deutlich, dass es sich dabei heutzutage keineswegs mehr um ein willkürliches Tüfteln handelt, sondern um einen hochgradig strukturierten Prozess, dessen Komplexität mit der von Ingenieursarbeit vergleichbar ist. Diese Komplexität muss durch geeignete Modelle reduziert werden. Nur mithilfe von Modellen können komplexe Softwaresysteme verstanden und anlysiert werden. Modelle formulieren zudem verbindliche Vorgaben und Anforderungen an das fertige System und stellen überhaupt erst eine Kommunikationsgrundlage für die verschiedenen beteiligten Personen her. Ohne ein entsprechendes Modell ist es oft gar nicht möglich, über ein System zu kommunizieren, weil eine geeignete

Terminologie und eine verbindliche Abgrenzung von Komponenten fehlen.

Auch in der Informatik-Fachdidaktik gibt es kaum ein anderes Thema, das so intensiv diskutiert wurde und wird wie die Softwareentwicklung, insbesondere die Programmierung. Der Berliner Rahmenlehrplan der Sekundarstufe II nennt das Softwareprojekt als einzigen verbindlichen Inhalt des vierten Kurshalbjahres (vgl. LISUM 2006a, S. 27). Projektarbeit bezeichnet dabei eine grundlegend handlungs- und produktorientierte Unterrichtsform, die von den Lernenden möglichst selbstständig geplant und durchgeführt wird (vgl. Gudjons 2008, S. 76-92). Das Softwareprojekt eignet sich somit, um eine Vielzahl verschiedener Kompetenzbereiche zu fördern. Tabelle 5 zeigt die Ausschnitte des Rahmenlehrplans, die hier meiner Meinung nach von besonderer Relevanz sind.

Zunächst geht aus Tabelle 5 insbesondere hervor, dass ein Softwareprojekt von den Lernenden verlangt, dass sie das System vorab modellieren. Der Rahmenlehrplan hält explizit fest, dass sich die Projektarbeit „nicht auf reine Programmiertätigkeiten beschränken“ darf (LISUM 2006a, S. 27). Der letzte Abschnitt hat jedoch deutlich gemacht, dass der Sinn und Nutzen eines Modells immer nur vor dem Hintergrund seiner Anwendung evaluiert werden kann. Für das konkrete Projektthema lässt der Rahmenlehrplan jedoch, völlig zurecht, eine große Wahlfreiheit:

> Das Thema sollte aus dem Erfahrungskreis der Schülerinnen und Schüler kommen [...] oder auf Themen aufbauen, die aus einem Vertiefungsgebiet bereits bekannt sind. (LISUM 2006a, S. 27)

In der Konsequenz kann an dieser Stelle nicht pauschal für den Einsatz von Petrinetzen im Rahmen des Projektes argumentiert werden. Genauso wenig können sie jedoch pauschal davon ausgeschlossen werden. Die Abs. 3.2.1–3.2.3 haben gezeigt, dass sich Petrinetze sehr gut zur Modellierung und Analyse interaktiver und eingebetteter Systeme, von Netzwerkkommunikation sowie nebenläufiger Prozesse

Abschlussorientierte Standards:	**Kompetenz- und Inhaltsbezug:**
Informatisches Modellieren (S. 14): • analysieren Realitätsauschnitte und wählen ein geeignetes Modellierungsverfahren aus • entwickeln, implementieren, testen und validieren einfache Modelle • reflektieren und beurteilen die eigene Modellierung Problemlösen (S. 16): • wenden die Phasen des Problemlöseprozesses (informelle Problembeschreibung, formale Modellierung, Implementierung und Realisierung, Bewertung und Modellkritik) an • setzen informatische Methoden [...] zielorientiert ein Kommunizieren u. Kooperieren (S. 16): • organisieren selbstständig Projektarbeit	Softwareentwicklung (S. 21): • Algorithmen und Datenstrukturen • Grundlagen systematischer Softwareentwicklung (Software-Life-Cycle)

Tabelle 5: Bezug des Themenbausteins *Softwareprojekt* zum Berliner Rahmenlehrplan der Sekundarstufe II (LISUM 2006a)

eignen. Es ist durchaus möglich, dass die Softwareprojekte der Schülerinnen und Schüler Berührungspunkte mit mindestens einem dieser Themen aufweisen.

Ferner macht Tabelle 5 auch noch einmal das eigenverantwortliche Arbeiten deutlich, das bei einem Projekt von den Lernenden gefordert wird. Sie sollen informatische Methoden zielorientiert einsetzen und insbesondere „ein geeignetes Modellierungsverfahren“ wählen (LISUM 2006a, S. 14). Zu entscheiden, wann ein bestimmtes Modellierungsverfahren sinnvoll eingesetzt werden kann, ist also ein zentraler Teil der Kompetenz Informatisches Modellieren. Meiner Meinung nach

wird dieser Aspekt jedoch zu oft im Unterricht vernachlässigt. Aufgaben und Probleme – auch im Rahmen von Projekten – werden häufig so formuliert, dass sie von vornherein auf eine ganz bestimmte Modellierungstechnik abzielen oder diese sogar explizit vorgeben, so z.B. auch bei Hubwieser et al. (2009, S. 143-144). Auch die im Rahmenlehrplan ersichtliche Dominanz der objektorientierten Softwaremodellierung ist ein Beispiel dafür (vgl. LISUM 2006a, S. 21). Natürlich müssen die Lernenden im Verlauf des Informatikunterrichts verschiedene idealtypische Beispiele für Systeme kennenlernen, die sich mehr oder weniger gut für die Modellierung mit einer bestimmten Technik eignen. Die Abs. 3.2.1–3.2.4 tun schlussendlich nichts anderes für Petrinetze. Im Rahmen eines Projektes, dessen Thema und Zielanforderungen maßgeblich von den Lernenden selbst bestimmt werden können, halte ich es jedoch nicht für sinnvoll, auch gleich die zu verwendende Modellierungstechnik vorzugeben. Vielmehr sollte von den Lernenden verlangt werden, ihr System aus mindestens zwei verschiedenen Sichten zu modellieren und selbst zu entscheiden, welche Modellierungsverfahren sie dazu für geeignet halten. Petrinetze sind im Rahmen des Projektes somit eines von mehreren „Werkzeugen im Werkzeugkoffer“ der Lernenden. Sie bereichern die Vielfalt der Informatischen Modelle im Unterricht und können somit einen Beitrag zur Modellierungskompetenz leisten.

4 Erprobung und qualitative Evaluation

Im Folgenden soll nun das in Abs. 3 vorgestellte Spiralcurriculum in Ansätzen validiert werden. Dazu werden zwei der vorgestellten Themenbausteine exemplarisch durch je einen Unterrichtsblock á 90 Minuten erprobt und qualitativ evaluiert. Aus den Erprobungen sollen insbesondere methodisch-praktische Erkenntnisse zur Vermittelbarkeit von Petrinetzen in den verschiedenen Jahrgangsstufen gewonnen werden (vgl. Abs. 2.1).

4.1 Leitfragen

Die folgenden Leitfragen liegen der Untersuchung zugrunde:

1. **Sind die eingeführten Petrinetz-Konzepte für die Lernenden fasslich? Wo liegen gegebenfalls Verständnisprobleme?**
   Die jeweils verwendeten Modellierungs- und Analysekonzepte, also der Aufbau und die Funktionsweise von Petrinetzen, müssen für die Lernenden fasslich sein. Dies ist eine offensichtliche Vorraussetzung für ihre spätere Anwendung, insbesondere im Hinblick auf die fortschreitende Curriculumspirale. Tritt an dieser Stelle eine Überforderung auf, müssen die entsprechenden Konzepte noch einmal überdacht und gegebenfalls weiter reduziert oder vereinfacht werden.

2. **Tragen Petrinetze dazu bei, dass die Lernenden die im Unterricht modellierten Systeme verstehen?**
   Modelle sind stets Modelle *von etwas* und können deshalb, wie bereits mehrfach erwähnt, nur im Kontext ihrer Anwendung evaluiert werden. Modellierungstechniken rein um ihrer selbst Willen zu unterrichten, halte ich daher nicht für sinnvoll, da eben diese Zweckgebundenheit damit vernachlässigt wür-

de. Für Petrinetze im Informatikunterricht gilt es daher zu prüfen, ob sie tatsächlich dazu beitragen, dass die Lernenden die jeweils modellierten Systeme verstehen. Werden die behandelten Systeme tatsächlich mithilfe von Petrinetzen diskutiert und verstanden? Oder bedienen sich Lernende und Lehrkraft am Ende ganz anderer Veranschaulichungen, versuchen also das „Hindernis Petrinetze“ zu umgehen?

3. **Ist der von mir entwickelte Petrinetz-Editor *Versys* für die schulische Praxis geeignet? Wo liegen mögliche Probleme beim Einsatz?** Die Erprobung bietet erstmals die Möglichkeit, *Versys* mit einer größeren Anwendergruppe zu testen. Sie dient daher auch dem Finden eventuell noch vorhandener Fehler im Programm. Abgesehen von der technischen Korrektheit muss sich *Versys* jedoch vor allem als didaktisches Werkzeug bewähren. Der Entwicklung lagen diesbezüglich verschiedene Überlegungen zugrunde. Zunächst einmal sollten natürlich alle in Abs. 3 genannten Modellierungstechniken unterstützt werden. Dies ist der Fall. Dennoch darf die Handhabung nicht unnötig kompliziert sein. Die Überlegungen ähnelten dabei denen zu didaktischen Programmierumgebungen. Die Lernenden sollen sich bei der Arbeit auf das Erstellen der eigentlichen Algorithmen bzw. Modelle konzentrieren können, ohne dabei von komplexen Menüs oder verwirrenden Einstellungen abgelenkt zu werden. Es gilt daher zu prüfen, ob *Versys* die Lernenden tatsächlich bei der Modellierung unterstützt, oder ob es am Ende nicht einfacher gewesen wäre, die Modelle per Hand zu zeichnen und „im Kopf“ zu simulieren.

4.2 Theorie und Methodik

Da im Rahmen dieser Arbeit nur zwei der Themenbausteine (und auch diese nur in einem sehr begrenzten Rahmen) erprobt werden können, muss zuächst eine

sinnvolle Auswahl getroffen werden. In Bezug auf das Spiralcurriculum halte ich es für sinnvoll, einen Baustein aus der Sekundarstufe I und einen aus der Sekundarstufe II zu erproben. Auf diese Weise können verschiedene Komplexitäts- und Anforderungsniveaus untersucht werden. In der Sekundarstufe I halte ich den Themenbaustein *Eingebettete Systeme* für besonders relevant. Er stellt den Beginn der Curriculumspirale dar, setzt also kein konkretes Vorwissen zu Petrinetzen voraus und enthält somit eine erste Einführung der Modellierungskonzepte. Eine solche Einführung ist im Rahmen der Untersuchtung meiner Meinung nach von besonderem Interesse. In der Sekundarstufe II bietet sich der Baustein *Nebenläufige Prozesse* an. Er ist der einzige, der sich sinnvoll auf 90 Minuten reduzieren lässt: weder eine Einführung in formale Sprachen, inklusive Exkurs zu Petrinetzen, noch ein Softwareprojekt können im Rahmen der verfügbaren Zeit umgesetzt werden. Ferner stellt er, analog zu den eingebetteten Systemen, den ersten Themenbaustein in der Sekundarstufe II dar und hat somit eine besondere Stellung in der Curriculumspirale. Nicht zuletzt ist Nebenläufigkeit – als ein Kernkonzept der Modellierung mit Petrinetzen – im Informatikunterricht bislang eher unzureichend methodisch aufgearbeitet. Das Thema bietet sich also an, um den methodischen Wert von Petrinetzen praktisch zu demonstrieren.

Der Unterricht findet an einem Berliner Gymnasium statt, an dem Informatik ab der 9. Jahrgangsstufe als Wahlpflichtkurs und ab der 11. Jahrgangsstufe als Grund- bzw. Leistungskurs angeboten wird. Der Unterricht zu eingebetteten Systemen findet in einer 9. Klasse im Wahlpflichunterricht statt. Informatik wird in diesem Kurs mit zwei Wochenstunden (als 90-Minuten-Block) unterrichtet. Für eine detailliertere Lerngruppenanalyse siehe Anhang A.1. Der Unterricht zu nebenläufigen Prozessen findet in einer 12. Klasse im Leistungskurs statt. Informatik wird in diesem Kurs mit fünf Wochenstunden unterrichtet. Für eine detailliertere Lerngruppenanalyse siehe Anhang B.1. Beide Kurse werden von derselben Infor-

matiklehrkraft unterrichtet.

Die Planung der Stunden sowie die Erarbeitung der entsprechenden Materialien erfolgt in Kooperation mit der Lehrkraft. Die vollständigen Unterrichtsentwürfe, Verlaufspläne und Arbeitsmaterialien sind ebenfalls in den Anhängen A.1 bzw. B.1 dokumentiert. Die Stunden werden auch von der regulären Lehrkraft gehalten. Dadurch soll vor allem der Eindruck einer Ausnahmesituation für die Lernenden verringert werden. Außerdem ist die Durchführung durch eine Person, die nicht an der theoretischen Konzeption des Curriculums beteiligt war, objektiver im Sinne der Erhebung.

Kuckartz et al. (2008) zufolge müssen die Methoden, die im Rahmen einer qualitativen Erhebung zum Einsatz kommen, vor allem „gegenstandsangemessen“ sein (S. 13). In diesem Fall handelt es sich bei dem Untersuchungsgegenstand um schulischen Informatikunterricht. Als theoretische Basis für die Untersuchungen soll daher das von Jank und Meyer (2002) entwickelte Strukturmodell des Unterrichts dienen (vgl. S. 63). Es teilt die Lehr-Lern-Situation in die fünf Dimensionen Inhalte, Ziele, Handlungen, Beziehungen und Zeit. Diese beeinflussen sich wechselseitig und können somit nicht vollständig voneinander abgegrenzt werden. Für die Validierung des vorgestellten Curriculums stehen dennoch die Inhalts- und Zieldimension im Vordergrund, jene Dimensionen, mit denen sich ein Curriculum vorrangig beschäftigt.

Für jede der beiden Erprobungen sollen drei verschiedene qualitative Verfahren eingesetzt und im Anschluss miteinander trianguliert werden. Die Triangulation dient Flick (2011) zufolge dabei nicht primär der gegenseitigen Validierung der einzelnen Methoden (vgl. S. 49). Die verschiedenen Ergebnisse sollen sich vielmehr gegenseitig ergänzen und so „ein breiteres, umfassenderes oder ggf. vollständigeres Bild des untersuchten Gegenstandes liefern“ (S. 49). Dazu sollen die eingesetzten Methoden verschiedene Ebenen des Untersuchtungsgegenstandes fo-

kussieren:

> What is important is to choose at least one method which is specifically suited to exploring the structural aspects of the problem and at least one which can capture the essential elements of its meaning to those involved. (Fielding und Fielding in Flick 2011, S. 41)

Das oben angeführten Strukturmodell von Jank und Meyer (2002) unterteilt jede der fünf Dimensionen noch einmal in eine innere und eine äußere Seite:

> Die äußere Seite erfasst das, was unmittelbar zu beobachten ist, also die stattfindenden Lehr-Lerntätigkeiten, die dabei verbalisierte Sprache, die zeitlichen Abstände zwischen einzelnen Unterrichtsschritten, die produzierten Ergebnisse usw. (S. 62)

Sie entspricht somit den im oberen Zitat angeführten *structural apects*. Diese äußerlich wahrnehmbaren Handlungen sind jedoch lediglich Ausdruck einer verborgenen, inneren Seite, die „diesen Tätigkeiten, Sprechakten, Arbeitsprozessen und -ergebnissen zugrunde liegt“ (ebd., S. 62). Erst diese innere Seite gibt den äußeren Handlungen eine Bedeutung. Jank und Meyer (ebd.) sprechen von den „‚Logiken‘ des Unterrichts“ (S. 62). Die innere Seite entspricht somit den *elements of meaning* im oberen Zitat. Entsprechend sollen die hier gewählten Erhebungsverfahren jeweils die innere oder die äußere Seite des Unterrichts fokussieren, um sich im Rahmen der anschließenden Triangulation entsprechend ergänzen zu können.

Zur Erfassung der äußeren, strukturellen Seite des Unterrichts wird eine teilnehmende Beobachtung durchgeführt und ein entsprechendes Beobachtungsprotokoll angefertigt. Ein direktes Eingreifen des Beobachters in das Unterrichtsgeschehen ist zwar nicht geplant, kann in Extremfällen aber nicht ausgeschlossen werden. Eine Interaktion mit den Lernenden ist streng genommen schon allein aufgrund der körperlichen Anwesenheit gegeben. Das Protokoll soll möglichst genau den zeitlichen Verlauf des Unterrichts, die verschiedenen Lernaktivitäten

sowie gegebenfalls „kritische Ereignisse“ festhalten. Letztere bezeichnen nach Brosziewski und Maeder (2007) „Übergänge, Vorfälle und Ereignisse [. . .], welche im Interaktionsgeschehen im Unterricht vom Beobachter (und den Beobachteten) als für das weitere Geschehen als [sic] wichtig eingeschätzt werden“ und somit „eine Veränderung des Interaktionsgeschehens“ markieren (S. 36).

Der Fokus der Beobachtung liegt auf der Seite der Lernenden. Ihre Handlungen und Äußerungen sind Ausdruck ihrer Auseinandersetzung mit den Unterrichtsinhalten und -methoden und somit der stattfindenen Lernprozesse. Diese gilt es im Rahmen der Untersuchung zu erschließen. Die Lehrkraft kann dabei natürlich nicht vollständig ausgeklammert werden. Ihr Handeln und ihre Kompetenzen sind jedoch nicht der zentrale Untersuchungsgegenstand.

Den zweiten Teil der Erhebung bildet ein leitfadengestütztes Interview, das im Anschluss an den Unterricht mit der Lehrkraft geführt wird. Zunächst einmal kann auch die Lehrkaft als teilnehmende Beobachtungsperson gesehen werden. Ihre Beobachtungen können die Erhebungen zur äußeren Seite des Unterrichts ergänzen. Das Interview soll aber vor allem eine erste Einschätzung der inneren Seite des Unterrichts, d.h. insbesondere der stattgefundenen Lernprozesse liefern. Die Annahme ist hierbei, dass die Lehrkraft die Schülerinnen und Schüler bereits zu einem gewissen Grad kennt und daher in der Lage ist, ihre Handlungen entsprechend zu interpretieren.

Daraus folgt unmittelbar ein Problem: da die Äußerungen der Lehrkraft im Rahmen der anschließenden Auswertung wiederum interpretiert werden müssen, handelt es sich dabei schlussendlich um eine Interpretation einer Interpretation. Die Sicht auf den eigentlichen Untersuchungsgegenstand ist also nicht nur durch eine, sondern durch zwei subjektive Wahrnehmungen verzerrt. Das Interview sollte daher nicht als alleinige Methode zur Erfassung der inneren Seite des Unterrichts herangezogen werden.

Wie oben erwähnt, ist das primäre Ziel des Interviews, eine Einschätzung der Stunde aus Sicht der Lehrkraft zu erhalten. Bei einer solchen Einschätzung ist davon auszugehen, dass sie vorrangig aus den Textsorten Beschreibung, Argumentation und Bewertung bestehen wird, während Erzählungen wohl eher eine untergeordnete Rolle spielen werden (vgl. Nohl 2012, S. 20-22). Sowohl Nohl (ebd.) als auch Witzel (2000) schreiben im Rahmen qualitativer Interviews jedoch genau diesen Erzählungen einen hohen Wert zu. Für Argumentationen und Bewertungen hält Nohl (2012) hingegen fest:

> Da Forschende und Erforschte meist unterschiedlichen Milieus angehören, müssen sich diese Plausibilisierungen und Stellungnahmen auf Wissensbestände beziehen, die über die Milieugrenzen hinweg gesellschaftlich geteilt werden. Ein solches gesellschaftlich geteiltes, kommunikatives Wissen ist notwendiger Weise abstrakt und damit von der Handlungspraxis abgehoben. (S. 43)

In diesem Fall liegen die Milieus von Forscher und Erforschtem jedoch eng beieinander. Ich, als angehende Informatiklehrkraft, befrage eine andere Informatiklehrkraft zu einer Unterrichtsstunde, die wir beide aus erster Hand miterlebt haben. Die ergänzende Funktion der Beobachtung wird hier bereits deutlich. Die Argumentationen und Bewertungen können somit hoffentlich sehr nah an der jeweiligen Handlungspraxis bleiben.

Das Interview wird mitgeschnitten und im Anschluss für die Auswertung transkribiert. Da das Forschungsinteresse hier nicht bei der interviewten Person selbst, sondern dem Inhalt des Gesprächs liegt, reicht ein vergleichsweise einfaches Transkriptionssystem, in dem Dialekt, Umgangssprache und Intonation weitgehend geglättet werden (vgl. Dresing und Pehl 2013, S. 17). Auf einen Kurzfragebogen z.B. zu Alter, Geschlecht, etc. (vgl. Kuckartz et al. 2008, S. 22-23) wird ebenfalls verzichtet. Da beide Interviews mit derselben Person geführt werden, ist eine demografische Differenzierung hinfällig. Der für die Interviews

verwendete Leitfaden orientiert sich an Witzel (2000) und ist in Anhang A.3.1 dokumentiert.

Das Interview sollte, wie bereits erwähnt, nicht als einzige Methode die innere Seite des Unterrichts fokussieren. Es wird daher durch eine kurze Lernerfolgskontrolle ergänzt, welche zu Beginn der nächsten Informatikstunde durchgeführt wird. Die Annahme ist, dass die Testergebnisse die Ergebnisse der Lernprozesse der Schülerinnen und Schüler wiederspiegeln. Der zeitliche Abstand zum Unterricht soll dabei verhindern, dass im Test lediglich Inhalte reproduziert werden, die den Lernenden noch frisch im Kurzzeitgedächtnis sind. Der Test wird zudem in beiden Kursen benotet, stellt für die Lernenden also eine „echte“ Testsituation dar.

Durch die Vergabe von Punkten für die jeweiligen Testantworten werden die Daten scheinbar quantifiziert. Da die Anzahlen der Lernenden jedoch weit unter der Schwelle einer statistischen Repräsentanz liegen, wird der Test nicht als standardisierter, empirischer Test, sondern nach didaktisch-pädagogischen Kriterien konstruiert. Die Aufgaben orientieren sich an den von der Kultusministerkonferenz (2004) vorgegebenen Anforderungsbreichen (vgl. S. 10-11). Diese sind zwar für die Abiturprüfungen konzipiert, aber allgemein genug, um auch in der Sekundarstufe I als Orientierung zu dienen. Der Test enthält somit auch Aufgaben, bei denen die Lernenden nicht nur Wissen reproduzieren, sondern auch anwenden sollen. Eine rein quantitative Auswertung der Testergebnisse ist auch allein deshalb nicht zielführend, weil mit dem Test keine summative Leistungsfestellung bezweckt wird. Vielmehr sollen im Rahmen der Auswertung die *Gründe* für die (höchstwahrscheinlich) auftretenden Fehler identifiziert und mit den jeweiligen Unterrichtsstunden in Beziehung gesetzt werden. Die Antworten müssen daher auch hier interpretiert werden. Die vergebenen Punkte können lediglich dazu dienen, eventuelle Problembereiche zu identifizieren, welche im Anschluss einer

genaueren qualitativen Analyse unterzogen werden müssen. Um die Punktvergabe möglichst transparent und objektiv zu gestalten, wird zuvor ein Erwartungshorizont angefertigt, welcher anschließend als Bewertungsgrundlage dient (siehe Anhänge A.4.1 und B.4.1).

Der Analyse- und Auswertungsprozess der erhobenen Daten orientiert sich an den von Kuckartz et al. (2008) vorgestellten Schritten (vgl. S. 33-57). Dazu werden die verschiedenen Beobachtungen, Interviewäußerungen und Testergebnisse zunächst entsprechend ihrer Inhalte kodiert. Die Kodierung erfolgt mithilfe der freien Software RQDA (Huang 2012). Die Basis für die Kodierung bildet ein vorab konzipiertes Kategoriensystem, welches zunächst „deduktiv [...] an das Material herangetragen“ wird (Bortz und Döring 2002, S. 330). Es orientiert sich an den in Abs. 4.1 vorgestellten Leitfragen und ist in Tabelle 6 dargestellt. Unter Umständen muss das Kategoriensystem jedoch im Laufe des Analyseprozesses induktiv erweitert werden (vgl. ebd., S. 330). Die verschiedenen Kategorien sind theoretisch zunächst klar voneinander zu trennen, können in der Praxis aber durchaus gemeinsam auftreten: z.B., wenn ein Schüler versucht, mithilfe einer *Versys*-Simulation die Schaltregel nachzuvollziehen. Es kann daher durchaus vorkommen, dass einem Datum mehrere Kategorien zugeordnet werden.

Das kodierte Material bildet die Grundlage für die anschließende Triangulation und Interpretation. Die verschiedenen Daten werden auf Basis der ihnen zugeordneten Kategorien fallübergreifend zusammengeführt, sodass sich, wie oben beschrieben, ein möglichst umfangreiches Bild zu der jeweiligen Kategorie bzw. Leitfrage ergibt. Ziel der Auswertung ist es, verschiedene Thesen zu generieren, mit deren Hilfe die aufgetretenen Effekte und Phänomene erklärt und theoretisiert werden können. Eine weiterführende empirische Überprüfung dieser Thesen kann im Rahmen dieser Arbeit jedoch nicht mehr stattfinden.

Oberkategorien	Unterkategorien
Petrinetz-Konzepte: die Syntax, d.h. der grundlegende Aufbau und die Funktionsweise von Petrinetzen, unabhängig davon, was sie darstellen	**Netzstruktur:** der grundlegende Aufbau eines Petrinetzes aus Plätzen, Transitionen und Kanten. **Markierung:** Marken auf Plätzen. **Schaltregel:** die Aktivierung und das Schalten von Transitionen
Modellierte Systeme: die Systeme und Konzepte, die (auch) mithilfe der Modelle dargestellt, veranschaulicht und verstanden werden sollen; beinhaltet insbesondere Modellkritik und -analyse	**Eingebettetes System:** Aufbau und Verhalten der betrachteten eingebetteten Systeme **Wechselseitiger Ausschluss:** zwei Prozesse greifen auf eine gemeinsame knappe Ressource zu **Speisende Philosophen:** der Aufbau und das Verhalten der speisenden Philosophen **Verklemmung:** ein System verklemmt, weil alle Prozesse auf etwas warten **Atomarität:** eine Operation kann nicht durch einen anderen Prozess unterbrochen werden **Fairness:** wenn ein Prozess etwas tun möchte, dann kommt er auch irgendwann an die Reihe
Versys: der Umgang mit dem Software-Werkzeug.	**Menüführung:** Operationen in *Versys*, die nicht primär dem Erstellen oder Simulieren von Modellen dienen (z.B. Dateien öffnen und speichern) **Modellierung:** das Erstellen von Petrinetzen mit *Versys* **Simulation:** das Schalten von Transitionen in *Versys* **Versys-Fehler:** *Versys* funktioniert nicht wie erwartet, stürzt ab, bleibt hängen, etc.

Tabelle 6: Deduktive Kategorien zur Kodierung der erhobenen Daten

4.3 Auswertung

Für den Unterricht im Wahlpflichtkurs ist im Folgenden zu beachten, dass ein erhebliches Zeitproblem auftrat. Bereits der Stundeneinstieg dauerte wesentlich länger als die geplanten fünf Minuten, „einfach aus dem Grund heraus, dass hier mehr als nur ein Einstieg passiert ist, nämlich hier ist eigentlich ein Stück weit eine Erarbeitungsphase dieses Begriffes (.) am Anfang gewesen“ (A.3.2, Z. 106-108). Für die abschließende Modellierungsphase blieben daher anstatt der geplanten 20 Minuten nur noch gut zehn. Die anschließende Auswertung fiel bereits vollständig in die Pausenzeit (vgl. A.2.1, Z. 78-87).

Der Unterricht im Leistungskurs verlief hingegegen vollständig nach Plan. Lediglich die Aufgabe zur Fairness wurde am Ende zugunsten der Atomarität nicht mehr bearbeitet. Es gab im Material daher keine Stellen, die der entsprechenden Kategorie (vgl. Tabelle 6) zugeordnet werden konnten. Das Konzept der Fairness wird in der folgenden Auswertung somit nicht weiter berücksicht.

4.3.1 Petrinetz-Konzepte

Die grundlegende Netzstruktur, d.h. die Syntax von Petrinetzen, spielte in beiden Blöcken eine eher untergeordnete Rolle. Im Leistungskurs diente lediglich die kurze Wiederholung am Anfang dazu, sich an den Aufbau und die Funktionsweise von Petrinetzen zu erinnern. Dabei war überraschend, wie schnell sich die meisten Lernenden scheinbar an das erinnerten, was sie vor über einem halben Jahr in der Kompaktwoche erarbeitet hatten (vgl. B.3.2, Z. 473-491). Die anschließende Analyse des Netzes zum wechselseitigen Auschluss stellte zwar eingangs eine gewisse Herausforderung dar, wurde jedoch von allen zufriedenstellend bewältigt (B.3.2, Z. 492-497). Die erste Testaufgabe (B.4, Aufg. 1) wurde ebenfalls von allen Lernenden hervorragend bewältigt, sodass davon auszugehen ist, dass sie die

Begriffe Platz, Transition und Kante souverän beherrschen. Auch im Wahlpflichtkurs wurde die Syntax der Netzstruktur nur im Rahmen der allerersten Aufgabe erarbeitet (vgl. A.1, AB1) und danach, ähnlich wie im Leistungskurs, nicht weiter explizit thematisiert.

Petrinetze wurden in beiden Kursen danach nur noch unter semantischen Gesichtspunkten betrachtet. Sie wurden analysiert und modelliert, wobei selbst die Lernenden im Wahlpflichtkurs sehr schnell begannen, im Unterrichtsgespräch die korrekte Terminologie zu verwenden (A.3.2, Z. 292-298). Die korrekte Syntax der Netzstruktur war jedoch stets vorgegeben: entweder durch das zu analysierende Netz selbst oder durch das Programm Versys, welches keine syntaktisch falschen Netze erlaubt. Diese Vernachlässigung der Syntax hat meiner Meinung nach zu dem schlechten Ergebnis der ersten Testaufgabe im Wahlpflichtkurs geführt, bei der die Lernenden Fehler in der Netzstruktur erkennen sollten (vgl. A.4, Aufg. 1). Sie hatten jedoch, aufgrund der oben genannten Umstände, im Unterricht zuvor nie die Möglichkeit, solche Fehler zu sehen oder gar selbst zu begehen. Sie waren, mit Ausnahme der ersten Erarbeitung, nie gezwungen, sich bewusst mit der Syntax der Netzstruktur auseinander zusetzen. Dennoch waren sie in der Lage, Petrinetze zu analysieren und mithilfe von Versys zu modellieren (vgl. Abs. 4.3.2).

These 1. *Petrinetze können im Unterricht erfolgreich eingesetzt werden, ohne dass sich die Lernenden zunächst intensiv mit ihrer Syntax auseinander setzen müssen.*

Im Gegensatz zur Syntax der Netzstruktur spielte die Bedeutung von Marken auf Plätzen im Wahlpflichtkurs eine verhältnismäßig große Rolle. In den Aufgaben 2 und 3 zum Schließsystem (vgl. A.1, AB1) wurden sie als zentrale Bedingung für das Aktivieren und Schalten von Transitionen erarbeitet (vgl. A.2.1, Z. 33-46). Auch bei der Modellierung des Licht/Lüfter-Systems schienen die Marken

für die Lernenden eine zentrale Stellung zu haben. Sie erkannten die Fehler in dem am Ende vorgestellten Modell scheinbar daran, dass „Marken ‚verloren gehen'" (A.2.1, Z. 80). Die Testaufgabe 2 zur Schaltregel wurde entsprechend von vielen Schülerinnen und Schülern gut oder sehr gut bewältigt (vgl. A.4, Aufg. 2). Interessant ist zudem, dass viele Lernende bei der ersten Testaufgabe leere Plätze oder korrekte Marken fälschlicherweise als Fehler identifiziert haben. In einem Fall wurde sogar eine Marke auf einem Platz durchgestrichen und auf einem anderen Platz eine eingezeichnet (vgl. A.4, Aufg. 1, Sara). Dies legt die Vermutung nahe, dass hier weniger über die Syntax, d.h. die Netzstruktur, und mehr über korrekte Markierung, d.h. das Verhalten des Netzes, nachgedacht wurde.

These 2. *Für die Lernenden im Wahlpflichkurs stellten die beobachteten Marken und ihre Bewegungen das zentrale Merkmal dar, anhand dessen die Korrektheit eines Petrinetz-Modells bewertet wurde.*

Im Gegensatz dazu wurden Marken im Leistungskurs, mit Ausnahme der kurzen Wiederholung am Anfang, fast gar nicht explizit thematisiert. In der anschließenden Aufgabe zum wechselseitigen Ausschluss waren die meisten Lernenden scheinbar in der Lage, die Marken „im Kopf" durch das Netz wandern zu lassen (vgl. B.3.2, Z. 639-645). Zudem verwendeten alle von den Lernenden erstellten Modelle am Ende sogar farbige Marken. Dies mag in einigen Fällen zwar durch den expliziten Hinweis auf dem Arbeitsblatt motiviert worden sein (vgl. B.1, AB3), schien aber auch keine nennenswerte Herausforderung darzustellen. Die Lehrkraft musste lediglich an einer Stelle kurz darauf hinweisen, dass die farbigen Marken auch entsprechende Kantengewichte verlangen, aber „[d]ie Information bei den beiden hat ja gereicht" (B.3.2, Z. 707). Überraschend ist vor diesem Hintergrund jedoch, dass viele Leistungskursschüler bei den Modellierungsaufgaben im Test vergessen haben, entsprechende Marken auf die Plätze zu verteilen (vgl. B.4, Aufg. 2 u. 4). Der Grund dafür könnte einfache Schusseligkeit sein. Es ist möglich, dass die

Lernenden beim Modellieren auf Papier die Marken, wie bei der Aufgabe zum wechselseitigen Ausschluss, zunächst „im Kopf“ simuliert haben und am Ende schlicht vergessen haben, sie noch einzuzeichnen.

These 3. *Für die Lernenden im Leistungskurs stellte die statische Netzstruktur das zentrale Element dar, anhand dessen die Korrektheit eines Petrinetz-Modells bewertet wurde. Markenbewegungen simulierten sie gegebenfalls im Kopf.*

Die Thesen 2 und 3 können auch lerntheoretisch durch das Repräsentationsmodell nach Bruner (vgl. Humbert 2006, S. 37) weiter gestützt werden. Den fortgeschrittenen Lernenden im Leistungskurs fällt es leichter, abstrakte Symbole zu analysieren und sich gegebenfalls auch für statische Netzstrukturen ein dynamisches Verhalten vorzustellen. Im Gegensatz dazu sind die jüngeren Lernenden im Wahlpflichkurs noch stärker auf eine ikonische Darstellung angewiesen. Sie müssen in vielen Fällen die Dynamik der Modelle direkt sehen, um sie analysieren zu können und legen somit einen stärkeren Fokus auf die Marken eines Netzes.

Zusammenfassend lässt sich für die erste Leitfrage somit festhalten, dass der Aufbau und die Funktionsweise der eingesetzten Petrinetzmodelle scheinbar in keinem der beiden Kurse zu einer Überforderung geführt hat. Im Leistungskurs verwendeten die Lernenden souverän farbige Marken und entsprechende Kantengewichte. Die Terminologie zu Petrinetzen beherrschen sie scheinbar sicher. Auch im Wahlpflichtkurs haben die Lernenden sehr schnell damit begonnen, die korrekten Begrifflichkeiten im Unterrichtsgespräch zu verwenden. Die Bedeutung der Marken sowie das Aktivieren und Schalten von Transitionen scheinen sie erfasst zu haben. Lediglich die Syntax der Netzstruktur bereitete Schwierigkeiten, was vermutlich mit dem Einsatz von *Versys* zusammenhängt (vgl. Abs. 4.3.3).

4.3.2 Modellierte Systeme

Die Erarbeitung des Begriffs „eingebettetes System“ geschah im Wahlpflichtkurs zunächst völllig unabhängig von Petrinetzen mithilfe von Beispielbildern. Erst im Rahmen von Aufgabe 4 sollten die Lernenden versuchen, das modellierte Schließsystem mithilfe des Petrinetzes zu analysieren und dabei auch das Modell selbst einer Kritik zu unterziehen (vgl. A.1, AB1). Bei der Auswertung der Aufgabe entfernte sich die Diskussion jedoch immer wieder vom gegebenen Modell. Einige Lernende versuchten, die gestellten Fragen mithilfe ihres Alltagswissens, nicht mithilfe des Modells, zu beantworten (vgl. A.2.1, Z. 49-52). Dasselbe Phänomen trat bei der Testaufgabe 3b auf. Auch hier haben einige Lernende mithilfe ihres Alltagswissens über Alarmanlagen, nicht mithilfe des gegebenen Petrinetzes argumentiert (vgl. A.4, Aufg. 3, WPS2).

Die Lernenden hatten scheinbar Probleme damit, das reale System und das Modell voneinander abzugrenzen. Ich vermute, ihnen fehlte an dieser Stelle ein allgemeiner Modellbegriff, d.h. eine Vorstellung davon, was ein Modell eigentlich ist und nach welchen Kriterien es bewertet und analysiert werden kann. Vermutlich haben einige (leistungsstarke) Lernende währen der Auswertung von Aufgabe 4 (A.1, AB1) solch eine Vorstellung zumindest intuitiv entwickelt. Prinzipiell kann ein allgemeiner Modellbegriff meiner Meinung nach auch aus solch einem konkreten Beispiel heraus entwickelt werden. Aufgabe 4 ist in ihrer jetzigen Form jedoch nicht dafür ausgelegt.

Auch das anschließende Ausfüllen der Tabelle und das Identifizieren der Transitionen für das Petrinetz (A.1, AB2) bereitete Schwierigkeiten und stellte für einige Lernende vermutlich eine Überforderung dar (vgl. A.3.2, Z. 237-261). Umso erstaunlicher ist es, dass einige im Anschluss ein fast korrektes Modell des Systems erstellt haben, und das trotz der sehr knappen Zeit (vgl. A.2.2, Julian).

These 4. *Diejenigen Lernenden im Wahlpflichtkurs, die im Verlauf des Unterrichts eine zumindest intuitive Vorstellung eines Modellbegriffs entwickelt haben, waren am Ende in der Lage, die Modellierungsaufgabe und die im Test geforderte Analyse (A.4, Aufg. 3b) zu bewältigen.*

Das am Ende von einem Schüler vorgestellte Modell des Licht/Lüfter-Systems (A.2.2, Julian) wurde von ihm selbst und den anderen Lernenden als unvollständig beschrieben: „Das funktioniert nur für einen Ablauf“ (A.2.1, Z. 79). Die Lernenden haben erkannt, dass sich das Verhalten des Modells noch von dem des realen Systems unterscheidet. Im Plenum wurden dann sehr schnell Lösungsansätze gefunden, die in die richtige Richtung gingen: „Wenn man den Lüfter anschaltet, dann geht eine Marke zurück auf ‚Licht an‘“ (A.2.1, Z. 81-82). Die Testaufgabe 3a stellte eine ähnliche Anforderung dar. Auch hier musste das beschriebene Verhalten eines Systems mit einem gegeben Petrinetz-Modell verglichen und die entsprechenden Elemente darin identifiziert werden. Diese Aufgabe haben fast alle Schülerinnen und Schüler gut oder sehr gut gelöst (vgl. A.4, Aufg. 3a).

Ich denke, dass die Lernenden diese Art der Analyse besser bewältigen konnten, weil hier explizit von ihnen gefordert war, das reale System und das Modell miteinander zu identifizieren. Die Modelle waren hier nichts weiter als grafische Veranschaulichungen der entsprechenden Systeme.

These 5. *Ein Petrinetz-Modell als Veranschaulichung eines Systems zu verwenden erfordert keinen allgemeinen Modellbegriff, da die Lernenden System und Modell direkt miteinander identifizieren können.*

Im Leistungskurs war das zentrale Thema der wechselseitige Ausschluss. Nach der anfänglichen Wiederholungsübung waren offenbar die meisten Lernenden in der Lage, das Petrinetz zum wechselseitigen Ausschluss zu analysieren, die entsprechenden Zustände zu identifizieren und auch die wechselseitige Verwendung

der knappen Ressource zu beschreiben (vgl. B.3.2, Z. 492-499).

Als sich die Diskussion jedoch dann vom konkreten Petrinetz löste und der wechselseitige Ausschluss als allgemeines Prinzip beschrieben und in anderen Anwendungen identifiziert werden sollte, traten einige Schwierigkeiten auf. Ein Schüler begann, das Konzept mithilfe von Methoden und Funktionen in der Programmierung zu beschreiben (vgl. B.2.1, Z. 401 u. 410-412), wobei es dann Schwierigkeiten gab, diese Sichtweise mit dem Petrinetz-Modell in Einklang zu bringen (vgl. B.3.2, Z. 802-806). Nur ein Schüler konnte noch weitere Beispiele für einen wechselseitigen Auschluss im Alltag nennen und Beispiele aus der Technik konnten nur mithilfe der Lehrkraft erarbeitet werden (vgl. B.2.1, Z. 405-414), sodass an dieser Stelle vermutlich eine leichte Überforderung auftrat. Dennoch haben die meisten Lernenden in der zweiten Testaufgabe zur Baustelle die korrekte Netzstruktur für den wechselseitigen Ausschluss ergänzt und die gemeinsame Ressource identifiziert (vgl. B.4, Aufg. 2).

These 6. *Das elementare Petrinetz-Modell zum wechselseitigen Ausschluss eignet sich gut, um das allgemeine Konzept der knappen Ressource zu erarbeiten. Für eine sofortige eigenständige Übertragung auf andere Anwendungskontexte, insbesondere im Rahmen von Computer und Technik, ist es jedoch selbst für Leistungskursschüler zunächst zu abstrakt.*

Damit könnte auch das eher schlechte Ergebnis der vierten Testaufgabe zur Fußgängerampel erklärt werden. Hier gab es keine offensichtliche knappe Ressource, sodass das elementare Netz zum wechselseitigen Ausschluss nicht unmittelbar angewandt werden konnte. Die Hälfte der Lernenden hat es dennoch versucht und entweder die Ampel oder die Straße fälschlicherweise als knappe Ressource identifiziert. Nur zwei Lernende haben eine korrekte Ampelschaltung modelliert (vgl. B.4, Aufg. 4).

Die Modellierung und Veranschaulichung der speisenden Philosophen und des Problems der Verklemmung verlief hingegen erstaunlich gut. Die Aufgabenstellung zu den speisenden Philosophen musste mit den Lernenden zwar zunächst ausführlicher besprochen werden als geplant:

> Es gab ja Nachfragen, wie man jetzt das wirklich genau interpretieren soll. [...] Das mag zwar eindeutig darin stehen, aber es ist ja immer die zweite Sache, es zu glauben, dass es wirklich so hundertprozentig gemeint ist, wie es da steht. (B.3.2, Z. 527-531)

Die anschließende Modellierung der Philosophen wurde meiner Meinung nach von den Lernenden aber hervorragend bewältigt. Alle hatten am Ende ein Modell erstellt, das zumindest das Aufnehmen der Essstäbchen für einen der vier Philosophen korrekt modellierte (vgl. B.2.2 u. B.2.1, Z. 431-436). Die Tatsache, dass die Modelle unvollständig waren, wurde von den Lernenden dabei scheinbar auch nicht als Misserfolg oder Problem wahrgenommen: „Sie haben ganz klar gesagt: ‚Ja, ich habe nur den einen Philosophen hingekriegt, aber das muss man jetzt ja bloß fortsetzen,‘ und das war vollkommen in Ordnung“ (B.3.2, Z. 680-682). Die unterschiedlichen Modellierungsansätze führten am Ende noch zu einer kurzen Diskussion unter den Lerndenen, welche Zustände, d.h. welche Plätze, in dem Petrinetz-Modell denn nun vorkommen müssten (B.2.1, Z. 437-438). Diese Modellkritik war an dieser Stelle nicht geplant und zeigt meiner Meinung nach, dass die Lernenden tatsächlich versucht haben, das System der speisenden Philosophen mithilfe der Petrinetz-Modelle zu verstehen und zu analysieren.

These 7. *Die von ihnen erstellten Petrinetz-Modelle waren das primäre Mittel, mit dem die Lernenden das System der speisenden Philosophen analysiert und verstanden haben.*

Dasselbe gilt meiner Meinung nach für das Problem der Verklemmung. Die Lernenden fingen nach der Besprechung der Aufgabenstellung direkt damit an, das

Verhalten der Philosophen zu analysieren, waren jedoch zu diesem Zeitpunkt offenbar noch nicht in der Lage, das Problem der Verklemmung zu erfassen und zu formulieren (vgl. B.2.1, Z. 415-417). Im Laufe der Modellierungsphase wurden die geäußerten Beschreibungen und Analysen dann zunehmend präziser (vgl. B.2.1, Z. 418-420). In der abschließenden Besprechung im Plenum wurde die Verklemmung dann auch explizit im Verhalten des Petrinetzes identifiziert (B.2.1, Z. 447-449). „Das zeigt wiederum, dass dieses Hilfsmittel des Petrinetzes, dieses Modellieren und diese Auseinandersetzung damit ihnen nachher die Worte [zur Beschreibung des Problems] gegeben hat“ (B.3.2, Z. 539-540). In der dritten Testaufgabe zur Straßenkreuzung haben dann auch alle Lernenden das Problem der Verklemmung erkannt und korrekt beschrieben. Die meisten haben auch im entsprechenden Modell der Straßenkreuzung zumindest die korrekten Abhängigkeiten in eine Richtung ergänzt (vgl. B.4, Aufg. 3).

These 8. *Die Modellierung der speisenden Philosophen als Petrinetz hat den Lernenden geholfen, das Problem der Verklemmung zu erfassen und ihnen ein Mittel an die Hand gegeben, es präzise zu formulieren.*

Die Petrinetz-Modelle eigneten sich scheinbar auch sehr gut dazu, das Prinzip der Atomarität zu verdeutlichen. Die Lernenden haben es zunächst selbstständig in der Modellierung angewandt, ohne es explizit zu benennen: „beide gleichzeitig aufnehmen“ (B.2.1, Z. 441-442). In der anschließenden Auswertung wurde das Prinzip der Atomarität dann explizit von der Lehrkraft genannt, von den Lernenden wiedererkannt (vgl. B.2.1, Z. 454) und somit in einen größeren Kontext eingebettet (vgl. B.3.2, Z. 719-720).

Zusammenfassend lässt sich für die zweite Leitfrage somit festhalten, dass die verwendeten Petrinetz-Modelle in beiden Kursen zumindest als hilfreiche grafische Veranschaulichungen der jeweils modellierten Systeme dienten. Im Leistungskurs waren die Modelle scheinbar das primäre Mittel, mit dem die Lerneden die jewei-

ligen Systeme verstanden und analysiert haben. Sie unterzogen die betrachteten Modelle am Schluss sogar völlig selbstständig einer Modellkritik. Auch im Wahlpflichkurs haben die Lernenden die Modelle zumindest als hilfreiche Veranschaulichungen der jeweiligen Systeme wahrgenommen. Sie waren in der Lage, Elemente des realen Systems im Modell zu verorten und zu erkennen, wann das Verhalten des Modells nicht dem des Systems entspricht. Eine eigenständige Modellierung und Modellkritik bereitete jedoch noch Probleme. Hier fehlte vermutlich die Erarbeitung eines allgemeinen Modellbegriffs.

4.3.3 Versys

Der Einsatz von *Versys* verlief aus technischer Sicht zunächst weitgehend problemlos. Es gab nur vereinzelte Situationen, in denen das Programm nicht so funktionierte wie erwartet. Im Leistungskurs gab es Probleme mit der Farbdarstellung der Marken (vgl. B.2.1, Z. 427-429) und dem Speichern und Laden von Dateien mit farbigen Marken (vgl. B.2.1, Z. 452-453). Im Wahlpflichtkurs blieb das Programm, vermutlich aufgrund zu schnell gemachter Eingaben, für einen Moment stecken, lief dann jedoch problemlos weiter (vgl. B.3.2, Z. 274-279). Keiner dieser Fehler hat den Unterrichtsverlauf signifikant gestört oder gar unterbrochen. Leider ließen sich die Fehler im Nachhinein nicht reproduzieren, sodass nicht ausgeschlossen werden kann, dass sie auch in der aktuellen Version (1.0) auftreten.

Auch die Menüführung von *Versys* stellte in beiden Kursen scheinbar kein Problem dar. Weder mir noch der Lehrkraft sind Situationen aufgefallen, in denen sich ein Schüler oder eine Schülerin auffällig lang im Menü des Programms aufgehalten hätte, weil eine gesuchte Funktion o.Ä. nicht auffindbar war.

Bei der Modellierung mit Versys ist mir lediglich eine Situation aufgefallen, in der eine Schülerin scheinbar von der Bedienung des Proramms verwirrt war. Sie

versuchte, von einem Platz aus eine Kante „ins Leere“ zu erzeugen, d.h. sie nicht an einer Transition, sondern auf leerer Fläche enden zu lassen. *Versys* erlaubt jedoch zu keiner Zeit, dass sich ein syntaktisch falsches Netz auf der Arbeitsfläche befindet. Die Schülerin klickte daraufhin scheinbar ziellos im Programm umher (vgl. A.2.1, Z. 19-20), offenbar in dem Versuch, die Operation abzubrechen. In der aktuellen Version des Programms lassen sich solche Operationen nun intuitiver, durch einen Rechtsklick, abbrechen. Abgesehen davon schien das Erstellen von Petrinetzen in *Versys* für die Lernenden sehr intuitiv zu sein und erforderte auch keine weitere Unterstützung durch die Lehrkraft (vgl. A.3.2, Z. 131-132).

These 9. *Die Bedienung von Versys war für die Lernenden sehr intuitiv und bedurfte deshalb kaum weiterer Erklärungen.*

Es war sogar möglich, dass sich die Lernenden im Wahlpflichtkurs die Syntax von Petrinetzen selbstständig mit Hilfe des Programms erarbeiteten. Sie erkannten, nach welchen Regeln Kanten erzeugt werden können (vgl. A.2.1, Z. 21) und schrieben die Namen der Elemente (Platz, Transition, Kante, Marke) aus der Programmoberfläche ab (vgl. A.2.1, Z. 22-23 u. A.3.2, Z. 292-294). Eine lehrerzentrierte, frontale Einführung war somit nicht nötig.

These 10. *Das Modellieren mit Versys ermöglicht es, dass sich die Lernenden im Unterricht die Syntax von Petrinetzen eigenständig erarbeiten.*

Die Tatsache, dass *Versys* bei der Modellierung keine syntaktischen Fehler erlaubt, hatte jedoch zur Folge, dass sich die Lernendne im weiteren Verlauf des Unterrichts nicht mehr explizit mit der Syntax von Petrinetzen auseinander setzen mussten. Das schlechte Ergebnis der ersten Testaufgabe spiegelt dies wieder (vgl. A.4, Aufg. 1 u. Abs. 4.3.1).

These 11. *Das Modellieren mit Versys trägt nicht automatisch dazu bei, dass die Lernenden ein explizites Syntaxwissen entwickeln.*

Die Thesen 10 und 11 wirken konträr, widersprechen sich aber nicht. Die Syntax von Petrinetzen wird von *Versys* implizit vorgegeben, indem es schlicht keine syntaktisch falschen Netze erlaubt. Sollen sich die Lernenden bei der Arbeit mit *Versys* aber bewusst mit der Syntax auseinander setzen, muss eine entsprechende Aufgabenstellung dies explizit von ihnen verlangen.

Bei der Schaltregel verhält es sich da offenbar etwas anders. Auch diese wurde im Wahlpflichtkurs, ähnlich wie die Netzstruktur, von den Lernenden eigenständig mithilfe der Simulation in *Versys* erarbeitet. Sie haben im Programm beaobachtet, wann verschiedene Transitionen des Schließsystems aktiviert sind und was sich verändert, wenn sie schalten (vgl. A.2.1, Z. 34-38). Die anschließende Formulierung einer allgemeinen Regel brauchte zwar noch etwas Unterstützung durch die Lehrkraft, wurde im Allgemeinen aber gut bewältigt (vgl. A.2.1, Z. 42-48), ebenso wie die Testaufgabe 2 zur Schaltregel (vgl. A.4, Aufg. 2).

These 12. *Die Simulationsfunktion in Versys ermöglicht es, dass sich die Lernenden die Schaltregel für Transitionen eigenständig erarbeiten.*

These 12 ist analog zu These 10 über die eigenständige Erarbeitung der Netzstruktur. Ein zentraler Unterschied besteht jedoch darin, dass die Lernenden, im Gegensatz zur Syntax der Netzstruktur, im Nachhinein scheinbar in der Lage waren, die Schaltregel auch selbstständig anzuwenden (vgl. Abs. 4.3.1).

Ein Grund dafür könnte in der bewussten Verwendung der Simulation liegen. Die Lernenden haben die Simulationsfunktion auch während der Modellierungsphasen ausgiebig eingesetzt und die Effekte der schaltenden Transitionen dabei offenbar bewusst beobachtet. Ein Schüler des Wahlpflichkurses versuchte sogar mithilfe der Simulation, eine Markierung experimentell zu erarbeiten (vgl. A.2.1, Z. 73-77). Auch im Leistungskurs nutzten die Lernenden die Simulation, um zu testen, ob sich ihr Modell so verhielt, wie sie es erwarteten (vgl. B.2.1, Z. 450-452).

Abweichungen vom erwarteten Verhalten wurden dabei bewusst wahrgenommen und das Modell anschließend modifiziert.

These 13. *Die Simulationsfunktion in Versys unterstützt die Lernenden bei der Modellierung, indem sie ihnen erlaubt, ihre Modelle selbstständig zu testen und ggf. auch experimentell zu modifizieren. Das Aktivieren und Schalten von Transitionen wird dabei bewusst beobachtet.*

Zusammenfassend lässt sich für die dritte Leitfrage somit festhalten, dass *Versys* für die Lernenden offenabr durchaus ein hilfreiches Werkzeug bei der Modellierung und Analyse von Petrinetzen darstellte. Die Bedienung erforderte nahezu keine weitere Erläuterung. Es war sogar möglich, dass sich die Lernenden verschiedene Petrinetz-Konzepte mithilfe des Programms selbstständig erarbeiteten, sodass *Versys* an dieser Stelle einen Beitrag zu einer höheren Lernerorientierung im Unterricht leistete. Ein Verständnis der Syntax von Petrinetzen wird durch das Programm aber offenbar nicht gefördert.

4.4 Diskussion und Kritik

In diesem Abschnitt sollte das in Abs. 3 vorgestellte Spiralcurriculum in Ansätzen validiert werden. Dazu wurden die Themenbausteine *Eingebettete Systeme* und *Nebenläufige Prozesse* exemplarisch durch je einen Unterrichtsblock á 90 Minuten erprobt und qualitativ evaluiert. Zu jedem Block wurde eine teilnehmende Beobachtung, ein leitfadegestütztes Interview mit der Lehrkraft sowie ein abschließender Test mit den Lernenden durchgeführt. Die erhobenen Daten wurden, orientiert an den Leitfragen, kodiert und anschließend fallübergreifend miteinander trianguliert und ausgewertet. Es wurden insgesamt 13 Thesen formuliert, in einem Versuch, die aufgetretenen Phänomene und Effekte zu erklären. Eine abschließende empirische Überprüfung dieser Thesen steht noch aus. Für die

folgende Diskussion wird jedoch zunächst angenommen, dass sie zutreffend sind.

Für den Block zu eingebetteten Systemen im Wahlpflichtkurs ließe sich damit festhalten, dass dort vor allem die Erarbeitung eines allgemeinen Modellbegriffs gefehlt hat. Die Probleme, die bei der Modellkritik und dem Ausfüllen der Tabelle auftraten, hingen vermutlich damit zusammen, dass die Lernenden keine explizite Vorstellung davon hatten, wie und warum sich ein Modell von dem jeweils modellierten System unterscheidet (vgl. These 4). Eine solche Erarbeitung in den Block einzufügen würde natürlich noch einmal extra Zeit beanspruchen, sodass bei einer entsprechend überarbeiteten Einheit wohl vollständig auf die Modellierungsaufgabe am Ende verzichtet werden müsste (vgl. A.3.2, Z. 344-347). Dies ist hinsichtlich der in Abs. 3.2.1 geforderten Handlungs- und Produktorientierung bedauerlich. Im Anschluss darf jedoch vermutet werden, dass die Lernenden eher in der Lage sind, entsprechende Modellierungsaufgaben eigenständig zu bewältigen.

Nichtsdestotrotz ist deutlich geworden, dass Petrinetze auch bereits in der Sekundarstufe I erfolgreich eingesetzt werden können, um den Aufbau und das Verhalten von interkativen eingebetteten Systemen zu thematisieren und zu veranschaulichen (vgl. Thesen 1 und 5). Dabei sollte zumindest am Anfang noch darauf geachtet werden, dass die Modelle einen möglichst ikonischen Charakter haben, d.h. das jeweilige System nicht zu stark abstrahieren, sondern möglichst grafisch und anschaulich repräsentieren. Dies ist meiner Meinung nach zumindest für die in den Abb. 6 und 7 gezeigten (und im Unterricht verwendeten) Petrinetz-Modelle des Schließsystems und des Licht/Lüfter-Systems hinreichend gegeben. In beiden Modellen lassen sich die realen Komponenten des System und ihre jeweiligen Aktionen noch gut indentifizieren. Das Modell der Ampelsteuerung von Rotering (2010) ist bereits etwas abstrakter (vgl. S. 22-23), sodass es meiner Meinung nach nicht als erstes Einführungsbeispiel verwendet werden sollte. Der grundlegende Ansatz des in Abs. 3.2.1 vorgestellten Themenbausteins *Ein-*

gebettete Systeme kann vor dem Hintergrund der hier vorgestellten Erprobung aber durchaus als praktikabel angesehen werden. Es bleibt jedoch zu prüfen, ob die vorherige Modellierung mit Petrinetzen auch tatsächlich eine Hilfe für das anschließende Implementieren eines echten interaktiven Systems darstellt.

Der Block zum wechselseitigen Auschluss und den speisenden Philosophen im Leistungskurs hat gezeigt, dass sich Petrinetze sehr gut zur Erarbeitung, Veranschaulichung und Analyse nebenläufiger Prozesse eignen (vgl. Thesen 6, 7 und 8). Alle Lernenden haben zumindest einen korrekten Ansatz zur Modellierung gefunden und das Problem der Verklemmung mithilfe der Petrinetz-Modelle erfolgreich identifiziert und analysiert. Petrinetze können somit wohl tatsächlich, wie in Abs. 3.2.3 beschrieben, einen wertvollen Beitrag zur Methodik des Informatikunterrichts leisten, indem sie es erlauben, nebenläufige Prozesse grafisch und interaktiv zu veranschaulichen, ohne dabei auf die komplexe Umsetzung in einer Programmiersprache angewiesen zu sein.

Für *Versys* lässt sich, wie oben bereits erwähnt, festhalten, dass seine Bedienung für die Lernenden scheinbar sehr intuitiv war (vgl. These 9) und sein Einsatz zu einer höheren Lernerorientierung im Unterricht beitragen konnte (vgl. Thesen 10 und 12). Eine vertiefte Auseinandersetzung mit der Netzstruktur wurde, im Gegensatz zur Schaltregel, durch das Programm aber offenbar nicht gefördert (vgl. Thesen 11 und 13). Dies muss nicht unbedingt als Nachteil gewertet werden. Wie bereits in Abs. 3.1 dargestellt, herrscht auch in aktuellen Diskussionen zu didaktischen Programmierumgebungen zunehmend ein Konsens, dass die Syntax bei der Programmierung eine eher untergeordnete Rolle spielen sollte. Die Lernenden sollen nicht eine konkrete Programmiersprache lernen, sondern die darunterliegenden allgemeinen algorithmischen Konzepte. Analog ließe sich für Petrinetze argumentieren, dass die Lernenden nicht primär die Funktionsweise von Petrinetzen verstehen sollen, sondern die allgemeineren, von ihnen modellier-

ten Konzepte wie Interaktion, Kommunikation, Nebenläufigkeit oder Kausalität. Dadurch, dass *Versys* die Lernenden bei der Modellierung weitgehend von syntaktischen Problemen entlastet, können sie sich eher auf die Semantik des jeweiligen Modells konzentrieren, d.h. auf das *etwas*, das das Modell darstellen soll. *Versys* unterstützt somit die mehrfach in dieser Arbeit geforderte Zweckorientierung informatischer Modellierung.

All diese Schlüsse können, wie bereits erwähnt, nur unter der Annahme gezogen werden, dass die in Abs. 4.3 formulierten Thesen und Interpretationen auch valide sind. Einige der Thesen können durch weitere Argumente gestützt werden. Für die Thesen 2 und 3 wurde bereits ein weiteres lerntheoretisches Argument vorgestellt. Auch der erste Teil von These 5 ist zunächst intuitiv plausibel: grafische Veranschaulichen wurden, insbesondere in Lehr-Lernkontexten, quasi schon immer eingesetzt, ohne dass dafür jedes Mal erst ein allgemeiner Modellbegriff mit den Lernenden erarbeitet werden musste. Ob der Grund dafür jedoch wirklich die in These 5 genannte unmittelbare Identifikation von Modell und System ist, bleibt zu prüfen. Für eine abschließende Validierung der Thesen müssten daher weitere Untersuchungen durchgeführt werden.

Eine höhere Validität könnte jedoch schon erreicht werden, indem die erhobenen Daten noch ein zweites Mal von einer unabhängigen Person ausgewertet und interpretiert, und die Ergebnisse mit den hier vorgestellten verglichen würden. Der Grund dafür ist die wohl starke subjektive Färbung der Untersuchung. Sie wurde fast ausschleßlich von einer Person geplant, durchgeführt und ausgewertet. Es muss wohl auch nicht groß erläutert werden, dass ich durchaus ein persönliches Interesse an einem positiven Ergebnis der Untersuchung hatte. Zwar war an der Planung und Durchführung der Unterrichtsstunden noch eine weitere Person (die Lehrkraft) beteiligt, die dadurch gewonnene Objektivität kann jedoch als eher gering eingeschätzt werden. Die Lehrkraft war bereits vor über einem halben

Jahr an der Planung der Kompaktwoche (s. Anhang B.1) beteiligt und hat sich seitdem sehr für das Thema meiner Arbeit interessiert. Es ist somit davon auszugehen, dass sie ebenfalls ein gewisses Interesse daran hatte, die Untersuchung zu einem positiven Abschluss zu führen. Eine weitere Auswertung der Daten durch eine unabhängige Person könnte diese subjektive Färbung zumindest in Teilen kompensieren.

Eine Validierung der Interpretationen impliziert jedoch noch nicht ihre Generalisierbarkeit. Die meisten der in Abs. 4.3 formulierten Thesen beziehen sich zunächst nur auf den hier durchgeführten Unterricht und die damit untersuchten konkreten Lerngruppen. Eine Übertragung der Ergebnisse auf andere Lerngruppen würde ebenfalls weitere, vorrangig quantitative Untersuchungen erfordern. Alternativ könnte versucht werden, die hier untersuchten Einzelfälle in allgemeinere fachdidaktische und lerntheoretische Modelle einzubetten (vgl. Klüver 1995, S. 291). Eine solche fundierte theoretische Einbettung kann im Rahmen dieser Arbeit nicht mehr stattfinden. Sie würde jedoch wahrscheinlich dazu führen, dass auch die Thesen selbst weiter abstrahiert, „sozusagen ‚bereinigt' und in eine idealtypische Form gebracht" werden müssten (ebd., S. 295). Dabei müsste darauf geachtet werden, dass durch die Abstraktion nicht auch die „Besonderheit der Petrinetze" verloren geht und die verbleibenden Aussagen somit nur noch Aussagen über allgemeine Modellbildung im Informatikunterricht wären.

5 Fazit und Ausblick

Ziel dieser Arbeit war es, Petrinetze – eine grafische und simulationsfähige Modellierungstechnik für verteilte Systeme – für den schulischen Informatikunterricht der Sekundarstufen I und II didaktisch-methodisch aufzuarbeiten und curricular einzuflechten. Zu diesem Zweck wurde ein Spiralcurriculum entwickelt, welches aus insgesamt fünf Themenbausteinen besteht. Dabei wurden Themen gewählt, die bereits fest im schulischen Informatikunterricht etabliert und auch praktisch-methodisch gut untersucht sind. Es wurde für jeden Themenbaustein gezeigt, wie Petrinetze darin sinnvoll integriert werden können und welche Lernziele und Kompetenzen sich auf diese Weise prinzipiell fördern lassen. Anschließend wurden zwei der fünf Bausteine exemplarisch erprobt und qualitativ evaluiert.

Für die Sekundarstufe I wurde der Themenbaustein zu eingebetteten Systemen in einer neunten Klasse im Wahlpflichtkurs erprobt. Die Evaluation deutet darauf hin, dass sich Petrinetze bereits in jüngeren Jahrgangsstufen erfolgreich einsetzen lassen, um interaktive und eingebettete Systeme zu modellieren und zu veranschaulichen. Im Gegensatz zu vielen anderen Modellierungstechniken, die bisher im Unterricht verwandt werden, gehen Petrinetze im Kern nicht von einer sequentiellen und terminierenden Verarbeitung im Sinne des EVA-Prinzips aus. Für viele moderne Informatiksysteme machen Fragen nach der Terminierung oder dem produzierten Endergebnis schlicht keinen Sinn. Petrinetze können derartige Systeme authentischer abbilden und im Unterricht somit eine völlig neue Perspektive auf den Begriff „Algorithmus“ eröffnen. Diesbezüglich wäre es sehr interessant, in Zukunft auch längerfristige Effekte zu untersuchen. Führt eine spiralcurriculare Betrachtung von Petrinetzen im Verlauf mehrer Jahrgangsstufen dazu, dass die Lernenden andere Vorstellungen darüber entwickeln, nach welchen grundlegenen Prinzipien Informatiksysteme und Algorithmen arbeiten?

Für die Sekundarstufe II wurde der Themenbaustein zu nebenläufigen Prozessen in einer zwölften Klasse im Leistungskurs erprobt. Die Evaluation deutet darauf hin, dass sich Petrinetze sehr gut zur Veranschaulichung und Analyse von nebenläufigen Prozessen und den damit verbundenen Konzepten des wechselseitigen Ausschlusses und der Verklemmung eignen. Petrinetze ermöglichen es somit, das Verhalten solcher Systeme grafisch und interaktiv für die Lernenden erfahrbar zu machen, ohne, dass diese zunächst aufwändig in Software implementiert werden müssen. Vor dem Hintergrund, dass eine solche Software-Implementation vermutlich nur wenig zum Verständnis der eigentlichen nebenläufigen Prozesse beiträgt, stellen Petrinetze hier also einen echten didaktisch-methodischen Mehrwert dar. Sie sind nicht länger zusätzlicher Unterrichtsstoff, für den kostbare Zeit geopfert werden muss. Sie sind hier die Methode, die es erlaubt, die eigentlichen Unterrichtsinhalte einfach und intuitiv zu veranschaulichen.

Im Rahmen der Arbeit habe ich immer wieder für einen handlungs- und zweckorientierten Umgang mit Petrinetzen bzw. mit Modellen im Allgemeinen argumentiert. Die Lernenden sollen Modelle nicht nur erstellen, sondern sie auch benutzen um ein konkretes Problem zu lösen. Sie sollen nicht Petrinetze um ihrer selbst Willen verwenden, sondern um mit ihrer Hilfe das modellierte System zu verstehen. Eine wichtige Rolle spielte dabei die Simulation, die es den Lernenden erlaubte, ihre Modelle immer wieder zu testen und so das Verhalten des Modells direkt zu beobachten. Insbesondere in jüngeren Jahrgangsstufen ist die Möglichkeit einer interaktiven Simulation meiner Meinung nach ein klarer Vorteil von Petrinetzen gegenüber den eher statischen Strukturmodellen, die sonst im Informatikunterricht vorherrschen. Pseudo-Codes, Programmablaufpläne oder Struktogramme beinhalten keinerlei dynamische Elemente und eignen sich somit nicht zur grafischen Simulation des von ihnen modellierten Verhaltens. Eine Simulation eines Ablaufs ist höchstens in Form eines sogenannten „Schreibtischtests“ mit

Stift und Papier möglich. Eine solche manuelle Abarbeitung von Instruktionen stellt jedoch eher eine mühsame Konzentrationsübung als eine hilfreiche Funktion zum Testen eines Modells dar. Petrintze erlauben es hingegen, die Bewegungen der Marken und somit das Verhalten des Modells direkt zu beobachten.

Eine automatisierte Simulation am Computer erfordert jedoch ein entsprechendes Software-Werkzeug. Ein solches Werkzeug zu finden, das alle gewünschten Konzepte unterstützt, den schulischen Rahmenbedingungen gerecht wird, d.h. möglichst plattformunabhängig und frei verfügbar ist, und zudem noch einfach zu bedienen ist, gestaltete sich jedoch als schwierig. Daher habe ich im Rahmen dieser Arbeit selbst einen Petrinetz-Editor entwickelt, der speziell für den schulischen Einsatz konzipiert ist. *Versys* wurde ebenfalls im Rahmen der Erprobung evaluiert. Es hat sich gezeigt, dass der Umgang mit dem Programm für die Lernenden scheinbar sehr intuitiv war. Die Bedienung bedurfte so gut wie keiner weiteren Erklärungen. *Versys* erlaubt, analog zu verschiedenen didaktischen Programmierumgebungen, keine Syntaxfehler. Dies ermöglichte es, dass sich die Lernenden in der Sekundarstufe I sowohl die Netzstruktur als auch die Schaltregel für elementare Petrinetze mithilfe des Programms selbstständig erbeiten konnten, was wiederum zu einer höheren Lernerorientierung führte. Es hatte jedoch auch zur Folge, dass sich die Lernenden im weiteren Unterrichtsverlauf nicht mehr bewusst mit der Netzstruktur, d.h. der Syntax, von Petrinetzen auseinandersetzen mussten. Dies muss nicht als Nachteil gewertet werden. *Versys* entlastet die Lernenden bei der Modellierung von Fragen der Syntax und erlaubt es ihnen somit, sich stärker auf die Semantik, d.h. auf das zu modellierende System, zu konzentrieren. *Versys* unterstützt daher die vielfach in dieser Arbeit geforderte handlungs- und zweckorientierte Verwendung von Petrinetzen.

Natürlich ist mit dieser Arbeit weder die Frage nach der Vermittelbarkeit von Petrinetzen noch die Frage nach einer sinnvollen curricularen Einbettung

abschließend geklärt. Das vorgestellte Spiralcurriculum bleibt zunächst ein eher theoretisches Konzept, welches weiter praktisch erprobt, evaluiert und gegebenfalls angepasst werden muss. Die hier durchgeführte qualitative Evaluation zweier Unterrichtsblöcke kann dabei nur ein erster Schritt sein. So blieb zum Beispiel das Konzept der variablen Kantengewichte in den Erprobungen völlig unberücksichtigt, da es in keinem der behandelten Modelle enthalten war. Ob und wie Lernende auch in der Sekundarstufe I schon damit umgehen können bleibt somit zu prüfen. Auch die in Abs. 4.3 formulierten Thesen sind erst einmal nichts weiter als Thesen. Sie können jedoch Ansatzpunkte für weitere empirische Untersuchungen liefern.

Die weitere Evaluation und Validierung des Curriculums ist meiner Meinung nach ein erstrebenswertes Ziel. Petrinetze können einen wertvollen Beitrag zum schulischen Informatikunterricht leisten. Sie eignen sich an vielen Stellen, um informatische Systeme zu modellieren und ihr Verhalten grafisch zu veranschaulichen. Sie ermöglichen es, allgemeinere Konzepte wie Kommunikation, Nebenläufigkeit und Kausalität zu diskutieren und eröffnen in einigen Bereichen sogar völlig neue Perspektiven auf die jeweiligen Unterrichtsgegenstände, z.B. im Bereich der eingebetteten Systeme oder der formalen Sprachen. In jedem Fall haben sie aber das Potential, die Vielfalt der informatischen Modelle im Unterricht zu bereichern und somit einen wertvollen Beitrag zum allgemeinbildenden Wert des Informatikunterrichts zu leisten.

Literaturverzeichnis

Abramsky, Samson (2008). „Petri Nets, Discrete Physics, and Distributed Quantum Computation." In: *Concurrency, Graphs and Models*. Hrsg. von Pierpaolo Degano et al. Berlin und Heidelberg: Springer, S. 527–543.

Baumann, Rüdiger (2011). „Eingebettete Systeme verstehen: Teil 1: Kreatives Experimentieren mit Arduino." In: *Log In* Heft Nr. 171, S. 49–61.

Bause, Falko und Kritzinger, Pieter S. (2002). *Stochastic Petri Nets: An Introduction to the Theory*. 2. Aufl. Braunschweig und Wiesbaden: Vieweg.

Berliner Landesinstitut für Schule und Medien (LISUM) (2006a). *Rahmenlehrplan für die gymniasale Oberstufe: Informatik*. Hrsg. von Senatsverwaltung für Bildung, Jugend und Sport. 1. Aufl. Berlin.

– (2006b). *Rahmenlehrplan für die Sekundarstufe I: ITG/Informatik*. Hrsg. von Senatsverwaltung für Bildung, Jugend und Sport Berlin. 1. Aufl. Berlin.

Bortz, Jürgen und Döring, Nicola (2002). *Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler*. 3. Aufl. Berlin, Heidelberg und New York: Springer.

Brandes, Torsten (2009). „Ein Ausflug in den Compilerbau: Syntaktische Analyse und Berechnung arithmetischer Ausdrücke." In: *Log In* Heft Nr. 156, S. 59–64.

Brauer, Wilfried und Reisig, Wolfgang (2009). „Carl Adam Petri and 'Petri Nets'." In: *Fundamental Concepts in Computer Science*. Hrsg. von Erol Gelenbe und Jean-Pierre Kahane. Bd. 3. London: Imperial College Press, S. 129–139.

Brosziewski, Achim und Maeder, Christoph (2007). *Fokus Unterricht: Unterrichtsentwicklung durch Beobachtung*. Zürich: Seismo.

Brühne, Thomas (2009). *Erneuerbare Energien als Herausforderung für die Geographiedidaktik: Perspektiven der Integration in Theorie und Praxis*. 1. Aufl. Wiesbaden: VS Verlag für Sozialwissenschaften.

Bruner, Jerome S. (1980). *Der Prozeß der Erziehung*. Übers. von Arnold Harttung. 5. Aufl. Berlin: Berlin-Verlag.

David, René und Alla, Hassane (2005). *Discrete, Continuous and Hybrid Petri Nets*. Berlin, Heidelberg und New York: Springer.

Dresing, Thorsten und Pehl, Thorsten (2013). *Praxisbuch Interview, Transkription & Analyse: Anleitungen und Regelsysteme für qualitativ Forschende*. 5. Aufl. Marburg. URL: `http://www.audiotranskription.de/Praxisbuch-Transkription.pdf`.

Flick, Uwe (2011). *Triangulation: Eine Einführung*. Hrsg. von Ralf Bohnsack et al. 3. Aufl. Wiesbaden: VS Verlag für Sozialwissenschaften.

Gesellschaft für Informatik e. V. (GI) (2008). *Grundsätze und Standards für die Informatik in der Schule: Bildungsstandards Informatik für die Sekundarstufe I*. Beilage zu Log In Heft Nr. 150/151.

Gramm, Andreas et al. (2012). *E-Mail (nur?) für Dich*. Website. URL: http://medienwissenschaft.uni-bayreuth.de/informatik-im-kontext/index.php/entwuerfe/email-nur-fuer-dich/.

Gudjons, Herbert (2008). *Handlungsorientiert lehren und lernen: Schüleraktivierung, Selbsttätigkeit, Projektarbeit*. 7. Aufl. Bad Heilbrunn: Klinkhardt.

Haddad, Serge und Poitrenaud, Denis (1999). „Theoretical Aspects of Recursive Petri Nets." In: *Application and theory of Petri nets 1999*. Berlin: Springer, S. 228–247.

Hermes, Alfred (2003). „Das Problem der speisenden Philosophen." In: *Log In* Heft Nr. 124, S. 23–32.

Hirsch, Michael et al. (2000). „Zugänge zur Informatik mit Mindstorms: Mit LEGO-Robotern in der Sekundarstufe I spielerisch ein Informatiksystem entdecken und gestalten: Teil 1." In: *Log In* 20 Heft Nr. 2, S. 34–46.

Huang, Ronggui (2012). *RQDA: R-based Qualitative Data Analysis*. R package version 0.2-3. URL: http://rqda.r-forge.r-project.org/.

Hubwieser, Peter (2007). *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. 3. Aufl. Berlin: Springer.

Hubwieser, Peter et al. (2009). *Informatik: Lehrwerk für Gymnasien*. 1. Aufl. Bd. 4. Stuttgart und Leipzig: Klett.

– (2010). *Informatik: Lehrwerk für Gymnasien*. 1. Aufl. Bd. 5. Stuttgart und Leipzig: Klett.

Hubwieser, Peter et al. (2013). *Fundamente der Informatik: Funktionale, imperative und objektorientierte Sicht, Algorithmen und Datenstrukturen*. 2. Aufl. München: Oldenbourg.

Humbert, Ludger (2006). *Didaktik der Informatik*. 2. Aufl. Wiesbaden: Teubner.

Jank, Werner und Meyer, Hilbert (2002). *Didaktische Modelle*. 5. Aufl. Berlin: Cornelsen Scriptor.

Jensen, Kurt und Kristensen, Lars M. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Dordrecht u. a.: Springer.

Klüver, Jürgen (1995). „Das Besondere und das Allgemeine: Über die Generalisierbarkeit in der qualitativen Sozialforschung." In: *Bilanz qualitativer Forschung Band I: Grundlagen qualitativer Forschung*. Hrsg. von Eckard König und Peter Zedler. Weinheim: Deutscher Studien Verlag, S. 285–307.

Koch, Ina et al., Hrsg. (2011). *Modeling in Systems Biology: The Petri Net Approach*. London: Springer.

Köhler, Michael et al. (2001). „Modelling a sociological case study." In: *Sozionik aktuell* 3. URL: http://www.informatik.uni-hamburg.de/TGI/forschung/projekte/sozionik/journal/3/masho-4.ps.gz.

Kuckartz, Udo et al. (2008). *Qualitative Evaluation: Der Einsteig in die Praxis*. 2. Aufl. Wiesbaden: VS Verlag für Sozialwissenschaften.

Kultusministerkonferenz (2004). *Einheitliche Prüfungsanforderungen Informatik*. Online. URL: http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf.

Lehner, Martin (2009). *Allgemeine Didaktik*. 1. Aufl. Bern, Stuttgart und Wien: Haupt.

– (2012). *Didaktische Reduktion*. 1. Aufl. Bern, Stuttgart und Wien: Haupt.

Marwedel, Peter (2000). „Eingebettete Systeme." In: *Log In* 20 Heft Nr. 6, S. 16–18.

Meyer, Hilbert (2009). *Unterrichtsmethoden I: Theorieband*. 13. Aufl. Berlin: Cornelsen Scriptor.

Modrow, Eckart und Strecker, Kerstin (2011). „PuMa II: Haus-Automatisierung mit S4A, PicoBoard und Arduino." In: *Log In* Heft Nr. 169/170, S. 121–124.

Moser, Karin S. (2003). „Mentale Modelle und ihre Bedeutung: Kognitionspsychologische Grundlagen des (Miss-)Verstehens." In: *Sinnbildlich schief: Missgeschicke bei Symbolgenese und Symbolgebrauch*. Bd. 13. Schriften zur Symbolforschung. Bern u. a.: Peter Lang, S. 181–205. URL: http://www.symbolforschung.ch/files/pdf/karin_moser.pdf. Online-Revision.

Nohl, Arnd-Michael (2012). *Interview und dokumentarische Methode: Anleitungen für die Forschungspraxis*. Hrsg. von Ralf Bohnsack et al. 4. Aufl. Wiesbaden: Springer VS.

Powers, K. et al. (2007). „Through the Looking Glass: Teaching CS0 with Alice." In: *ACM SIGCSE Bulletin* 39.1, S. 213–217.

Priese, Lutz und Wimmel, Harro (2008). *Petri-Netze*. 2. Aufl. Berlin und Heidelberg: Springer.

Reisig, Wolfgang (1998). *Elements of Distributed Algorithms: Modeling and Analysis with Petri Nets*. Berlin und Heidelberg: Springer.

– (2010). *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. 1. Aufl. Berlin: Vieweg+Teubner.

– (2013). Persönliches Gespräch vom 8. Okt. 2013 zu den Anwendungsgebieten von Petrinetz-Sprachen.

Resnick, Mitchel et al. (2009). „Scratch: Programming for all." In: *Communications of the ACM* 52.11, S. 60–67.

Rotering, Thorsten (2010). *Petrinetze als Modellierungswerkzeug auf dem Weg zur Implementation von Steuerungsprozessen in der Sekundarstufe II*. Staatsexamensarbeit. URL: http://rotering-net.de/edu/files/staatsarbeit2.pdf.

Schildknecht, Alfreda (2003). „Kooperation und Konkurrenz." In: *Log In* Heft Nr. 124, S. 17–22.

Schöning, Uwe (2008). *Theoretische Informatik – kurz gefasst*. 5. Aufl. Heidelberg: Spektrum.

Schubert, Sigrid und Schwill, Andreas (2011). *Didaktik der Informatik*. 2. Aufl. Heidelberg: Spektrum.
Schwill, Andreas (1993). „Beschreibung paralleler Abläufe mit Petri-Netzen.“ In: *Log In* 13 Heft Nr. 5, S. 45–51.
Stein, Lynn Andrea (2006). „Interaction, Computation, and Education.“ In: *Interactive Computation: The New Paradigm*. Hrsg. von Dina Q. Goldin. Berlin: Springer, S. 463–484.
Strecker, Kerstin (2009). „Informatik für Alle: Wie viel Programmierung braucht der Mensch?“ Diss. Göttingen: Georg-August-Universität Göttingen.
TGI-Gruppe der Universität Hamburg (2013). *Petri Nets Tool Database*. Webseite. URL: `http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html`.
Thomas, Marco (2002). „Informatische Modellbildung: Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht.“ Diss. Potsdam: Universität Potsdam.
Weigend, Michael (2007). „Intuitive Modelle der Informatik.“ Diss. Potsdam: Universität Potsdam.
Wikarski, Dietmar (1997a). „Netze: Wie man über sie und mit ihrer Hilfe kommunizieren kann: Teil 1 – Einführung.“ In: *Log In* 17 Heft Nr. 2, S. 41–45.
– (1997b). „Netze: Wie man über sie und mit ihrer Hilfe kommunizieren kann: Teil 2 – Netztypen und ein Anwendungsbeispiel.“ In: *Log In* 17 Heft Nr. 3/4, S. 76–80.
– (1997c). „Netze: Wie man über sie und mit ihrer Hilfe kommunizieren kann: Teil 3 (Schluß) – werkzeuge und Methoden.“ In: *Log In* 17 Heft Nr. 6, S. 48–51.
Witzel, Andreas (2000). „Das problemzentrierte Interview.“ In: *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research* 1.1. URL: `http://www.qualitative-research.net/index.php/fqs/article/view/1132/2519`.

Alle Online-Quellen wurden zuletzt am 17. Dezember 2013 abgerufen.

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbstständig verfasst und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Weiterhin erkläre ich, eine Masterarbeit in diesem Studiengebiet erstmalig einzureichen.

Berlin, den 17. Dezember 2013 .....

Anhang

A Eingebettete Systeme **91**

A.1 Unterrichtsentwurf und -material 91

A.2 Beobachtung 97

A.2.1 Protokoll 97

A.2.2 Exemplarische Schülernetze 99

A.3 Interview 101

A.3.1 Leitfaden 101

A.3.2 Transkript 102

A.4 Schülertest 108

A.4.1 Aufgaben und Erwartungshorizont 108

A.4.2 Ergebnisse 110

B Nebenläufige Prozesse **115**

B.1 Unterrichtsentwurf und -material 115

B.2 Beobachtung 120

B.2.1 Protokoll 120

B.2.2 Exemplarische Schülernetze 121

B.3 Interview 124

B.3.1 Leitfaden 124

B.3.2 Transkript 124

B.4 Schülertest 132

B.4.1 Aufgaben und Erwartungshorizont 132

B.4.2 Ergebnisse 135

Die Namen der Schülerinnen und Schüler wurden geändert.

Verwendete Abkürzungen:

L	:	Lehrkraft
SuS	:	Schülerinnen und Schüler
PPT	:	Projektor-Präsentation
EA	:	Einzelarbeit
PA	:	Partnerarbeit
WPS / LKS	:	ein(e) Wahlpflicht- bzw. Leistungskurs-Schüler(in), welche(r) genau konnte nicht rekonstruiert werden

Das für die Interviews verwendete Transkriptionssystem orientiert sich an den Empfehlungen von Dresing und Pehl (2013, S. 19-24):

(..)	:	kurze Pause; Punktanzahl entspricht Pausenzeit in Sek.
(4)	:	längere Pause von mehr als drei Sekunden
NEIN	:	betont
Das habe/	:	Abbruch des Satzes
(doch?)	:	Unsicherheit bei der Transkription
(unv., Rauschen)	:	unverständliche Äußerung, ggf. Grund
(lacht)	:	emotionale oder nonverbale Ereignisse
I	:	interviewende Person
B	:	befragte Person

Die Protokolle sind mit Zeitangaben der Form `[h'mm]`, die Transkripte mit Zeitangaben der Form `[mm:ss]` versehen.

A Eingebettete Systeme

A.1 Unterrichtsentwurf und -material

Bei der Lerngruppe handelt es sich um einen Informatik-Wahlpflichtkurs in der neunten Jahrgangsstufe, bestehend aus vier Schülerinnen und neun Schülern. Die Lernenden nehmen in diesem Halbjahr zum ersten Mal am regulären Informatikunterricht teil. Informatische Modelle, insbesondere Petrinetze, wurden bislang nicht im Unterricht behandelt. Es ist daher davon auszugehen, dass das Vorwissen in diesem Bereich, wenn überhaupt, sehr gering ist. Ich rechne nicht damit, dass ein Schüler oder eine Schülerin schon einmal etwas von Petrinetzen gehört hat.

Das Thema der 90-minütigen Unterrichtseinheit ist eine allgemeine Einführung in eingebettete Systeme und die Modellierung mit elementaren Petrinetzen. Für eine Sachanalyse und Legitimation am Rahmenlehrplan siehe Abs. 3.2.1. Motiviert wird das Thema zu Beginn der Stunde mithilfe verschiedener Bilder von eingebetteten Systemen (PPT). Die Lernenden sollen erkennen, dass sich in all diesen Dingen ein kleiner Computer befindet, der darin eine ganz spezielle Aufgabe übernimmt. Es folgt eine (didaktisch reduzierte) Definition für eingebettete Systeme: „Ein eingebettetes System ist ein kleiner Computer mit einer ganz spezifischen Aufgabe, der in ein größeres System eingebettet ist. Meistens regelt, steuert oder überwacht er dabei etwas.“ Im weiteren Verlauf des Blocks sollen die Lernenden zwei eingebettete Systeme mithilfe von Petrinetzen analysieren und modellieren.

Zunächst gilt es, die grundlegenden Konzepte zu Petrinetzen einzuführen. Dies geschieht mithilfe eines Arbeitsblattes (AB1) zum elektronischen Schließsystem (Abb. 6). Die Lernenden sollen das gezeigte Model zunächst in *Versys* nachbilden. Dazu wird ihnen eine unvollständige Version zur Verfügung gestellt. Das Modell wurde aus didaktischen Gründen um eine Schlinge zwischen Pin eingeben und Tür geschlossen erweitert, da solche Schlingen später bei der Modellierung des Licht/Lüfter-Systems benötigt werden. Die Lernenden sollen zunächst mithilfe des Programms sowohl die Netzstruktur als auch die Schaltregel für elementare Petrinetze eigenständig erarbeiten. Im Anschluss erfolgt eine kurze Analyse des Schließsystems mithilfe des Modells.

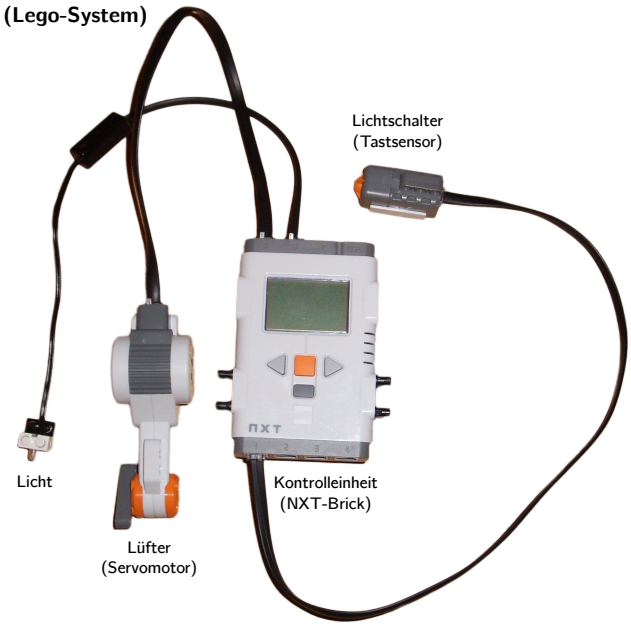
Im zweiten Teil des Blocks wird das Licht/Lüfter-System (Abb. 7) behandelt. Den Lernenden wird zunächt eine reale Implementation in Form eines Lego-Systems vorgeführt (Lego-System). Das System verhält sich entsprechend des Modells: Der Lüfter kann nur starten, wenn das Licht eingeschaltet ist und nur stoppen, wenn es ausgeschaltet ist. Er reagiert dabei mit einigen Sekunden Verzögerung. Ein schnelles Ein- und wieder Aussschalten des Lichtes (oder umgekehrt)

hat also keinen Effekt auf den Lüfter. Es ist davon auszugehen, dass einige Lernende ein solches System bereits aus dem Alltag kennen.

Die Lernenden sollen das Verhalten des Systems nun als Petrinetz modellieren (AB2). Dazu sollen sie zunächst die verschiedenen Aktionen des Systems und ihre entsprechenden Effekte auf das Licht bzw. den Lüfter identifizieren. Sie sollen erkennen, dass die Aktionen des Systems den Transitionen des Netzes und die Zustände den Marken und Plätzen entsprechen. Der Rest der Stunde steht für die Modellierung und anschließende Präsentation der Modelle zur Verfügung.

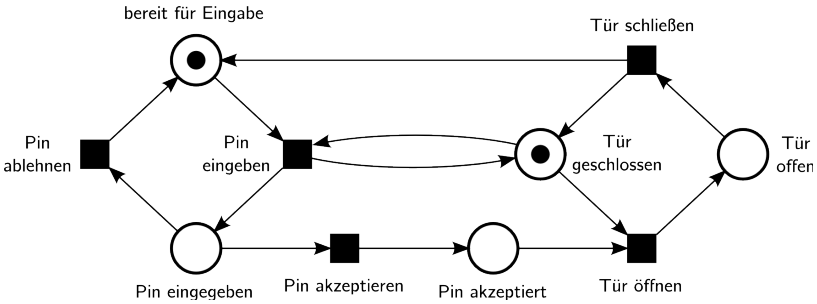
(PPT)

(Lego-System)



Petrinetze und eingebettete Systeme I (AB1)

Das Bild zeigt ein Petrinetz-Modell für ein elektronisches Schließsystem.



1. Starte das Programm *Versys* und öffne die Datei Schließsystem.xml. Kennzeichne im Bild alle Unterschiede und vervollständige das Modell, sodass es aussieht wie im Bild. Finde heraus, welche Begriffe in die Lücken gehören. *(**Hinweis**: Ein Doppelklick auf ein ○, ■ oder ↗ öffnet das jeweilige Eigenschaften-Fenster.)*

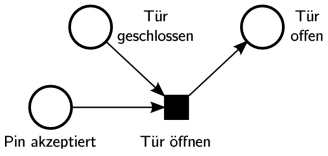
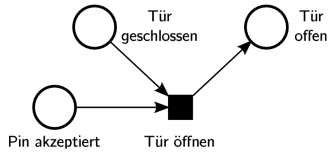
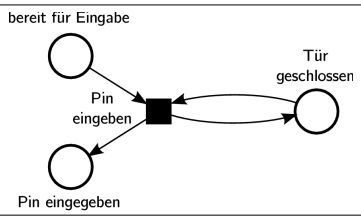
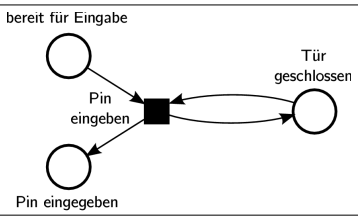
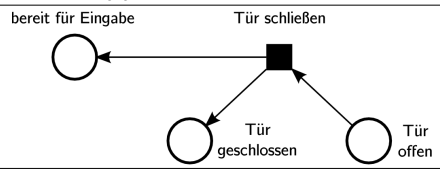
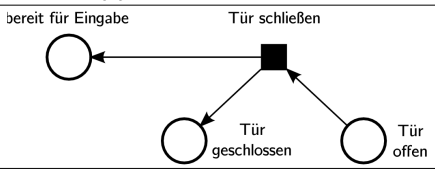
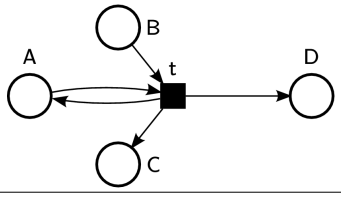
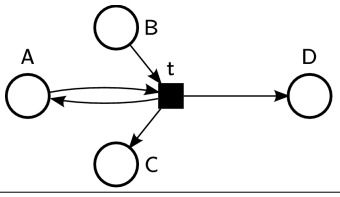
   a) Ein Petrinetz besteht aus

   ○ _____, ■ _____ und ↗ _____.

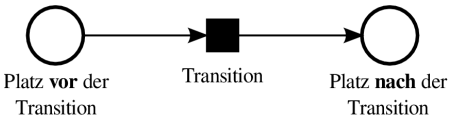
   b) Eine ↗ verbindet immer _____ mit _____ oder umgekehrt.

   c) Auf den ○ _____ können ● _____ liegen.

2. Starte eine manuelle Simulation (▶). Ist eine Transition grün, dann ist sie ***aktiviert***. Klicke auf eine aktivierte Transition um sie ***schalten*** zu lassen. Ergänze in den folgenden Abbildungen die korrekten Marken.

Transition	ist aktiviert (grün)	hat geschaltet (angeklickt)
Tür öffnen		
Pin eingeben		
Tür schließen		
Keine Transition aus dem Schließsystem		

3. Formuliere allgemeine Bedingungen. Benutze die Begriffe ***vor der Transition*** und ***nach der Transition***.



   a) Eine Transition ist aktiviert, wenn _____

   _____

   _____

   _____

   b) Wenn eine Transition schaltet, dann _____

   _____

   _____

   _____

   _____

4. Ein Modell zeigt immer nur einen verkürzten Ausschnitt der Realität. Versuche, die folgenden Fragen mit dem Petrinetz-Modell zu beantworten. Nicht alle können beantwortet werden.

   a) Welche PIN wird akzeptiert?

   b) Wann muss entschieden werden, ob die PIN korrekt ist oder nicht?

   c) Wer entscheided ob die PIN akzeptiert wird (Mensch oder Computer)?

   d) Unter welchen Bedingungen kann die Tür geöffnet werden?

Petrinetze und eingebettete Systeme II (AB2)

Arbeite zusammen mit einem Partner / einer Partnerin.

In einem Petrinetz stellen die Transitionen die verschiedenen Aktionen des Systems dar. Im Schließsystem waren das z.B. **Tür öffnen**, **Tür schließen** oder **PIN akzeptieren**.

1. Beobachtet das Verhalten des Licht/Lüfter-Systems.

   a) Welche Aktionen kommen in dem System vor und welchen Effekt haben sie auf das Licht bzw. den Lüfter?

Aktion Ein Mensch kann... Das Programm kann...	**Vorher**		**Nachher**	
	Licht	**Lüfter**	**Licht**	**Lüfter**
	aus	egal	an	wie vorher
	an	egal	aus	wie vorher
	an	aus	an	an
	aus	an	aus	aus

   b) Welcher Teil der Tabelle beschreibt die Transitionen des Netzes, welcher die Plätze und Marken?

2. Startet *Versys.*

   a) Modelliert das Verhalten des Licht/Lüfter-Systems als Petrinetz.

   b) Testet euer Modell und vergleicht es mit dem Verhalten des Lego-Systems. Kann das Lego-System etwas, was euer Modell nicht kann, oder umgekehrt?

3. Wenn ihr sicher seid, dass euer Modell korrekt arbeitet: Beantwortet die folgenden Fragen und *begründet* eure Antworten mithilfe eures Modells.

   a) Ist garantiert, dass der Lüfter startet, wenn das Licht brennt?

   b) Wenn der Lüfter startet, könnte er danach sofort wieder ausgehen?

Verlaufsplan

Zeit	Phase	Lernsituation	Medien, Sozialform
5 min	Einstieg	Begrüßung. L zeigt PPT: „Was haben alle diese Gegenstände gemeinsam?“ Vorstellung des Themas *Eingebettete Systeme*. **Fazit:** Definition „Eingebettetes System“	PPT, frontal
15 min	Arbeitsphase	SuS bearbeiten Aufg. 1 und 2 zum elektr. Schließsystem.	AB1, *Versys*, EA
5 min	E-Sicherung	SuS stellen ihre Ergebnisse vor.	Plenum
10 min	Arbeitsphase	SuS bearbeiten Aufg. 3 und 4 zum elektr. Schließsystem.	AB1, EA
10 min	E-Sicherung	SuS stellen Ergebnisse vor.	Plenum
5 min	Überleitung	Vorführexperiment: Lego-System. SuS dürfen im Anschluss auch selbst mit dem System interagieren.	Lego-System, Plenum
5 min	Arbeitsphase	SuS bearbeiten Aufg. 1 zum Licht/Lüfter-System.	AB2, Lego-System, PA
5 min	E-Sicherung	SuS nennen die Aktionen des Licht/Lüfter-Systems und identifizieren Transitionen und Plätze. **Fazit:** Aktionen = Transitionen / Zustände = Marken auf Plätzen	
20 min	Arbeitsphase	SuS bearbeiten die Aufg. 2 und 3 zum Licht/Lüfter-System.	AB2, *Versys*, Lego-System, PA
10 min	E-Sicherung	SuS stellen ihre Modelle vor und vergleichen sie mit dem realen Lego-System. **Fazit:** Modelle können uns dabei helfen, die Funktionsweise von technischen Systemen zu verstehen.	Plenum

A.2 Beobachtung

A.2.1 Protokoll

Datum: 12.11.2013 | Beginn: 13:35 Uhr | Jahrgangsstufe: 9 (Wahlpflicht)

Drei SuS schreiben im Nachbarraum einen Test nach. Unvollständiger Merksatz an der Tafel: „Ein eingebettetes System ist ...“ Anwesenheitskontrolle. [0'00]

L-Rechner mit den Bildern stürzt mit Blue Screen ab. L fragt, ob die SuS die Bilder gesehen haben. SuS nennen die gezeigten Dinge. L: „Was haben die gemeinsam?“ Sara: „Sie verbrauchen Strom.“ Hanna: „Sie brauchen etwas, das sie steuert.“ Frank: „Eine Eingabe.“ L: „Intern haben die alle...?“ Sara: „Einen Prozessor.“ L vergleicht die Funktionsweise eingebetteter Systeme mit dem EVA-Prinzip: „Das funktioniert hier nicht so ganz.“ Hanna: „Es fehlt ein Speicher.“ [0'01]

SuS öffnen Hefter. Schreiben die unvollständige Definition von der Tafel ab. L ergänzt: „kleiner Computer mit einer _ Aufgabe.“ Gregor: „Eine oder ganz wenige.“ L ergänzt die Lücke: „ganz speziellen“. L ergänzt: „der in ein größeres System eingebettet ist“ [0'03]

SuS identifizieren die Kontrolleinheit und das Umgebungssystem in den Beispielen aus den Bildern. L ergänzt an der Tafel: „meistens wird dabei etwas geregelt, gesteuert oder überwacht.“ Frank nennt als Beispiel eine Festplatte in einem Rechner und die Steuerung für den Magnetarm. L: „Würde ich nicht sofort ablehnen.“ [0'08]

L teil Arbeitsblatt zum Schließsystem aus. [0'12]

SuS wechseln an die Rechner. Einige SuS versuchen, die Datei direkt im Ordner zu öffnen, ohne vorher Versys zu starten. [0'13]

Alle bis auf Gregor haben nun die Datei in Versys geöffnet. Offenbar gibt es am Rechner von Gregor Probleme mit der Java-Umgebung. Stefanie versucht, eine Kante ins Leere zu erzeugen. Als dies nicht funktioniert, klickt Stefanie scheinbar ziellos im Programm umher. Jens: „Die Datei steckt fest.“ Volker: „Man kann Pfeile nur von da nach da.“ [0'19]

Fast alle SuS haben das Modell korrekt vervollständigt. Gregor schreibt die Namen der Petrinetz-Elemente aus Versys ab. [0'23]

Die erstem SuS beginnen mit der Simulation. [0'25]

Dennis, Emil und Stefan kehren vom Nachschreiben zurück. Stefan verlässt den Unterricht sofort wieder, um an der Versammlung der Gesamtschülervertretung teilzunehmen. Hanna öffnet in Versys das Fenster für die Markenfarben und schließt es nach einigen Sekunden wieder, ohne darin etwas verändert zu haben. SuS beginnen verstärkt, sich untereinander auszutauschen. [0'27]

L: „Bitte wieder in die Mitte!“ SuS kehren an die Tische in der Mitte zurück. Verschiedene SuS nennen die korrekten Begriffe zur ersten Aufgabe. L hält die Begriffe auf einer Overhead-Projektor-Folie fest. L versucht festzustellen, wie weit die SuS im Schnitt mit den Aufgaben gekommen sind und betrachtet die Netze auf den Rechern: „Da ist noch keine Simulation.“ WPS: „War noch nicht so weit.“ L ermahnt Stefanie: „Jetzt sind wir beim Vergleichen.“ Verschiedene SuS nennen die korrekten Marken für die ersten drei Transitionen in Aufgabe 2. L: „Die ersten drei waren eine Beobachtung. Beim Letzten möchte ich eure Einschätzung.“ Sara geht nach vorne und trägt die korrekten Marken auf der Folie ein. L fragt Stefanie, ob die Lösung richtig ist. Antwort: „Ich würde sagen, das ist richtig.“ Stefanie benutzt die Begriffe „zur Transition hinführen“ und „von der Transition wegführen.“ Begründung ist inhaltlich korrekt. [0'29]

[0'36] L leitet zu Aufgabe 3 über.

[0'37] SuS beginnen mit Aufgabe 3.

[0'39] L: „Wer ganz schnell fertig ist, kann schon über 4. nachdenken."

[0'42] Zusammentragen der Ergebnisse zu den Aufgaben 3 und 4. WPS beschreibt, wann eine Transition aktiviert ist: „Wenn eine Kante von einem Platz mit einer Marke zur Transition führt." L: „Mir fehlt da etwas ganz bestimmtes." L hält fest, dass auf ALLEN Plätzen vor der Transition eine Marke liegen muss. Jens: „Achso, ich bin davon ausgegangen, dass nur in einem eine Marke liegen muss." Frank: „Dann geht es aber nicht, ein ODER zu bauen." WPS beschreibt die Veränderung nach dem Schalten einer Transition: „Dann haben alle Plätze nach der Transition eine Marke." L ergänzt: „Eine Marke MEHR."

[0'50] L fragt, wer noch etwas zu Aufgabe 4 hat. Julian hat scheinbar etwas zu (c) modelliert; beschreibt das Verhältnis von Mensch und Computer beim Schließsystem; redet nicht über das Modell, sondern über den intuitiven Sachverhalt: „Erstmal macht der Mensch ... dann macht der Computer." L fragt, woran erkennbar ist, dass die Tür geöffnet werden kann. Stefanie: „Dass sich die Transition grün färbt." WPS: „(a) können wir jetzt nicht bestimmen, bei dem System, das wir gegeben haben."

[0'57] L leitet zum Licht/Lüfter-System über und demonstriert das Lego-System.

[0'59] L: „Dreht mal das Blatt um, bitte."

[1'00] SuS bearbeiten Aufgabe 1 zum Licht/Lüfter-System. L: „Ja, die ersten guten Überlegungen sehe ich. Sehr schön."

[1'04] Zusammentragen der Ergebnisse. Stefanie: „Das zweite, da habe ich ‚Licht ausschalten.'" WPS: „Das letzte ist ‚Lüfter ausschalten.'" SuS zögern bei den Lösungen zur 1. und 3. Zeile.

[1'07] L: „Das Ziel hier ist jetzt so ein Petrinetz selbst zu bauen. Wo seht ihr Transitionen, wo Plätze und Marken?" Stefanie: „Da wo ‚aus' oder ‚an' steht." Stefanie geht nach vorne und zeigt auf den rechten Teil der Tabelle. Jens wird ermahnt. SuS zögern, geben keine weiteren Antworten. L: „Wie könnten wir in dem Petrinetz die Transitionen benennen?" Jens wird vor die Tür geschickt. WPS nennt die korrekten Bezeichnungen für die Transitionen. L ermahnt WPS. L: „Welche Transitionen brauchen wir?" WPS nennt die vier Transitionen. L: „Genau." Frank: „Wie kann man so etwas in einem Petrinetz machen? Es gibt ja keine Funktion ‚warte bla bla.'" L: „Genau. Den Zeitpunkt kannst du nicht modellieren, die Bedingungen schon."

[1'17] SuS wechseln an die Rechner. L zu mir: „Man sieht hier gut die Probleme des Alters und die Konzentration um fünf vor drei." L: „Ich fände es gut, wenn ihr in den nächsten fünf Minuten irgendetwas Sichtbares zustande bekommt. Also konzentriert arbeiten!"

[1'21] Einige SuS verharren noch beim geöffneten Schließsystem. Viele SuS beginnen mit den vier Transitionen. Viele SuS identifizieren die vier Plätze korrekt. Julian hat fast korrekte Netzstruktur. Es fehlen lediglich die Rückrichtungen bei den Schlingen sowie die Marken. Julian startet Simulation. Keine Transition ist aktiviert. Julian beginnt, Marken auf die Plätze zu verteilen. Währen der Simulation gehen die Marken „verloren," da die Schlingen fehlen. Julian platziert mehr Marken auf den Plätzen.

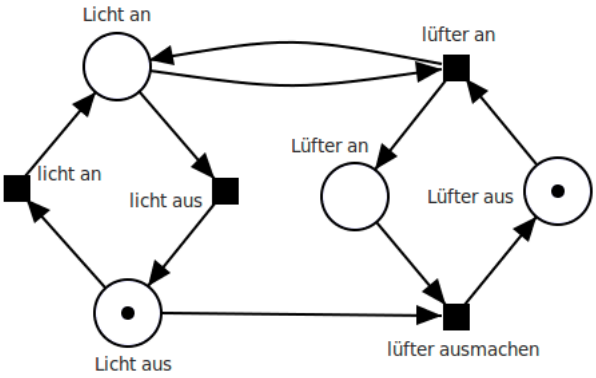
[1'27] SuS sollen Netze auf dem Transferlaufwerk speichern. L: „Kommt jetzt bitte in die Mitte, sonst haben wir keine Zeit mehr." Julian stellt Netz vor: „Das funktioniert nur für einen Ablauf." Unruhe. SuS erkennen das Problem, dass die Marken „verloren gehen." Einige SuS schlagen verschiedene Schaltfolgen vor, um das Problem zu umgehen. WPS: „Wenn man den Lüfter

anschaltet, dann geht eine Marke zurück auf ‚Licht an.‘“ Julian ergänzt eine Schlinge zwischen „Lüfter anschalten“ und „Licht an.“

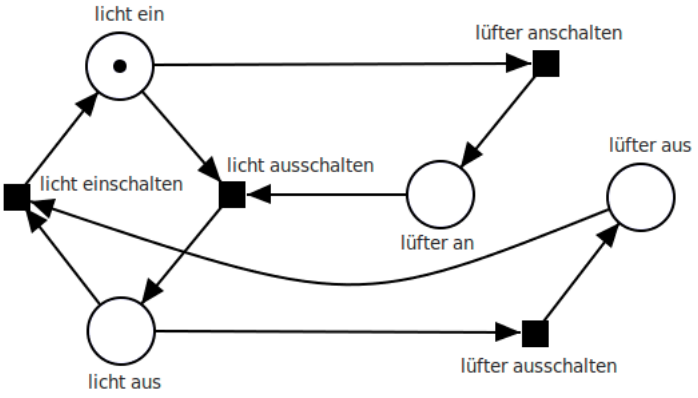
L: „Warum sind hier Marken verschwunden?“ SuS finden keine konkrete Begründung. L beendet Unterricht: „Ich gehe davon aus, dass jeder von euch so ein System lesen kann.“ L kündigt Test an. L stellt mich und meine Masterarbeit kurz vor. L bedankt sich bei den SuS für die zusätzliche Zeit. [1’40]

A.2.2 Exemplarische Schülernetze

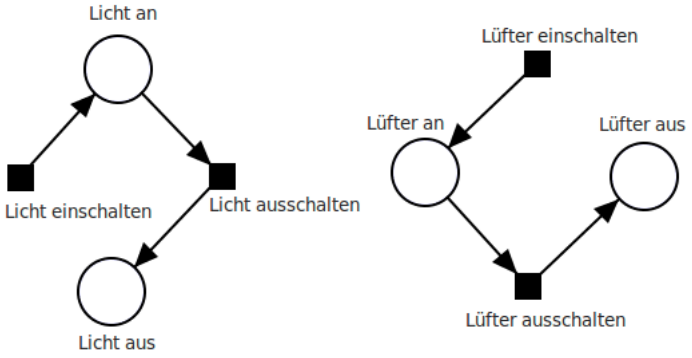
Die Netze wurden zur besseren Les- und Vergleichbarkeit anders angeordnet. Die Topologien, Markierungen und Beschriftungen sind aber in jedem Fall unverändert.



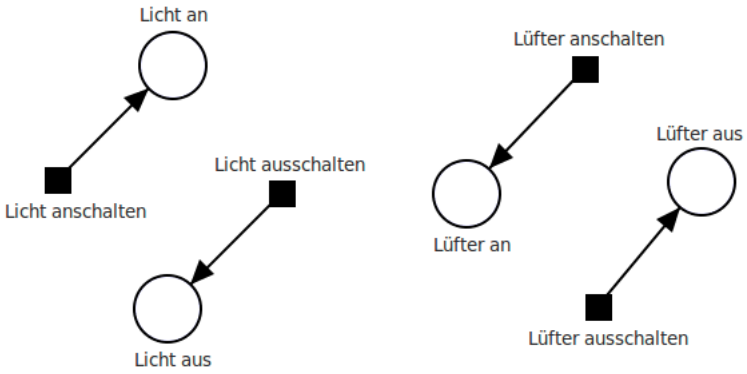
Julian



Jens



Stefanie



WPS

A.3 Interview

A.3.1 Leitfaden

1. **Rahmenbedingungen:**

   Audioaufzeichnung → Anonymisierung des Transkripts

   Erkenntnisinteresse: Lernprozesse der SuS und ihre Auseinandersetzung mit den Unterrichtsinhalten → keine Evaluation der Lehrkaft

2. **Vorformulierte Einstiegsfrage:**

   „Zu Beginn hätte ich gerne, dass du die gehaltene Stunde aus deiner Sicht rekapitulierst. Hier ist noch einmal der ursprünglich geplante Verlaufsplan (Lehrkraft bekommt den Verlaufplan vorgelegt). Geh die Stunde doch bitte einfach nochmal gedanklich durch. Was lief gut? Was lief nicht gut? Was war auffällig?“

   Ggf. Nachfragen zu Aspekten, die zur Einstiegsfrage genannt wurden: Insbesondere Begründungen für die geäußerten Einschätzungen. Warum . . . ?

   Ggf. Nachfragen zu vorab beobachteten „kritischen Ereignissen:“ Was ist da passiert?

3. **Leitfragen:**

   - **Gab es Situationen, in denen die SuS über- oder unterfordert waren?** (Konnkret: welche und warum? Zu Petrinetzen? Zu den modellierten Systemen?)
   - **Gab es Probleme beim Umgang mit Versys?** (Abstürze? Komplikationen? Frustration bei der Bedienung?)
   - **Was haben die SuS aus dem heutigen Unterricht mitgenommen?** (Ist das morgen wieder vergessen? Kann das später im Unterricht noch einmal nützlich werden?)
   - **Wenn du die Stunde noch einmal halten müsstest, was würdest du verändern?** (Aufgaben? Zeitplanung? Methoden? Sozialformen?)
   - **Was glaubst du, wie wird der Kurztest zu Beginn der nächsten Stunde ausfallen?** (Du kennst den Test, du kennst die SchülerInnen, du hast die Stunde erlebt, was meinst du, wie wird der Test ausfallen?)

4. **Zu jeder Leitfrage ggf. Sondierungen:**

   Zurückspiegelung, Verständnisfragen, Konfrontation

A.3.2 Transkript

[00:00] **I:** Ja, dann die Formalien kurz vorweg: ich würde das Interview gerne mitschneiden. Bist du damit einverstanden?

B: Ja.

I: Gut. Zum grundlegenden/ (.) Zur grundlegenden (..) Absicht des Interviews: also es geht darum, die Stunde zu den eingebetteten Systemen im Wahlpflichkurs neun nochmal zu reflektieren und ich hätte gerne deine Einschätzungen der Lernprozesse der Schüler, der Auseinandersetzungen mit den Aufgaben, der Inhalte und so weiter. Also es geht mir nicht darum, DICH in irgendeiner Weise zu bewerten.

B: OK.

I: Gut, dann hätte ich zu Beginn gerne mal, dass du die gehaltene Stunde einfach aus deiner Sicht rekapitulierst. Den Verlaufsplan, den wir vorab gemacht haben, hast du ja vor dir schon liegen. Gehe die Stunde doch bitte nochmal gedanklich durch. Was lief gut? Was lief nicht gut? Was war auffällig?

[00:55] **B:** OK. Also wir haben angefangen mit dem Bild, dem Bild mit den vier eingebetten Systemen. Dieses Bild haben die Schüler am Anfang der Stunde schon gesehen und auch wahrgenommen, sodass sie das auch auswerten konnten. Der Rechner ist dann abgestürzt, was aber grundsätzlich überhaupt kein Drama war, weil die Bilder so anschaulich waren, dass also jeder das auch abgespeichert hatte und (.) das einordnen konnte. Die Erarbeitung zum Begriff „eingebettetes System“ hat dann zeitlich deutlich länger gedauert als es geplant war, einfach aus dem Grund heraus, dass hier mehr als nur ein Einstieg passiert ist, nämlich hier ist eigentlich ein Stück weit eine Erarbeitungsphase dieses Begriffes (.) am Anfang gewesen. Zumindest so, wie wir es geplant hatten, dass da am Ende die Definition des eingebetteten Systems steht. Diese Erarbeitung hat grundsätzlich gut funktioniert. Das ist auch vollkommen in Ordnung, das an der Stelle so mit solchen Bildern zu tun. Würde ich auch so wieder machen, aber man muss sich einfach bewusst sein, dass das eben nicht in fünf Minuten zu machen ist, sondern eben mindestens diese viertel Stunde braucht, die es ja auch gebraucht hat. (..) Tafelbild hast du mitgeschrieben. Dieses, dass man den Merksatz dort eben Schritt für Schritt nochmal erarbeitet hat, nachdem viele Dinge schon genannt wurden. Das, denke ich, war in Ordnung soweit. (..)

[02:22] Dann ging es um das elektronische Schließsystem. Die beiden Aufgaben, dieses erste Arbeitsblatt. Da haben die Schüler (.) am Rechner gearbeitet und (..) hier habe ich vorher mit den Schülern/ (.) also den Schülern noch einmal gesagt, worauf sie achten müssen, was sie erwartet. Hier wäre es von der Planung her sicherlich ganz gut gewesen, wenn ich ihnen einmal dieses Bild, diese Datei gezeigt hätte. (.) Also, wenn ich ihnen einmal gezeigt hätte: „Hier ist der Vorlagenordner, da ist das Programm, das öffne ich jetzt, öffne ich die Datei, so sieht die aus“ und dann eventuell sogar am Platz erst einmal die Unterschiede deutlich, so wie wir das gesagt haben hier. Also die Unterschiede erst einmal visuell wahrnehemen und markieren. So war das jetzt eigentlich (.) überflüssig, dass sie das nochmal markieren sollten. Das hatte ich auch nicht so richtig gesagt. Hat dann den Effekt gehabt, dass ich herumgelaufen bin und dafür gesorgt habe, dass jeder auch die Datei findet. (4) Das ist im Übrigen auf einem Arbeitsblatt auch immer ganz gut, wenn man in irgendeiner Form den Pfad mit drauf hat, wo die Dateien zu finden sind. Also da (.) wäre es (.) gut gewesen in Details noch ein bisschen anders vorbereitet zu sein, aber grundsätzlich haben die Schüler damit gut gearbeitet. Sie haben sich mit dem System vertraut gemacht, haben also dann auch relativ selbstständig heraus gefunden, wie sie dieses System zu verändern haben. Das war sehr intuitiv. Da brauchte ich auch überhaupt nicht

helfen. (.) Wichtig war zwischendrin dann immer wieder der Hinweis auf das Arbeitsblatt, dass da also einiges auch zu vervollständigen ist. Dann haben sie das auch vollständig ausgefüllt, unterschiedlich, vielleicht nicht alle alles, aber zumindest in der Zeit, so jeder in seinem Tempo, hatten sie dort etwas, was wir auch vergleichen konnten. Also das war vollkommen zufriedenstellend. (..) Einschließlich der Transition, die nicht im Schließsystem vorkam. Allerdings habe ich da nicht darauf geachtet, wie viele Schüler das wirklich gehabt haben, ob es mehr als nur dieser eine Schüler hatte. Da habe ich einfach nicht mehr dran gedacht, da mal zu schauen. Da wäre es vielleicht ganz gut gewesen, von meiner Seite aus, noch einmal herum zu gehen, vorher auch einmal zu schauen: „Wie weit sind sie eigentlich?“ Da hatte ich so innerlich den Druck schon ein bisschen wegen der voran geschrittenen Zeit, dass ich da mir diese Ruhe nicht gelassen habe. (...)

So. Das Vorstellen und Vergleichen der Ergebnisse (.) war (.) vollkommen problemlos. Bis [04:58] dahin, dass wir/ dass eben der Schüler diese Lösung zu der (.) losgelösten Transition gebracht hat und hier auch schon erste Formulierungen, Begründungen gebracht wurden, warum das die richtige Lösung ist. Das fand ich sehr schön. Ich fand auch wieder sehr schön, deutlich wahrzunehmen, was der Unterschied ist zwischen dem: „Naja, ich bin davon überzeugt, dass das richtig ist“ bis zu dem (..) das schriftlich zu formulieren, nämlich in der darauf folgenden Aufgabe. Im Nachhinein muss ich sagen, die Probleme, die es bei der darauffolgenden Aufgabe gab, nämlich, dass zum Beispiel auf JEDEM Platz vor der Transition eine Marke liegen muss, darauf hätte ich hier an der Stelle schon eingehen können. Nochmal sagen/ Nochmal direkt die Frage hier stellen können, (.) ob denn nicht auch richtig wäre zum Beispiel, dass in der Spalte „Ist aktiviert grün“ NUR bei A oder NUR bei B eine Marke liegen. (.) Und dann eben wirklich schon einmal dieser Hinweis auf die Pfeile hier auch gegeben wird. Das habe ich jetzt nicht gemacht. Dadurch kam es dann im Endeffekt bei der Formulierung. Ich denke, dann hätte vielleicht der eine oder andere mehr noch hier eine Formulierung gefunden. (...) Das heißt, hier wäre es ganz gut gewesen, sich ein bisschen länger damit aufzuhalten, denn die (.) Formulierungen hier auszufüllen, das war doch nicht ganz trivial. Also, einige Schüler hatten es, von daher ist es gut als Aufgabe gewesen, aber es wäre ganz gut gewesen, wenn hier vorher noch ein bisschen mehr Redeanlass bestanden hätte. Villeicht (habe ich?) sogar (..) noch zwei weitere Transitionen, die aus anderen Systemen stammen, dass die Schüler wirklich da noch mehr Beispiele haben, bevor sie es wirklich verallgemeinern sollen. Das wäre, denke ich, ganz gut (.) gewesen. (..) Nichtsdestotrotz haben wir das formuliert, (..) auch die vierte Aufgabe. Da habe ich kurz überlegt, ob ich sie jetzt weglasse oder nicht weglasse, aber/ (4). Ich war dann hinterher ganz froh, dass zumindest Aufgabe 4b hier auch einmal besprochen wurde, weil hier deutlich dieser Bedingungsteil vorkommt. Also, dass die Tür eben nur geöffnet werden kann, wenn die Tür geschlossen ist UND der Pin akzeptiert ist. Das brauchten wir ja in dem nächsten Entwurf, also das war schon gut, dass ich da auch dran festgehalten habe, das zu machen, trotz der doch engen Zeit, sodass wir da also diese 10 Minuten, die ich am Anfang zusätzlich verbraucht habe, nicht so richtig einholen konnten mehr. (6) (Ich habe dann?) gesehen, dass ich da (unv.) Ich habe hinterher gesehen, dass ich da bei Aufgabe c nicht das Nein hingeschrieben habe, sondern dass der (.) Computer das an der Stelle entscheidet, aber das war/ fand ich jetzt nicht so dramatisch, weil wir ja im Anschluss dann genau dieses Thema wieder hatten, dass wir eigentlich in einem Petrinetz das nicht modellieren, wer es entscheidet, sondern nur aus dem Zusammenhang heraus erkennen. Das war also, denke ich, auch in Ordnung. Die Frage ist wichtig gewesen, (.) weil diese Frage war letztendlich, also 4c, war letztendlich eine Vorraussetzung dafür, dass sie das auch wirklich erfassen konnten im anschließenden Modellierungsbeispiel. Dass eben einmal der Computer Dinge schaltet, oder das Programm Dinge schaltet, und einmal der Mensch (.) der Benutzer Dinge schaltet. (.) Sodass ich denke, dass die Aufgaben, die Fragen tatsächlich da auch wirklich alle wichtig sind. Würde ich also nicht weglassen. (...)

[09:01] So. Dann kamen wir zu dem Vorführbeispiel, das mit dem Licht/Lüfter-System, (...) was, wenn wir jetzt einmal daran denken, dass wir eine neunte Klasse vor uns haben, auch ganz gut war als Überleitung, (.) weil das absolut die Aufmerksamkeit gebündelt hat. Es war wieder sehr ruhig und alle waren sehr konzentriert. Also dieses Vorführexperiment hat einfach einen Neuanfang signalisiert und hat auch wirklich einen Aufforderungscharakter an sich gehabt. Einfach dadurch, dass es eben ein NXT war, das/ dieses Lego-System, mit dem ja einige Schüler auch schon gearbeitet haben. Das war also sehr spannend an der Stelle für die Schüler, hatte ich den Eindruck. (12)

[09:52] Die Tabelle auszufüllen, da brauchten die Schüler schon noch einmal so ein paar Impulse (..) aber (..) das (...), denke ich, kann man wieder so einsetzen. Auch die Zuordnung, wo jetzt die Transitionen stehen (..) haben wir zusammen erarbeitet. Also, war nicht nötig, dass ich ihnen das vorgegeben habe, obwohl ich gemerkt habe, dass da jetzt schon auch an der Stelle das schwer wurde und auch einige Schüler ausgestiegen sind. Also ob das jetzt durch dieses sich gegenseitig ablenken ist, (..) oder einfach so, dass sie im Prinzip gar nichts gemacht haben und geträumt haben und abgewartet haben, wann es denn nun weiter geht, wann es wieder irgendetwas anderes gibt, was man vielleicht tun könnte. (..) Also, das war schon wahrscheinlich hier recht schwierig für die Schüler, dieser Modllierungsprozess. Muss ich jetzt erstmal so festhalten. Es ist allerdings so, dass bis dahin die Aktivierung der Schüler sehr, sehr hoch war, insbesondere, wenn ich diese Lerngruppe vergleiche mit anderen Unterrichtsabschnitten, wo es um andere Themen ging. So, diese Situation, die wir hatten, als es hier um diese Erarbeitung ging, war eher so eine Schwierigkeit, die ich in der Lerngruppe öfter habe. So diese Unaufmerksamkeit, Jens, den ich ja zwischendurch mal rausgestellt habe, der ja sehr große Schwierigkeiten hat, leise zu sein und wirklich zuzuhören und nicht nebenbei irgendwie bei sich mit anderen Dingen oder anderen Schülern zu beschäftigen. (..) Das ist schon eher (.) eine Lerngruppe, die im Moment noch sehr schwierig ist. Wir sind ja am Anfang des Schuljahres und ich habe die noch nicht so lange oder, ja. Also dieser erste Abschnitt ist noch nicht so lange. Einmal die Woche habe ich sie auch nur. Das heißt, müssen auch bestimmte Normen erst noch lernen. Da arbeiten wir noch dran. Und (.) das würde ich jetzt also nicht absolut dem Material hier zum Vorwurf machen, dass da die Aufmerksamkeit gesunken ist. Im Gegenteil, eher würde ich sagen, dass dieser erste Teil besonders gut strukturiert war, weil sie da wirklich die ganze Zeit aktiv waren, (..) so wie ich das (..) so als Gruppe nicht unbedingt immer nur aus dem Unterricht kenne. Also, so das ist/ (..) Also, als Fazit hier draus: wir haben die Transitionen erarbeitet, jeder wusste auch, dass das die Transitionen sind. Das heißt sie sind an den Rechner gegangen und haben selbstständig diese Transitionen auch hergestellt und haben Plätze ringsherum gebaut und haben jetzt versucht, diese Kanten zu ergänzen, hatten allerdings ja nur noch 5 Minuten Zeit oder 10 Minuten, knapp 10 Minuten Zeit. Das war also im Endeffekt sehr, sehr knapp. Überraschend ist, dass trotzdem zwei Beispiele da sind, die schon der richtigen Lösung relativ nahe kamen und auch einiges produziert haben und die in der Vorführung, der eine hat es ja vorgeführt, der eine Schüler, dann sogar auch noch fast vervollständigt werden konnten. Also das muss ich sagen, trotz der Kürze der Zeit ist das enorm. Also, das zeigt, dass, wenn hier noch 10 Minuten mehr Zeit gewesen wären, die also dieses Modell fertig gestellt hätten. Die Besseren, und die Schwächeren hätten einen Teil bekommen. Denn Ansatzpunkte, waren ja in den Beispielen auch, das wirst du sehen, wenn du sie ja auswertest im Detail noch einmal, was sie da gebracht haben. (...)

[13:33] Ich habe mich dann dafür entschieden, einfach nur die Stunde (..) abzurunden, in dem Moment, dass wir uns dieses eine Beispiel anschauen und eben sagen, was da noch/ Also, da nochmal gucken, ob wir da noch was verbessern dran können. Also, einfach nur erstmal (.) das bewerten. Vielliecht hätte man auch das Verbessern zurückfahren sollen und dort an der Stelle nur sagen: „Was erkennen wir denn als richtig hier an?“ Darauf nochmal Wert legen. Das ist

jetzt ein bisschen zu kurz gekommen. (..) Aber (..) womit ich eigentlich fest rechne, was die Schüler nach diesem Block können, ist: sie sind in der Lage solche Petrinetze zu analysieren. (.) Vielliecht noch nicht unbedingt jetzt die Funktionalität bis ins Kleinste zu verallgemeinern, aber die Analyse, vorherzusagen, wie dieses Petrinetz sich verhält, das können sie. Und sie haben einen Zusammenhang, wo man so etwas verwendet auch. Und auf der Basis, denke ich, können wir auch einen Test durchführen. Die Modellierung würde ich noch rauslassen bei der Testaufgabe. (..) Aber das geht. Mhm. Das war es soweit.

I: Mhm. ok. (...) Ja, du hast jetzt schon verschiedene Situationen angesprochen, die deiner Meinung nach für die Schülerinnen und Schüler herausfordernd waren. Also, die Formulierung am Anfang für die Schaltregel und die Aktivierung und auch die Ausführung der Tabelle. Ganz allgemein, gab es Situationen, in denen die Schüler direkt ÜBERfordert waren oder, anders herum, in denen es zu langweilig war, in denen sie UNTERfordert waren? [14:45]

B: (.) Also unterfordert definitiv nicht. (...) Eine Überforderung ansatzweise war diese Modellierung erst einmal noch. (...) Wir hatten ja/ Also die Aufgabe ist ja hier gewesen bei den/ (..) bei dem Licht/Lüfter-System, (.) Aufgabe 1 selbstständig zu machen. (..) Da habe ich ja schon viel mehr vorher/ (Naja das stimmt/ ?) Muss ich mal überlegen. Ich habe das vorgeführt. Ich habe erstmal noch nicht viel dazu gesagt, aber ich habe zumindest ja schon nochmal eingegriffen, weil ja irgendwie das große Rätseln losging. Habe sie nochmal darauf hingewiesen, dass da also „ein Mensch dann kann“ und „das Programm kann“ steht, aber selbst damit waren sie/ Da war so ein bisschen die Orientierungslosigkeit noch da. Das war sehr schwierig, das zu finden, wie das jetzt gemeint war. Da habe ich also durch Impulse, durch Herumgehen und einige Dinge, wo sie dann in die richtige Richtung gedacht haben, das verstärkt und dann auch schon einmal betont „Ah, jemand hat schon was.“ Und dann ging natürlich auch so ein bisschen dieser Austausch los: „Was hat der denn?“ „Was ist denn was Richtiges?“ Das war also so ein bisschen dieses (...) zu versuchen, das, was ich da gesehen habe, dass sich das mal ein bisschen verbreitet, sodass da alle nochmal einen Impuls bekommen, in welche Richtung das weitergeht. Also, das, da hätte man irgendwie/ Vielleicht wäre eine Möglchkeit gewesen, ein Beispiel, das stimmt. Vielleicht die erste Zeile, das „Licht einschalten“ (.) schon einmal auszufüllen (...) und den Rest ausfüllen zu lassen, vielleicht auch, dass man links und rechts Lücken hat. (..) Das war ja auch noch so eine Idee. Also, (...) über die Tabelle, da haben wir auch eine Weile gegrübelt beim Entwurf, die ist noch nicht zuende ausgereift. Das war (.) noch zu unklar, in welche Richtung die Schüler eigentlich denken sollen und deswegen eine Überforderung. Aber die Überforderung wirklich aus der Aufgabenstellung heraus und nicht aus dem Inhalt der Aufgabenstellung. (.) Nur aus der Aufgabenformulierung. Sie hatten also nicht so richtig einen Ansatzpunkt, wohin die Sache gehen soll. Die Zielorientierung fehlte da. (...) [15:21]

I: Irgendetwas anderes, was dir dazu noch einfällt? Also jetzt, generell, Überforderung, Unterforderung?

B: Ich denke, wenn sie ausreichend Zeit gehabt hätten, das ist das einzige an der St/ Die einzige Stelle, über die ich jetzt nachdenke, ist die Modellierung am Rechner mit Versys. Und wenn sie die Zeit gehabt hätten, die wir geplant haben, (.) dann wäre das auch/ hätte das auch gut funktioniert. Davon bin ich überzeugt. (.) [17:47]

I: OK. Ja, gab es es Probleme beim Umgang mit Versys?

B: (5) An einigen Rechner ließ sich das Programm GAR nicht öffnen. Das war aber jetzt nicht wirklich ein Problem. Die haben dann zu zweit gearbeitet, war ja sowieso Partnerarbeit. Das/ (...) Und ein Schüler, neunte Klasse, (.) hat wie wild durch die Gegend geklickt. Da hat erst einmal eine ganze Weile gar nichts mehr funktioniert, bis dann irgendwann das Programm auf [18:16]

seine ganzen vielen, schnell aufeinander folgenden Eingaben reagiert hat. (..) Der Jens vorne. (.) Das war: „Das geht ja gar nicht mehr!“ und dann (.) habe ich mal einen Moment gewartet und mit einem Mal erschienen da zehn Transitionen aus dem Nichts, die er durch irgendwelches wilde Umherklicken eben erzeugt hatte. Das heißt, (..) da muss/ (.) sollte man solchen Schülern in dem Alter eben den Hinweis geben, dass man auch erstmal dem Programm ein bisschen Zeit lassen sollte. Dass also nicht ganz so extrem, schnell aufeinander folgend die Eingaben kommen sollen. (..) Nein, sonst nicht. (.) Wie gesagt, das Netz, diese erste Aufgabe hier zum Schließsystem haben sie ja ganz intuitiv bearbeitet. (...) Ja. (..)

I: Mhm. OK. Hattest du jetzt auch am Anfang schon angesprochen, aber nochmal ganz konkret, vielleicht: Was haben die Schüler aus dem heutigen Unterricht mitgenommen? (...) Sowohl zu Petrinetzen als auch zu eingebetteten Systemen?

[19:45] **B:** Sie haben (..) eine Vorstellung von einem eingebetteten System, in Abgrenzung zu den Universalrechnern, (.) auch wenn wir den Universalrechner jetzt nicht explizit definiert haben, aber genannt haben wir ihn. (..) Sie haben eine Definition eines eingebetteten Systems aufgeschrieben, das heißt, das ist etwas, was Lernwissen auf jeden Fall auch werden kann. (.) Also wieder im Rahmen einer Wiederholung auch abgefragt werden kann, weil es eben im Hefter gesichert ist. Sie haben/ (..) Sie kennen die Bestandteile eines Petrinetzes und können sie benennen, nämlich Plätze, Transitionen, Kanten und Marken. (..) Sie haben mit diesen/ Sie haben sie einerseits aus dem Programm rausgeschrieben, aber sie haben damit auch argumentiert. Das heißt, die haben sie, im Gegensatz zu den eingebetteten Systemen, die wir ja sozusagen erarbeitet und aufgeschrieben haben, haben sie mit diesen Fachbegriffen auch wirklich gearbeitet. (.) Zu einem späteren Zeitpunkt. Einzelne Schüler und das ist also ein Stück weit auch schon in die Fachsprache übergegangen. Zumindest erst einmal kurzfristig. (.) Sie (..) haben die Bedeutung der Kanten (...) als Bedingungen (..) erkannt. Oder sie als Gruppe. Das haben wir auch formuliert. Sie haben auch das in der Modellierung verwendet, also gehe ich davon aus, dass sie das auch/ (.) Sie haben es bewusst eingesetzt, das heißt diese Bedeutung haben sie auch erfasst und können sie auch in der Analyse erkennen. Ebenso die Beutung der Marken als Vorraussetzung bzw. als Veränderung des Systems. Also, sie haben formuliert und ja auch beobachtet und (bei der?) Modellierung auch hinterfragt, dass diese Marken letztendlich das ist, was in dem System sich verändert und, dass das die Bedeutung dieses Systems letztlich auch ausmacht, dass diese Marken da wandern und mit einem Mal auch die Zustände der Transitionen sich ändern (.) dadurch. Das haben sie erfasst.

[22:06] (6) Und sie haben sich darin geübt, so etwas zu formulieren. Also ein System zu beschreiben, auf einem Abstraktionsgrad verallgemeinernd. Also das ist ja sehr leicht gewesen, zu sagen: „Wenn die Tür geschlossen ist und der Pin akzeptiert ist, DANN darf die Tür geöffnet werden,“ aber die Aufgabe 3 war ja schon auf einem hohen Abstraktionsgrad, nämlich von Transitionen und Marken und Plätzen zu reden und nicht von dieser konkreten Bedingung. Und darin haben sie sich geübt. Das heißt, sie sind also auch auf dieser (...) Ebene mit dem hohen Abstraktionsgrad ein Stück weiter gekommen. Das hat jetzt nicht unbedingt etwas mit Petrinetzen zu tun, sondern überhaupt. Sie haben sich darin geübt, so ein abstraktes System zu beschreiben. (5) Ja. (..)

[23:02] Und sie haben erkannt (..) nachdem sie wussten, wo es hingeht, (..) dass eben die Aktionen das ist, was die Transitionen ausmachen. (..) Das haben vielliecht nicht alle bis ins Letzte erkannt, dass sie es wiedergeben, dass sie die jetzt schon erkennen, dass sie in einem anderen Zusammenhang das als (.) Konzept schon anwenden würden, aber zumindest diesen Zusammenhang, der wurde (.) verdeutlicht. Verdeutlicht will ich es mal nennen. Für manche ganz deutlich, andere haben das erst einaml so hingenommen, aber der ist zumindest thematisiert worden, dieser Zusammenhang. (5)

I: Jetzt einmal abgesehen von dem offensichtlichen Zeitproblem, das es in der Stunde gab: Wenn du die Stunde noch einmal halten müsstest, was würdest du verändern?

B: (26) Diese erste Phase würde ich so lassen. Zum Thema eingebettete Systeme, zu dem Begriff. (...) Bei den Petrinetzen würde ich, habe ich schon gesagt, glaube ich, einfach ihnen einmal das zeigen, wo sie die Datei finden. Dann kann man diese Phase auch genauso lassen. (14) In Aufgabe 2 zu den Schließsystemen (.) das teilen und sagen (.) die Beobachtungsaufgaben am Rechner dokumentieren, (..) das vervollständigen (...) dann in die Mitte zurückkehren und dann erst einmal über diese (6) Transitionen, die unabhängig vom Schließsystem sind, nachdenken. Also ich glaube, da würde ich eine Arbeitsphase daraus machen. Eventuell sogar so, dass diese unabhängigen Transitionen Teil der Aufgabe 3 werden. (...) Das ist, denke ich, sinnvoller. (.) Oder die eine da drin lassen, die am Rechner schon mit bearbeitet werden kann (...) egal, wie sie es dann lösen und in Aufgabe 3 noch einmal zwei, drei solche Beispiele, die völlig unabhängig sind.(.) Dann wird das Formulieren vieleicht auch noch leichter. (9) Zu den (.) Licht/Lüfter-Systemen, habe ich schon gesagt, müsste man über die Tabelle noch einmal nachdenken (..) gegebenenfalls einfach eine Zeile schon komplett vorgeben und in den anderen drei Zeilen an verschiedenen Stellen (..) Lücken lassen oder wirklich nur „Licht ausschalten," „Lüfter anschalten," „Lüfter ausschalten" erkennen lassen und den Rest so lassen. (..) Das ist das, was mir jetzt spontan einfällt. (4) Ja. Ob es das dann schon ist, kann ich jetzt genau noch nicht sagen, (.) aber wäre ein Ansatzpunkt, wo man sich noch einmal ransetzen müsste und noch einmal überlegen müsste. (5) [24:58]

Man könnte allerdings auch/ (blättert im Material) (unv.) Muss ich nochmal schauen. (.) Vielleicht doch noch nicht gleich in die Modellierung einsteigen, (.) sondern erstmal noch, dazwischen geschaltet, ein anderes Petrinetz zur Analyse geben. Dass man in diesem Block tatsächlich nur auf der Analyse-Ebene bleibt. Das kann auch gut sein, jetzt, wenn ich mich darauf einlasse, JA, das ist wahrscheinlich einfach zu schnell gewesen. Dieser berühmte Fehler, Anfängerfehler: „Ich habe ein Beispiel gesehen und jetzt modelliere ich gleich mal." Ich muss die Analyse erst einmal vertiefen und sichern und (.) da üben, bevor ich mich an die Modellierung wage. Ich denke, der Schritt war zu schnell hier. Das ist wahrscheinlich die Ursache. Und dann kann man die Tabelle wahrscheinlich noch viel lückenhafter geben. Wenn sie ein paar mehr Beispiele analysiert haben, dann haben sie viel mehr Vorstellung davon, wie so eine Transition funktioniert, was eine Transition sein kann und was nicht. Ich denke, das war der Bruch hier. (.) Ja. (14) Also den Block fertig machen mit Analysen, andere Beispiele, (..) mit und ohne Rechner, (..) wahrscheinlich sogar eher ohne erst einmal, wirklich. (...) Übungen, nachher ableiten, (.) überlegen, was die/ (.) Wie sie sich verhalten (...) und dann erst, im nächsten Block über Modellierungsaufgaben nachdenken. Und dann vielleicht auch dort, erst einmal ein teilfertiges Petrinetz geben, das nach bestimmten Anforderungen erweitert werden kann. Wenn das möglich ist. Das weiß ich nicht, ob das möglich ist, aber so etwas in der Art macht man ja auch, wenn man Automaten oder Algorithmen (..) oder Objekt- oder Klassendiagramme modelliert, dass man erst einmal ihnen irgendetwas halbfertiges gibt und sagt: „Und jetzt erweiter mal, um die und die und die (.) Funktionalität zu erreichen." (...) [27:13]

I: Ok. Dann, den Test, der dann nächste Woche folgen soll, der kurze, den haben wir jetzt noch nicht fertig. Das heißt, über die Aufgaben da jetzt konkret zu sprechen, geht jetzt an der Stelle (.) noch nicht so wirklich, aber, ja, was hast du für ein Bauchgefühl, wenn wir jetzt davon ausgehen, wir würden die Aufgaben so bauen, (..)

B: Analytisch.

I: dass da jetzt keine/ Genau, analytisch also keine Modellierungsaufgaben drin sind.

[30:08] **B:** Keine Modellierugsaufgaben. (.) Analytische Aufgaben sollten die Schüler hinbekommen. (..) Also, (.) Aufgaben in ähnlicher Form wie das, was wir hier (...) in Aufgabe 2 bei den Schließsystemen hatten (5) sollten sie hinbekommen. Sie sollten in der Lage sein, Bedingungen zu formulieren für aktive Transitionen (5) und (5) auch solche Interpretationen, wie wir sie bei dem Türöffnen hatten. Dass wir eben (..) die Kanten interpretieren. Also (..) der Pin kann eben nur eingegeben werden, wenn (..) er bereit für die Eingabe ist und die Tür geschlossen ist. So etwas zum Beispiel. Das sollten sie auf jeden Fall (4) schaffen. (4) Ja. (...)

I: Ja, gut, dann, noch irgendetwas, was dir auf dem Herzen liegt? Irgendwelche Anmerkungen, die ich/ Irgendwie Sachen, auf die ich jetzt nicht eingegangen bin, die dir noch einfallen?

[31:14] **B:** Nein, ich denke ich habe alles gesagt, auch im Zusammenhang mit der Stunde schon. (..) Also (.) Nein. (..)

I: Okay, gut. Dann vielen Dank!

B: Gerne.

A.4 Schülertest

A.4.1 Aufgaben und Erwartungshorizont

Aufgabe 1: Falsche Netze. Für die Aufgabe werden maximal sechs Punkte vergeben: ein Punkt für jedes Netz. Wird in einem Netz ein Fehler nicht erkannt oder eine korrekte Stelle als Fehler erkannt, wird kein Punkt vergeben.

Aufgabe 2: Schaltregel. Für (a) werden maximal drei Punkte vergeben: ein Punkt für jedes Netz. Ein Punkt wird genau dann vergeben, wenn die aktivierte Transition korrekt identifiziert wurde. Bei einem Fehler wird für das Netz kein Punkt vergeben

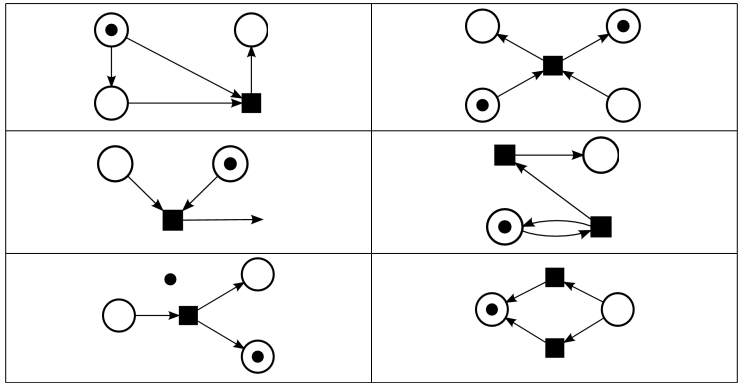
Für (b) werden maximal drei Punkte vergeben: ein Punkt für jedes Netz. Ein Punkt wird genau dann vergeben, wenn die korrekte Markierung eingezeichnet wurde. Bei einem Fehler wird für das Netz kein Punkt vergeben.

Aufgabe 3: Alarmanlage. Für (a) werden maximal drei Punkte vergeben: ein Punkt für jede Transition inkl. Vor- und Nachbereich, wobei sich die vierte Transition von selbst ergibt, wenn die anderen drei korrekt beschriftet wurden. Eine falsche Bezeichnung wird nur genau einmal als Fehler gezählt. Es werden also nicht zwei Punkte abgezogen, wenn ein Platz falsch beschriftet ist, weil er sich im Vorbereich einer Transition und im Nachbereich einer anderen befindet.

Für (b) wird maximal ein Punkt vergeben. Der Punkt wird vergeben, wenn anhand des Petrinetzes korrekt argumentiert wird, warum die Transition Alarm geben nicht schalten kann, bzw. warum keine Marke auf den Platz Alarm gelangen kann. Intuitive Argumentationen über Alarmanlagen werden nicht akzeptiert.

Aufgabe 1: Falsche Netze (6p)

Einige der folgenden Petrinetze sind fehlerhaft. Kreise die fehlerhaften Stellen ein.

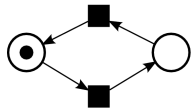
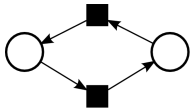
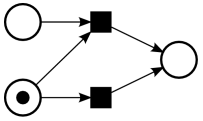
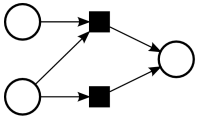
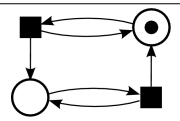
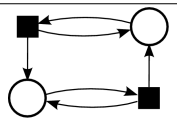


Aufgabe 2: Schaltregel (6p)

In den folgenden Petrinetzen ist jeweils eine Transition aktiviert.

a) Kreise in der linken Tabellenspalte in jedem Netz die aktivierte Transition ein.

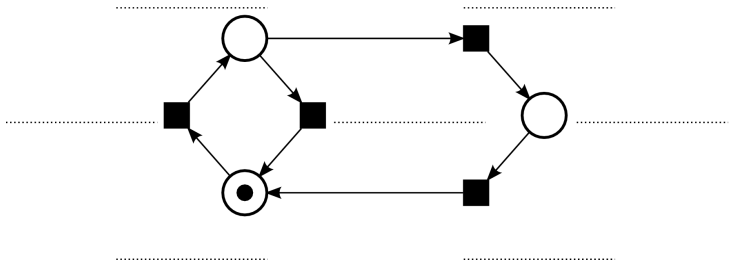
b) Ergänze in der rechten Tabellenspalte die korrekten Marken nachdem die Transition geschaltet hat.

(a) Welche Transition ist aktiviert?	(b) Markierung nach dem Schalten:
	
	
	

Aufgabe 3: Alarmanlage (4p)

Susie hat in ihrem Haus eine Alarmanlage. Wenn sie das Haus verlässt, schaltet sie die Alarmanlage an. Wenn sie zurückkommt, schaltet sie sie wieder aus. Einmal hatte Susie vergessen, ihren Kanarienvogel wieder in den Käfig zu sperren und als sie das Haus verließ, ging der Alarm los. Sie stellte den Alarm ab, sperrte den Vogel ein, und schaltete die Alarmanlage wieder an.

Das folgende Petrinetz modelliert die Alarmanlage von Susie.



a) Ordne den Plätzen und Transitionen die korrekten Bezeichnungen zu:
angeschaltet | ausgeschaltet | anschalten | ausschalten | Alarm | Alarm schlagen | Alarm abstellen

b) Bei Susie wurde eingebrochen, aber der Alarm ging nicht los, weil sie vergessen hatte, die Alarmanlage anzuschalten. Begründe anhand des Petrinetzes, warum der Alarm nicht starten konnte.

_____

_____

_____

_____

A.4.2 Ergebnisse

11 Schülerinnen und Schüler haben an dem Test teilgenommen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	$\tilde{x}$
					1	1	1	2	1	3	1	1					**9**

1. Falsche Netze

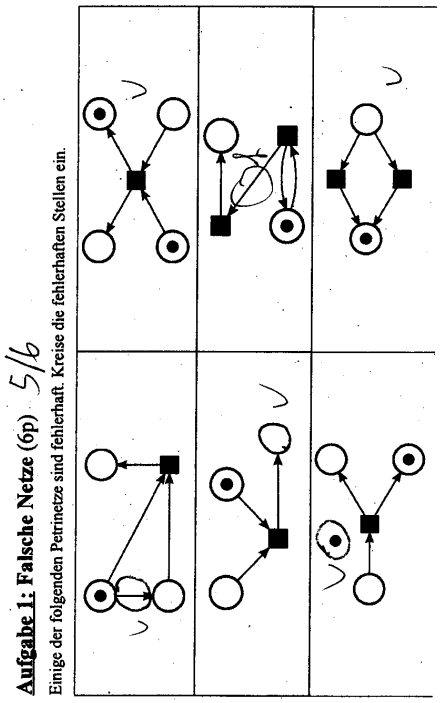
0	1	2	3	4	5	6	$\tilde{x}$
	2	5	3		1		**2**

2. Schaltregel

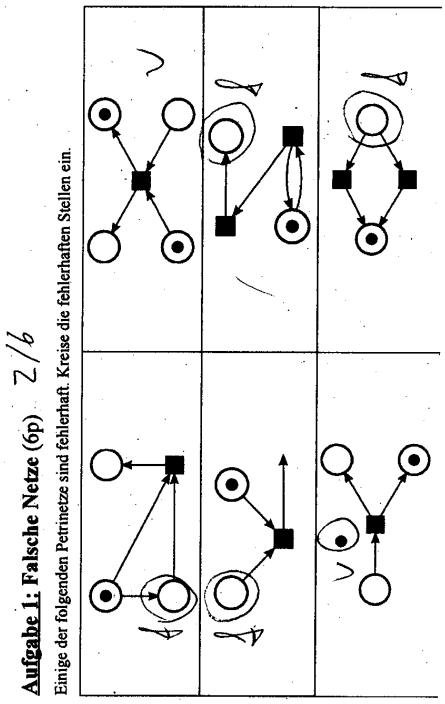
0	1	2	3	4	5	6	$\tilde{x}$
	1		2	2	3	3	**5**

3. Alarmanlage

0	1	2	3	4	$\tilde{x}$
1		2	5	3	**3**



Jens



WPS

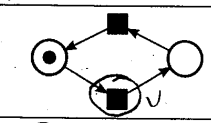
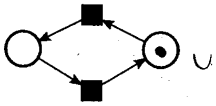
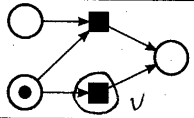
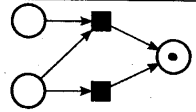
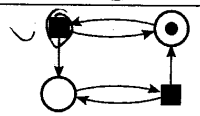
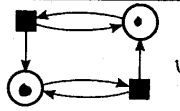
Hanna

Aufgabe 2: Schaltregel (6p) 6/6

In den folgenden Petrinetzen ist jeweils eine Transition aktiviert.

a) Kreise in der linken Tabellenspalte in jedem Netz die aktivierte Transition ein.

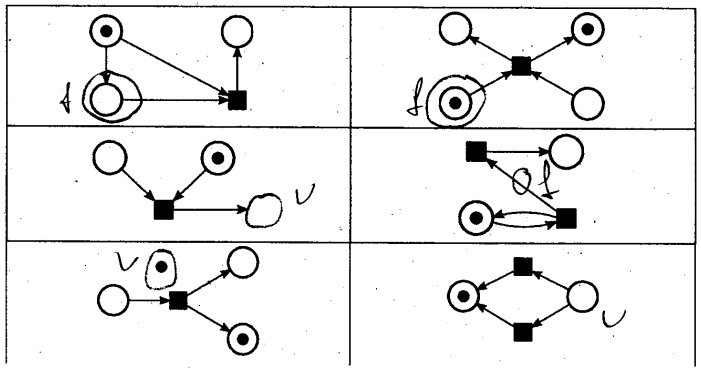
b) Ergänze in der rechten Tabellenspalte die korrekten Marken nachdem die Transition geschaltet hat.

(a) Welche Transition ist aktiviert?	(b) Markierung nach dem Schalten:
	
	
	

Stefanie

Aufgabe 1: Falsche Netze (6p) 3/6

Einige der folgenden Petrinetze sind fehlerhaft. Kreise die fehlerhaften Stellen ein.



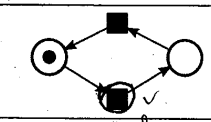
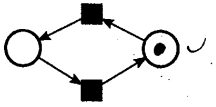
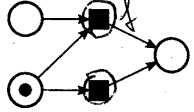
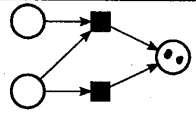
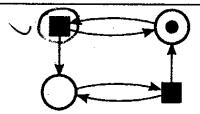
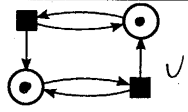
Jens

Aufgabe 2: Schaltregel (6p) 4/6

In den folgenden Petrinetzen ist jeweils eine Transition aktiviert.

a) Kreise in der linken Tabellenspalte in jedem Netz die aktivierte Transition ein. ein

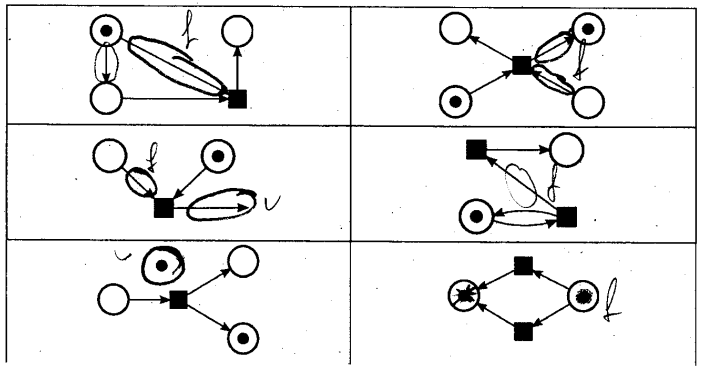
b) Ergänze in der rechten Tabellenspalte die korrekten Marken nachdem die Transition geschaltet hat.

(a) Welche Transition ist aktiviert?	(b) Markierung nach dem Schalten:
	
	
	

Sara

Aufgabe 1: Falsche Netze (6p) 1/6

Einige der folgenden Petrinetze sind fehlerhaft. Kreise die fehlerhaften Stellen ein.



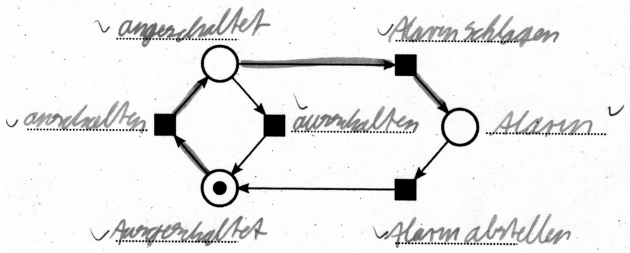
112

Jens

Aufgabe 3: Alarmanlage (4p) 4/4

Susie hat in ihrem Haus eine Alarmanlage. Wenn sie das Haus verlässt, schaltet sie die Alarmanlage an. Wenn sie zurückkommt, schaltet sie sie wieder aus. Einmal hatte Susie vergessen, ihren Kanarienvogel wieder in den Käfig zu sperren und als sie das Haus verließ, ging der Alarm los. Sie stellte den Alarm ab, sperrte den Vogel ein, und schaltete die Alarmanlage wieder an.

Das folgende Petrinetz modelliert die Alarmanlage von Susie.



a) Ordne den Plätzen und Transitionen die korrekten Bezeichnungen zu:
angeschaltet | ausgeschaltet | anschalten | ausschalten | Alarm | Alarm schlagen | Alarm abstellen

b) Bei Susie wurde eingebrochen, aber der Alarm ging nicht los, weil sie vergessen hatte, die Alarmanlage anzuschalten. Begründe anhand des Petrinetzes, warum der Alarm nicht starten konnte.

der Platz „Alarm" kann nur erreicht werden, wenn vorher eine Marke auf dem Platz „angeschaltet" lag, da der grüne weg der einzig mögliche ist, um Alarm zu schlagen.
das heißt, das kein Alarm zustande kommt, wenn Susi die Alarmanlage nicht einschaltet

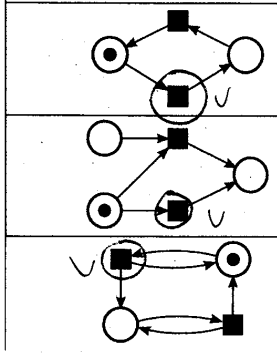
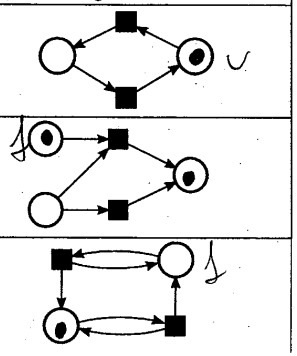
Stefanie

Aufgabe 2: Schaltregel (6p) 4/6

In den folgenden Petrinetzen ist jeweils eine Transition aktiviert.

a) Kreise in der linken Tabellenspalte in jedem Netz die aktivierte Transition ein.

b) Ergänze in der rechten Tabellenspalte die korrekten Marken nachdem die Transition geschaltet hat.

(a) Welche Transition ist aktiviert?	(b) Markierung nach dem Schalten:
	

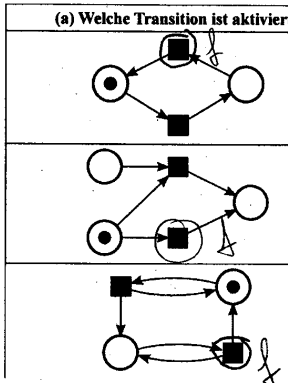
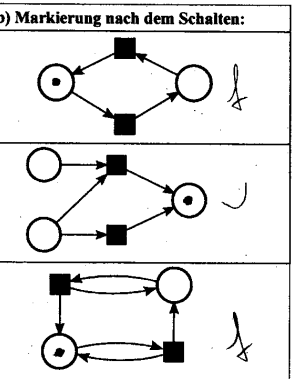
WPS

Aufgabe 2: Schaltregel (6p) 1/6

In den folgenden Petrinetzen ist jeweils eine Transition aktiviert.

a) Kreise in der linken Tabellenspalte in jedem Netz die aktivierte Transition ein.

b) Ergänze in der rechten Tabellenspalte die korrekten Marken nachdem die Transition geschaltet hat.

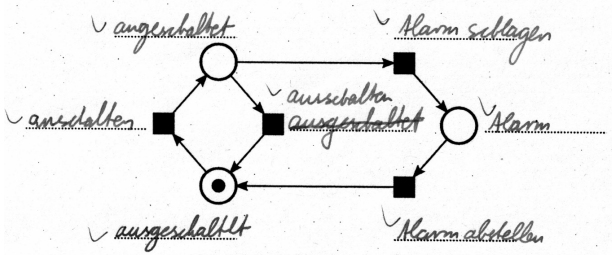
(a) Welche Transition ist aktiviert?	(b) Markierung nach dem Schalten:
	

WPS2

Aufgabe 3: Alarmanlage (4p) 3/4

Susie hat in ihrem Haus eine Alarmanlage. Wenn sie das Haus verlässt, schaltet sie die Alarmanlage an. Wenn sie zurückkommt, schaltet sie sie wieder aus. Einmal hatte Susie vergessen, ihren Kanarienvogel wieder in den Käfig zu sperren und als sie das Haus verließ, ging der Alarm los. Sie stellte den Alarm ab, sperrte den Vogel ein, und schaltete die Alarmanlage wieder an.

Das folgende Petrinetz modelliert die Alarmanlage von Susie.



a) Ordne den Plätzen und Transitionen die korrekten Bezeichnungen zu:
~~angeschaltet~~ | ~~ausgeschaltet~~ | ~~anschalten~~ | ~~ausschalten~~ | ~~Alarm~~ | ~~Alarm schlagen~~ | ~~Alarm abstellen~~

b) Bei Susie wurde eingebrochen, aber der Alarm ging nicht los, weil sie vergessen hatte, die Alarmanlage anzuschalten. Begründe anhand des Petrinetzes, warum der Alarm nicht starten konnte.

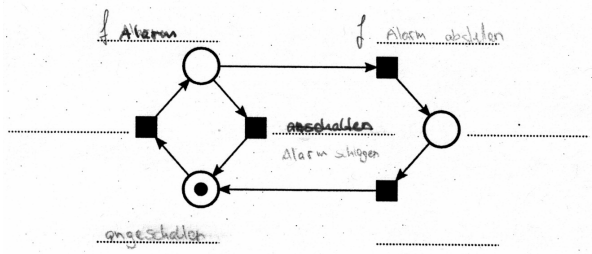
Weil sie ausgeschaltet war und sie vergessen hat sie wieder anzuschalten also bleibt sie aus. 1

WPS1

Aufgabe 3: Alarmanlage (4p) 0/4

Susie hat in ihrem Haus eine Alarmanlage. Wenn sie das Haus verlässt, schaltet sie die Alarmanlage an. Wenn sie zurückkommt, schaltet sie sie wieder aus. Einmal hatte Susie vergessen, ihren Kanarienvogel wieder in den Käfig zu sperren und als sie das Haus verließ, ging der Alarm los. Sie stellte den Alarm ab, sperrte den Vogel ein, und schaltete die Alarmanlage wieder an.

Das folgende Petrinetz modelliert die Alarmanlage von Susie.



a) Ordne den Plätzen und Transitionen die korrekten Bezeichnungen zu:
angeschaltet | ausgeschaltet | anschalten | ausschalten | Alarm | Alarm schlagen | Alarm abstellen

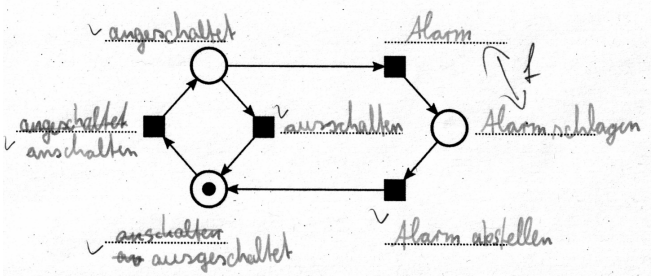
b) Bei Susie wurde eingebrochen, aber der Alarm ging nicht los, weil sie vergessen hatte, die Alarmanlage anzuschalten. Begründe anhand des Petrinetzes, warum der Alarm nicht starten konnte.

Stefanie

Aufgabe 3: Alarmanlage (4p) 2/4

Susie hat in ihrem Haus eine Alarmanlage. Wenn sie das Haus verlässt, schaltet sie die Alarmanlage an. Wenn sie zurückkommt, schaltet sie sie wieder aus. Einmal hatte Susie vergessen, ihren Kanarienvogel wieder in den Käfig zu sperren und als sie das Haus verließ, ging der Alarm los. Sie stellte den Alarm ab, sperrte den Vogel ein, und schaltete die Alarmanlage wieder an.

Das folgende Petrinetz modelliert die Alarmanlage von Susie.



a) Ordne den Plätzen und Transitionen die korrekten Bezeichnungen zu:
angeschaltet | ausgeschaltet | anschalten | ausschalten | Alarm | Alarm schlagen | Alarm abstellen

b) Bei Susie wurde eingebrochen, aber der Alarm ging nicht los, weil sie vergessen hatte, die Alarmanlage anzuschalten. Begründe anhand des Petrinetzes, warum der Alarm nicht starten konnte.

Die Alarmanlage konnte gar nicht „funktionieren, da die Marke auf „ausschalten" ist. Aktiv ist dann die Transition „Alarm abstellen". 1

B Nebenläufige Prozesse

B.1 Unterrichtsentwurf und -material

Bei der Lerngruppe handelt es sich um einen Informatik-Leistungskurs in der zwölften Jahrgangsstufe, bestehend aus einer Schülerin und fünf Schülern. Die Lernenden haben vor ungefähr einem halben Jahr, im Mai 2013, im Rahmen einer sogenannten Kompaktwoche bereits zwei Blöcke zu Petrinetzen gehabt. Sie haben die grundlegenden Modellierungskonzepte elementarer und gefärbter Netze kennengelernt und haben einen Fahrkartenautomaten sowie das Steinchenspiel von E.W. Dijkstra (vgl. Reisig 2010, S. 56-58) als Petrinetze modelliert. Vor allem letzteres haben die Lernenden mithilfe des Petrinetz-Modells analysiert und mittels systematischer Simulation herausgefunden, wovon die Farbe des letzten Spielsteins abhängt (vgl. ebd., S. 146-148). Die Lernenden wirken im Schnitt sehr leistungsstark, sodass ich davon ausgehe, dass sich die meisten sehr schnell wieder an die Funktionsweise von Petrinetzen erinnern werden.

Das Thema der 90-minütigen Unterrichtseinheit ist das Prinzip des wechselseitigen Ausschlusses, das System der speisenden Philosophen und das Problem der Verklemmung. Für eine Sachanalyse und Legitimation am Rahmenlehrplan siehe Abs. 3.2.3.

Der Unterricht beginnt mit einer kurzen Wiederholungsübung (AB1). Im Anschluss sollen die Lernenden das Prinzip des wechselseitigen Ausschlusses kennenlernenen, indem sie ein entsprechendes Petrinetz Modell (basierend auf ebd., S. 38) analysieren und weitere alltägliche und technische Beispiele für einen wechselseitigen Ausschluss finden (AB2, Aufg. 1 u. 2).

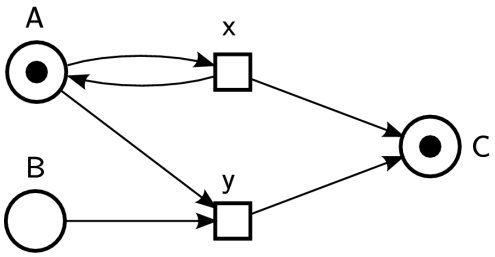
Im zweiten Teil des Blocks sollen die Lernenden das System der speisenden Philosophen zunächst selbstständig modellieren und anschließend das Problem der Verklemmung mittels einer Simulation identifizieren. Sollten die von den Lernenden erstellten Netze unvollständig oder fehlerhaft sein, wird ihnen für die Simulation und Analyse ein korrektes Netz zur Verfügung gestellt. Ist das Problem der Verklemmung erkannt, sollen die Lernenden das Modell so verändern, dass es vermieden wird (AB3).

Findet dabei bereits jemand die Lösung, bei der die Philosophen ihre beiden Essstäbchen gleichzeitig, d.h. atomar, aufnehmen, wird das Prinzip der Atomarität in der anschließenden Auswertung bereits thematisiert. In diesem Fall sollen sich die Lernenden zum Abschluss der Stunde noch einmal mit dem Problem der Fairness auseinandersetzten (AB2, Aufg. 3) und erkennen, dass sich die beiden Prozesse unfair behandeln können.

Hat noch niemand das Prinzip der Atomarität selbst umgesetzt, wird den Lernenden stattdessen ein Modell der Philosphen zur Analyse gegeben, in dem

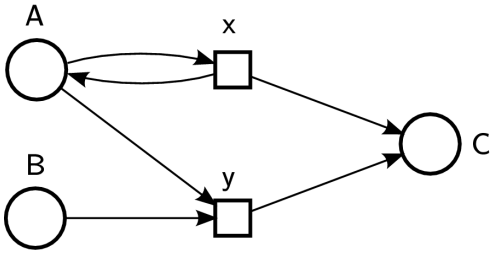
eine Verklemmung mittels atomarer Operationen vermieden wird. Die Lernenden sollen das Prinzip in dem Modell erkennen und beschreiben.

Petrinetze – Wiederholung (AB1)



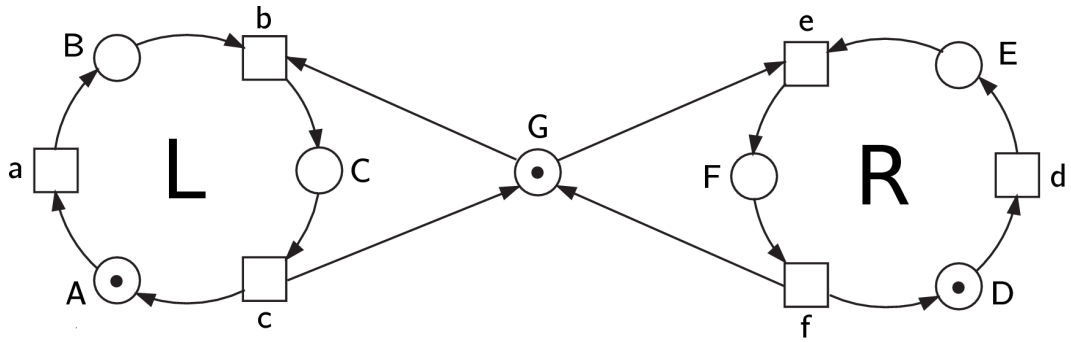
1. Kennzeichne in dem obigen Petrinetz die **Plätze**, **Transitionen**, **Kanten** und **Marken**.

2. Zeichne die Markierung des Netzes, nachdem die Transition **x** geschaltet hat:



3. Begründe, warum die Transition **y** nicht aktiviert ist, also nicht schalten kann.

Wechselseitiger Ausschluss (AB2)



Dieses Petrinetz modelliert zwei Bewohner einer WG (**L**aura und **R**alf), die sich eine gemeinsame Waschmaschine teilen.

1. Ordne den Plätzen des Netzes die folgenden Bezeichnungen zu: Waschmaschine ist frei | Laura hat saubere Wäsche | Ralf hat saubere Wäsche | Laura will Wäsche waschen | Ralf will Wäsche waschen | Laura wäscht Wäsche | Ralf wäscht Wäsche

   A _____ B _____

   C _____ D _____

   E _____ F _____

   G _____

2. Auf den Plätzen C, F und G zusammen liegt immer genau eine Marke. Anders ausgedrückt, gilt in dem Netz die Gleichung C + F + G = 1.

   a) Was bedeutet diese Gleichung für die Benutzung der Waschmaschine?

   _____

   _____

   b) Erkläre das Prinzip des wechselseitigen Ausschlusses im Allgemeinen.

   _____

   _____

   _____

   c) Nenne ein weiteres alltägliches und ein technisches Beispiel für einen wechselseitigen Ausschluss.

   _____

   _____

3. Ein Prozess ist nicht verpflichtet, die knappe Ressource zu benutzen. Wenn er sie aber benutzen will, dann sollte er auch irgendwann an die Reihe kommen, sie zu benutzen. Ist das in dem Petrinetz garantiert? Begründe.

   _____

   _____

   _____

   _____

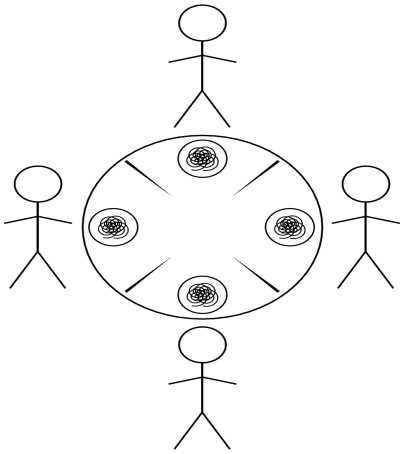
   _____

   _____

Die speisenden Philosophen (AB3)

Vier Philosophen sitzen um einem runden Tisch herum. Jeder Philosoph hat einen riesigen Teller Nudeln vor sich. Um die Nudeln zu essen benötigt ein Philosoph zwei Essstäbchen. Alles andere wäre Barbarei! Leider befinden sich im Haushalt der Philosophen insgesamt nur noch vier Essstäbchen, die nun zwischen den Tellern liegen. Ein Philosoph kann also nicht gleichzeitig mit seinen beiden Nachbarn essen.

Die meiste Zeit des Tages will ein Philosoph auch gar nicht essen. Er sitzt einfach da und denkt nach. Bekommt er dann doch irgendwann Hunger, greift er gedankenversunken neben sich, erst nach seinem linken und dann nach seinem rechten Essstäbchen. Ist ein Essstäbchen gerade in Benutzung, wartet der Philosoph solange, bis es wieder frei wird. Erst wenn er beide Stäbchen hat, isst er etwas von den Nudeln. Danach legt er beide Stäbchen gleichzeitig zurück und macht mit dem Denken weiter – solange, bis er wieder hungrig wird...



Arbeite zusammen mit einer Partnerin oder einem Partner.

1. **Modellierung:** Öffnet das Programm *Versys* und erstellt ein Petrinetz-Modell für das oben beschriebene Verhalten der vier speisenden Philosophen. (*Tipp*: Benutzt farbige Marken, um die verschiedenen Essstäbchen darzustellen.)

2. **Analyse:** Führt ein paar automatische Simulationen durch ( ). Beschreibt das Problem, das dabei auftritt.

   _____

   _____

   _____

   _____

   _____

   _____

3. **Problemlösung:** Verändert in *Versys* das Verhalten *eines* der Philosphen, sodass das Problem vermieden wird.

   a) Beschreibt eure Lösung. _____

   _____

   _____

   _____

   _____

   _____

   b) Findet ihr weitere Lösungsansätze? _____

   _____

   _____

   _____

   _____

Verlaufsplan

Zeit	Phase	Lernsituation	Medien, Sozialform
10 min	Einstieg	Begrüßung. Kurze Wiederholung zu Petrinetzen.	AB1, EA
15 min	Arbeitsphase	SuS bearbeiten Aufg. 1 und 2 zum wechselseitigen Ausschluss.	AB2, EA
10 min	E-Sicherung	SuS stellen ihre Ergebnisse vor. **Fazit:** Ein gemeinsame knappe Ressource verlangt, dass die Prozesse aufeinander warten müssen.	Plenum
15 min	Arbeitsphase	Wechsel an die Rechner. SuS bearbeiten Aufg. 1 zu den speisenden Philosophen.	AB3, *Versys*, PA
10 min	E-Sicherung	1-2 SuS stellen ihre Modelle vor. L: „Diejenigen, die jetzt noch kein funktionierendes Petrinetz haben, können sich eins vom Transferlaufwerk kopieren und damit weiterarbeiten.“	Plenum
10 min	Arbeitsphase	SuS bearbeiten Aufg. 2 und ggf. 3 zu den speisenden Philosphen.	AB3, *Versys*, PA
10 min	E-SIcherung	SuS beschreiben das Problem der Verklemmung. **Fazit:** Wenn Prozesse gegenseitig aufeinander warten, kommt es zu einer Verklemmung. SuS stellen ggf. eine Lösung für das Problem vor. **Fazit:** Um eine Verklemmung zu verhindern, müssen die Prozesse kommunizieren oder atomare Operationen verwenden.	Plenum
5 min	Arbeitsphase	Wenn eine Schülerlösung bereits atomare Operationen beinhaltete: Aufg. 3 zum wechselseitigen Ausschluss. **Fazit:** Prozesse können unfair zueinander sein. Sonst: Analyse einer Variante des Netzes ohne Verklemmung. **Fazit:** Um ein Deadlock zu verhindern, müssen die Prozesse kommunizieren oder atomare Operationen verwenden.	AB2, EA *Versys*, EA
5 min	E-Sicherung	Vergleichen bzw. Lösungen vostellen.	Plenum

B.2 Beobachtung

B.2.1 Protokoll

Datum: 08.11.2013 | Beginn: 9:50 Uhr Uhr | Jahrgangsstufe: 12 (Leistungskurs)

[0'00] L: „Wir knüpfen heute an das an, was wir in der Kompaktwoche gemacht haben." Arbeitsblatt zur Wiederholung wird ausgeteilt. SuS bearbeiten Arbeitsblatt. Ruhig. Keine Rückfragen.

[0'03] Vergleich der Wiederholung. Verschiedene SuS geben korrekte Antworten zu den Aufgaben. Saskia hatte keine Lösung zur zweiten Aufgabe: „Das hab' ich nicht." L: „Eine Transition kann schalten, wenn was gegeben ist? Ganz allgemein?"

[0'09] Merksatz an der Tafel: „Eine Transition schaltet, indem sie von jedem Platz vor ihr je eine Marke nimmt und auf jeden Platz nach ihr je eine Marke legt." (Unterstrichenes bezeichnet Lücken, die von den SuS vervollständigt wurden.)

[0'11] Arbeitsblatt wird ausgeteilt. SuS bearbeiten Aufgaben 1 und 2 zum wechselseitigen Ausschluss. Ruhig. Keine Rückfragen.

[0'19] L: „Kommt mal zum Abschluss. Das erste, was ich bei allen gesehen habe, ist G." Verschiedene SuS nennen die korrekten Platzbezeichnungen. Auf die Frage, wer alle Bezichnungen richtig hatte, melden sich drei SuS. Tom gibt Antwort zu Aufgabe 2: „Entweder einer wäscht oder keiner wäscht. Es kann immer nur einer die Marke haben." Richard hat eine Frage: „Wenn das so, wie es hier dargestellt ist, gleichzeitig abläuft, dann kommen auf F und C eine Marke." L fragt nach weiterer Explikation. Richard: „Was ist, wenn sie das gleichzeitig abarbeiten?" L schließt Gleichzeitigkeit aus: „Wenn wir an Prozesse im Computer denken, sind die zeitlich getaktet."

[0'23] Richard beschreibt Mutex: „Wechsel von zwei Methoden, die sich nicht überlagern."

[0'27] L: „Was wird hier verwendet? Direkt im Sachzusammenhang?" Merksatz an der Tafel: „Ein wechselseitiger Ausschluss bedeutet, dass die Waschmaschine nur von maximal einer Person verwendet werden kann." Begriff an der Tafel: „Gemeinsame Ressource."

[0'29] Vergleich der Aufgabe 2c. Tom: „Wenn man in der WG duschen will." Tom: „Fahrkartenautomaten auf dem Bahnsteig, wenn nur einer da ist." L: „Ja, das ist aus dem Alltag. Und wenn wir jetzt mal an die Technik denken?" SuS geben keine Antwort. L beschreibt Beispiel aus der Robotik, bei dem zwei parallele Prozesse in verschiedenen Situationen die Bewegung des Roboters steuern: „Was ist da die gemeinsame Ressource?" Tom: „Der Prozessor." L: „Nein." Marco nennt Zugriff auf die Motoren. Richard: „Wäre das nicht eine If-Else-Verzweigung?" L: „Ja, aber dann müsstest du die ganze Zeit immerzu gucken, ob du jetzt das eine oder das andere machst. Einfacher ist es, wenn du zwei parallele Prozesse machst." L versucht auf weiteres Beispiel zu lenken: „Welche gemeinsame Ressource brauchen alle Programme, die auf einem Rechner laufen?" Tom: „Das Rechenwerk?" L: „Genau."

[0'34] L: „Dreht mal die Seite um." SuS lesen Aufgabenstellung zu den vier Philosophen. L klärt Verständnis der Beschreibung. SuS fangen an, das Problem zu analysieren. „Das heißt, dass einer erst essen kann, wenn ... ne ..." „Dann kann ja..." Aussagen bleiben unvollständig.

[0'37] SuS wechseln an die Rechnerplätze. L: „Das liegt auf dem Vorlagen-Ordner." Tom: „Das funktioniert gar nicht. Wenn alle am Anfang Hunger haben und nach ihren linken Stäbchen greifen, kann keiner essen."

[0'42] SuS modellieren alle einzelne Stäbchenplätze. Niemand modelliert den Tisch als einen einzelnen Platz.

Tom dreht das Arbeitsblatt um und vergleicht offenbar den wechselseitigen Ausschluss mit den vier Philosophen. [0'46]

Richard hat einen Philosophen richtig modelliert. Modell verwendet nur schwarze Marken. Tom zu L: „Wie kann ich ihm hier sagen, dass er nur auf rote reagieren soll?“ Es geht um Transitionen und Marken in Versys. L: „Kante anklicken. Die Kante gibt die Bedingung an.“ Darstellungsfehler in Versys: orangene Marken werden an Kanten rot dargestellt. In Plätzen ist die Darstellung korrekt. [0'51]

SuS sollen die Arbeit beenden und die Netze auf dem Transferlaufwerk speichern. [1'02]

Richard stellt Netz vor: „Ich habe das nur für zwei Philosophen erst gemacht.“ Vorgestellte Lösung ist für die zwei Philosophen korrekt. L-Rechner stürzt mit Blue Screen ab. Wechsel an einen S-Rechner. Tom stellt Lösung vor, welche vollständig und korrekt ist. Richard meint jedoch, das Modell wäre noch nicht fertig. Auf die Frage, ob es ein Zeitproblem war: „Ne, er macht das nicht. Er legt die Marken da nicht rein.“ Es geht um Versys. Nils stellt Lösung vor. Alle Philosophen nehmen die Stäbchen korrekt auf. Es fehlen Transitionen zum Zurücklegen. Einige SuS beginnen, sich darüber zu unterhalten, welche Philosophen-Zustände in dem Modell vorkommen müssen: Essen-Denken oder Satt-Hungrig-Warten-Essen. [1'04]

Alle SuS arbeiten mit der Musterlösung vom Transferlaufwerk weiter. Bearbeiten Aufgaben 2 und 3 zu den speisenden Philosophen. [1'14]

Tom hat die Aktionen eines Philosophen atomarisiert. Richard: „... beide gleichzeitig aufnehmen.“ Nils modelliert das Zurücklegen des linken Stäbchens. [1'16]

L zu mir: „Die sind gut dabei. Ich fühl mich wie in einer Vorführstunde. Dieses Auf-Den-Punkt-Kommen, das mache ich sonst nicht so.“ [1'20]

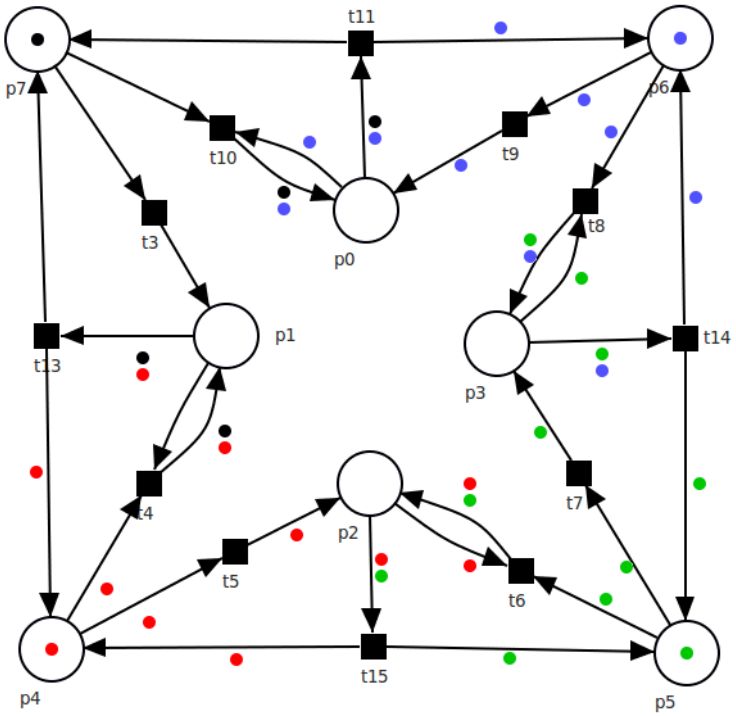
L: „Kommt mal so langsam zum Abschluss, egal, ob das jetzt schon zum Erfolg geführt hat oder nicht.“ SuS beenden ihre Arbeit am Rechner. L: „Dann kommt in der Mitte wieder zusammen.“ Tom beschreibt das Problem der Verklemmung: „Wenn alle Philosophen je ein Essstäbchen in der Hand haben, dann warten sie und es geht nicht weiter.“ L: „Woran sehen wir das im Petrinetz?“ Tom: „Keine Transition kann angeklickt werden.“ L: „Was habt ihr für Lösungen gefunden?“ Tom beschreibt die Lösung, bei der ein Philosoph seine Stäbchen gleichzeitig aufnimmt. Nils beschreibt das Zurücklegen des ersten Stäbchens: „Das Problem dabei war, dass der Philosoph fast nie gegessen hat.“ Lars beschreibt ein Problem mit den Farben der Marken, nach dem Laden der Datei. L: „OK, da gab es also noch technische Probleme.“ [1'21]

L zu Atomarität: „Wo hatten wir Atomarität schon einmal?“ Nils: „Datenbanken.“ L beschreibt das Prinzip der Atomarität mit dem Begriff „unteilbar.“ L gibt kurze Aussicht auf die nächste Stunde: „Da werden wir nächstes Mal noch einmal anknüpfen und das dann abschließen.“ [1'28]

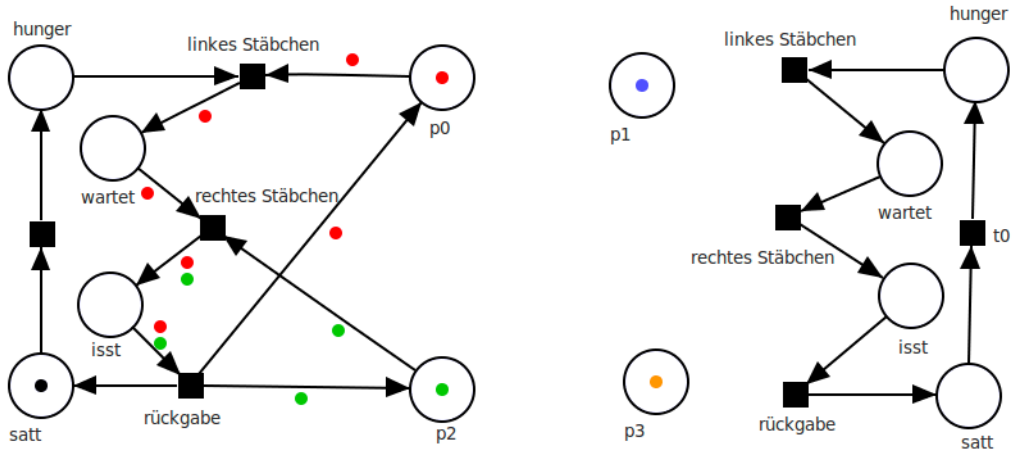
Verabschiedung. [1'30]

B.2.2 Exemplarische Schülernetze

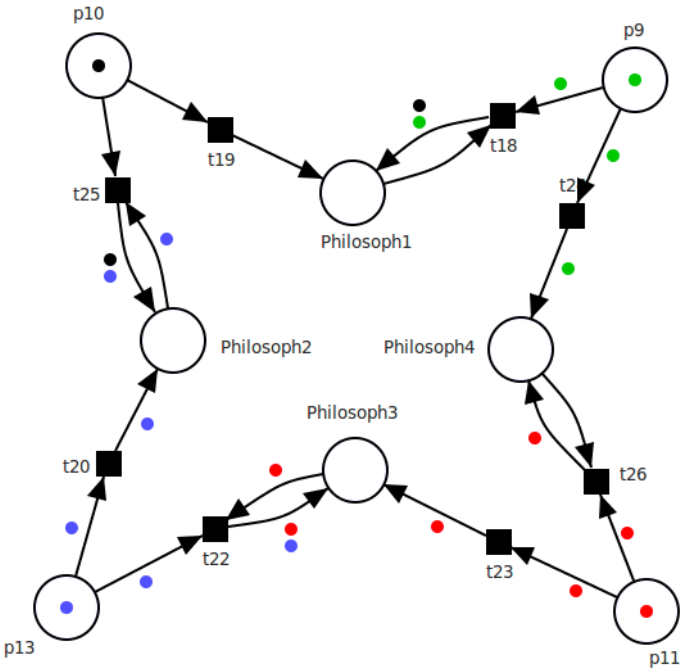
Die Netze wurden zur besseren Les- und Vergleichbarkeit anders angeordnet. Die Topologien, Markierungen und Beschriftungen sind aber in jedem Fall unverändert.



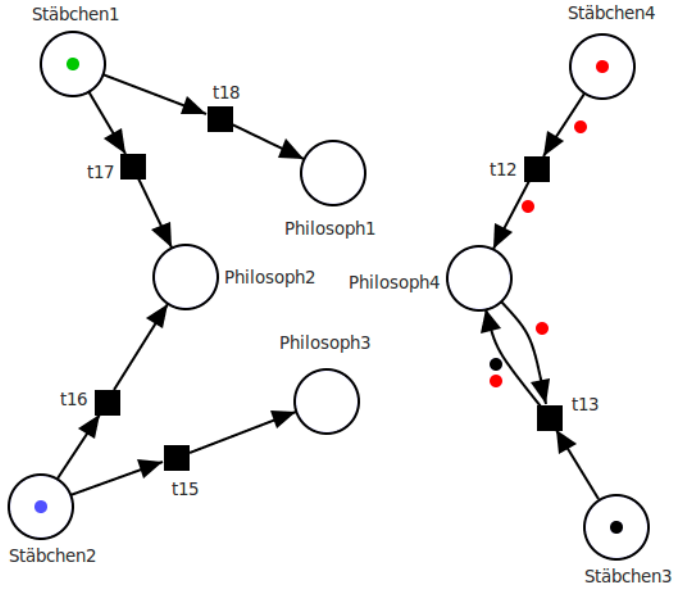
Tom (Philosophen innen, Essstäbchen außen. Die Schlingen bewirken, dass die Phil. ihre Esstäbchen nur in einer festgelegten Reihenfolge aufnehmen können.)



Saskia (Ein Philosoph korrekt modelliert.)



Nils (Philosophen innen, Essstäbchen außen. Das Aufnehmen der Stäbchen ist korrekt modelliert. Das Zurücklegen fehlt.)



Marco (Das Aufnehmen der Stäbchen ist für einen Philosophen korrekt modelliert.)

B.3 Interview

B.3.1 Leitfaden

Es wurde derselbe Leitfaden wie in Anhang A.3.1 verwendet.

B.3.2 Transkript

[00:00] **I:** Gut. Dann weise ich dich trotzdem jetzt einfach nochmal ganz offiziell für das Protokoll darauf hin, dass ich das Interview gerne mitschneiden würde. Bist du damit einverstanden?

B: Ja.

I: Sehr gut. Das Transkript werde ich dann im Anschluss natürlich anonymisieren. Ansonsten ganz kurz noch einmal zwei Worte zu dem Sinn und Zweck dieses Interviews. Ich hätte grundlegend gerne eine Einschätzung von dir jetzt zu den gehaltenen Stunden zum wechselseitigen Ausschluss und den Philosophen. Das heißt, in erster Linie die Unterrichtsinhalte, die Aufgaben und wie die Schüler damit umgegangen sind. Das heißt, es geht mir NICHT darum, DICH irgendwie zu bewerten. Du darfst und SOLLST natürlich auch gerne selbstkritisch sein, an Situationen, wo du das für sinnvoll hälst, aber es geht mir nicht darum am Ende irgendwie dich zu bewerten, sondern darum, die Schüleraktivitäten zu bewerten.

Gut, das vorweg. Dann hätte ich zu Beginn gerne, dass du die gehaltene Stunde einfach mal aus deiner Sicht rekapitulierst. Hier ist nochmal der Verlaufsplan, den wir vorher geplant hatten. Gehe die Stunde doch bitte einfach nochmal gedanklich durch. Was lief gut? Was lief nicht gut? Was war auffällig?

[01:23] **B:** Gut. Wir haben angefangen mit den Petrinetzen, mit der Wiederholung. Da war ich sehr erstaunt, wie schnell die Schüler damit umgegangen sind und, dass sie, obwohl wir das ja überhaupt nicht angekündigt haben, teilweise im Hefter geblättert haben. Also wenigstens Saskia hat versucht, die Unterlagen von der Kompaktwoche wiederzufinden und sich daran zu orientieren. Die meisten haben das aber ohne Unterlagen hinbekommen. Also das war eine sehr schöne wiederholende Übung. Und ist im Prinzip ja auch, (3) wenn ich mich richtig erinnere, von fünfen vollständig, naja da bin ich mir nicht ganz sicher, ob es vollständig war, von allen. Aber auf jeden Fall bei dem Zusammentragen haben also die fünf Jungs das alle sofort nachvollzogen wieder und verstanden und ergänzt. Und es hat einen guten Anlass geboten, das nochmal zu erklären, weil Saskia konnte sich nämlich nicht mehr so richtig erinnern. Und der hat die Erklärung da gereicht. Ich habe dann den Satz an die Tafel geschrieben, den wir auch aus dem Wahlpflichtunterricht als Merksatz hatten. So ungefähr. Nur ein Teil davon. Da hat man gemerkt, wie gut das geholfen hat, diese Visualisierung, diesen Merksatz nochmal als Erinnerung mit aufzuschreiben. Der muss hier in der Aufgabe nicht drin vorkommen, aber das war gut, denke ich, dass das da/ An der Stelle habe ich gemerkt, aha, jetzt hat Saskia es auch gehabt. Dieses mit dem vorher und nachher, weil das haben wir damals ja nicht gemacht. (5) Und es ging damit auch sehr schön schnell. Da haben wir die Zeit auch komplett eingehalten, diese Wiederholungsphase mit Besprechen, da waren die 10 Minuten auch wirklich gerade um. (2)

[03:07] Dann die erste Arbeitsphase. Da ging es um die Analyse des wechselseitigen Ausschlusses. (3) Das war im ersten Moment schon ganz schön anspruchsvoll. Also, an den Gesichtsausdrücken hat man das gesehen. Wenn ich Nils mir angeguckt habe. Die haben nicht sofort losgelegt und da etwas hingeschrieben, sondern das war schon richtig eine Weile knobeln. Aber es war machbar, dass jeder das alleine hinbekommen hat. Die haben das NICHT für nötig befunden,

sich dazu auszutauschen. Das haben sie erst gemacht als sie/ (2) Da gehe ich mal andersrum. Also Aufgabe 1 haben sie alle geknackt. Saskia mit ein paar kleinen Fehlern, aber sie haben es geknackt. Spannend fand ich: sie haben alle mit der Waschmaschine angefangen. Sie haben alle „G“ als erstes erkannt. Auch Saskia. Und dann war es teilweise auch nur eine Ungenauigkeit der Zuordnung bei ihr. Sie haben dann auch bei Aufgabe 2a die meisten noch etwas stehen gehabt. Und bei Aufgabe 2b nur noch ansatzweise und da fing dann so eine Unruhe an. An der Stelle war es schwer und an der Stelle sind sie ausgestiegen. Da kam dann nachher im Unterrichtsgespräch auch nicht mehr viel. Das heißt, dass ich da nach 10 Minuten unterbrochen habe, obwohl wir ja 15 Minuten geplant haben, war deutlich, da fand nicht mehr viel statt in der Arbeitsphase. Da haben die sich ausgetauscht, da hatten sie also alles, was sie als Ideen hatten wirklich runtergeschrieben. Das heißt, zur Aufgabe 2c ist ihnen wirklich alleine nichts bzw. fast nichts eingefallen. Ich habe jetzt noch nicht darüber nachgedacht ob das an der Formulierung der Aufgabe lag oder an dem Problem, dass sie darüber noch nie nachgedacht hatten. Im Gespräch ist nachher, wenn ich jetzt bei Aufgabe 2c bin, dann auch nur noch Tom darauf gekommen, dass er eben die Badbenutzung im Prinzip von der Waschmaschine auf das Bad übertragen hat, aber viel mehr war da nicht mehr, kam da nicht mehr. Also das war tatsächlich, die 10 Minuten waren da ausreichend und die Aufgaben waren so, dass sie das auch (.) gut meistern konnten. Bis auf diese Anwendung. Und das ist ja immer die schwerste Sache. Ich meine, sie haben es gerade kennengelernt und das dann sofort auf einen anderen Bereich zu übertragen war vielleicht auch ein bisschen viel an der Stelle gefordert. So. Das war der wechselseitige Ausschluss. Insgesamt war das Material wirklich so, dass man es gut machen konnte und die Schüler gut selbständig damit arbeiten konnten also. Das Vergleichen ging dann auch ganz gut. Ich habe mich dann ein bisschen verhaspelt da bei dem wechselseitigen Ausschluss da. Aber das/ Wenn man da ein bisschen mehr, bisschen sicherer im Stoff steht, passiert einem das nicht. Also, das hat dann ein bisschen Zeit gekostet, aber (4) war auch nicht so dramatisch. Im Endeffekt war es auch für die Schüler nicht schlimm, dass ich mehr oder weniger den Satz dann abgeschrieben habe und deutlich geworden ist, dass ich da auch ein bisschen geschwommen bin, aber das/ (2) Sie sind ja so, dass sie einem das verzeihen. (2) Gut.

Die speisenden Philosophen. (3) Da haben wir erstmal gemeinsam gelesen. Da bin ich ein [06:09] bisschen abgewichen von der Planung. Das ist mir eigentlich erst beim Austeilen aufgefallen, dass es gut ist, erstmal den Text zu lesen. Und das war ja auch zu merken. Es gab ja Nachfragen, wie man jetzt das wirklich genau interpretieren soll. Ob der jetzt wirklich den linken Stab immer in der Hand behält und wartet bis er den rechten findet, oder zurücklegt. Das mag zwar eindeutig darin stehen, aber es ist ja immer die zweite Sache, es zu glauben, dass es wirklich so hundertprozentig gemeint ist, wie es da steht. Und der Austausch, denke ich, ist da wichtig gewesen. (2) Interessant fand ich, dass sie an der Stelle schon gespürt haben, dass da irgendwas komisch ist, in diesem Beispiel. Also eigentlich, die Philosophen sind ja eine schöne Veranschaulichung. Und die sind schon so, dass auch ein Leistungskursschüler das sogar aus dem Spiel heraus, ohne das überhaupt zu modellieren, erkennt, dass da irgendwas, irgendwie das nicht gut ist, dass sie das so modellieren, dass immer alle die linke Gabel nehmen und warten bis sie die rechte kriegen. Dass das irgendwie komisch wird. Aber es war sehr schön zu merken (.) es fiel Tom schwer, das zu formulieren (.) an der Stelle. Das heißt, (3) da fehlten ihm die Worte und zum Schluss fiel es ihm ganz leicht. Das zeigt wiederum, dass dieses Hilfsmittel des Petrinetzes, dieses Modellieren und diese Auseinandersetzung damit ihnen nachher die Worte gegeben hat. Und das fand ich sehr spannend zu sehen. Ich habe schon einen Schreck gekriegt, in dem Moment als er das so erkannt hat. Da hab ich gedacht, so dieses (lachend) typische Beispiel von Lehrer: „Ja, darüber wollen wir jetzt noch nicht sprechen, das musst du jetzt noch nicht erkennen,“ aber er hat es erkannt, aber er konnte es nicht formulieren und das fand ich sehr schön. Also der hat genau gewusst, wo es hingeht und die anderen haben das auch gespürt,

aber es war nicht so, dass sie das sofort hingeschrieben hätten an der Stelle. (.) Ja. (.)

[08:04] Dann sind wir in die Arbeitsphase gegangen. (...) Da haben sie dann (.) alle (..) das mit den Stäbchen modelliert (..) aber sie / (4) Da müsstest du mir nochmal helfen. Richard hat sich wirklich an dem orientiert, was auf der anderen Seite stand. Der hat also die beiden/ Den wechselseitigen

I: Richard war hier hinten in der Ecke?

B: Hinten. Richard, Saskia.

I: Ach, Richard, Saskia. Ok.

B: Die sind die einzigen gewesen, die sozusagen, die verschiedenen Plätze so nachgebaut haben, wie wir es in der Waschmaschine hatten. (3) Die anderen haben/ (2) Haben die den wechselseitigen Ausschluss drin gehabt? Hatten sie drin, nur die haben nicht die Zwischenstufen drin gehabt, ne? Deswegen sah das so anders aus. Ja.

I: Genau.

B: Wäre das vollkommen korrekt gewesen? Da war ich mir nicht ganz sicher.

I: Also rein fachlich, ja, klar.

B: Das ist nur ein bisschen vereinfacht gewesen.

I: Die drei Lösungen, die jetzt am Ende vorgestellt wurden, waren zwar alle unvollständig, aber von der Sache her/

B: Fachlich richtig.

I: Fachlich richtig, genau. Also wenn das vollständig gewesen wäre, wäre das ein vollkommen korrektes Modell für die Philosophen gewesen.

[09:14] **B:** Gut. Das Gefühl hatte ich auch, aber da war ich ein bisschen unsicher. Gut. (blättert im Material) (9) Genau. Aber es war auf jeden Fall gut und wichtig, dass wir das fertige Modell hatten. (1) Dass sie damit experimentieren konnten, denn es gab ja auch noch Probleme, dass sie teilweise einfach Dinge übersehen hatten oder irgendwelche Kanten noch nicht so modellieren konnten. Das war/ Lag vielleicht teilweise in dem Programm, dass sie da irgendwas nicht hinbekommen haben, oder irgendwas komisch war, aber es lag eben auch daran, dass es so, wie sie es modelliert haben, relativ komplex war. Weil sie eben nicht erkannt haben, dass es alles auf den Tisch legen (2) können. Alles auf EINEN Tisch legen können. Du hast mir den Hinweis gegeben in der Stunde, dass ich danach mal fragen soll, aber ich glaube, wenn ich das gemacht hätte, hätte ich sie verwirrt. Weil sie nicht an der Stelle gesessen haben, sondern dieses alles auf einen Tisch zu packen, auf einen Platz zu packen, ist etwas, was ich erst richtig erfasse, wenn ich das andere durchdacht habe. Das ist ein höherer Abstraktionsgrad.

I: Ja.

B: Und deswegen habe ich da mal geguckt, um zu verstehen, was die da machen, bis ich gesehen habe, dass die diese einzelnen Plätze für jede Gabel sozusagen oder für jedes Stäbchen einen Platz verwendet haben. Und das ist ja auch das, glaube ich, was ich auch modelliert hatte, als ich das probiert habe.

I: Weiß ich jetzt nicht mehr. Kann sein, ja.

B: Es war mir jedenfalls vollkommen nachvollziehbar, dass sie das als erstes probieren und da wollte ich sie auch nicht von wegholen. Sonst hätten sie, glaube ich, das nicht so als Ergebnis erreicht. (3)

So. Das Testen war dann nachher letztendlich nur nochmal ein Überzeugen davon, was sie im Prinzip schon vermutet haben. Und diese Analyse, diese zweite Aufgabe, war eigentlich jetzt dazu da wirklich das Problem zu formulieren und jetzt war es MÖGLICH, das Problem zu formulieren. Schade eigentlich, dass mir das jetzt erst so richtig bewusst wird. Man hätte an der Stelle nochmal darauf, bei dem Vergleichen, darauf zurück kommen können und Tom daran erinnern können, dass er eigentlich diesen Gedankengang am Anfang schon hatte. Aber eigentlich jetzt erst es ihm richtig leicht fiel, das zu FORmulieren. Da habe ich nicht dran gedacht, weil mir es erst jetzt im Nachhinein auffällt, aber das ist tatsächlich eine Reflexion, die man hier noch hätte bringen können, um nochmal deutlich zu machen, wie viel so eine Visualisierung, so eine Modellierung auf dieser Ebene tatsächlich bringt. (3) [10:54]

Und zur Problemlösung, war ja ganz genial. Sie haben ja wirklich zwei verschiedene Varianten gefunden und dass sogar einer das mit der Atomarität ausprobiert hat, hatte ich vorher nicht gesehen, aber das war (wirklich?) dadurch sehr schön rund und sie haben ja sogar auch noch die Atomarität in den Datenbanken gebracht. Klar, die haben ja gerade Klausur geschrieben. (unv., lacht) wiederholen, obwohl es da nicht vorkam. Also da (2) haben sie das gleich nochmal, dieses große informatische Konzept, oder sich an verschiedenen Stellen wiederholende Konzept der Atomarität (2) es ist ja vielleicht kein großes, aber es ist zumindest eins, was wiederkommt, so ein roter Faden, eben auch erkannt. Und das (.) war gut. Also, bin ich sehr zufrieden mit. Und ich habe auch das Gefühl gehabt, dass es für die Schüler rund war. (3) [11:49]

Es war für die Schüler rund. Es war für die Schüler interessant. So, jetzt bin ich sozusagen durch die Stunde durch. Jetzt reflektiere ich gerade nochmal so allgemein. Es war für die Schüler interessant. Es hat sich im Nachhinein für die Schüler gezeigt, dass das, was du damals mit ihnen gemacht hattest, dass das also irgendwie nicht nur so eine Eintagsfliege war, sondern, dass das ein größeres Konzept ist. Und sie haben hier ein paar Fachbegriffe gehört, die sie/ (..) Die ihnen zumindest signalisieren, dass da schon noch ein großes Konzept hinter steht. Dass es also/ Wenn man einen Fachbegriff einer Sache zuordnet, dann ist es ja meistens nicht nur so, dass das mal eine Kleinigkeit am Rande ist, sondern dass es zumindest irgendwie, dass da deutlich noch mehr dahinter steht. Und dieser Ausblick, denke ich, ist schon deutlich geworden und (.) das hat ihnen ja auch Spaß gemacht. Also sie waren ja nicht frustriert an irgendeiner Stelle mal, wenn mal irgendwas nicht gleich geklappt hat. Also sie waren ja sehr engagiert dabei, hatte ich den Eindruck. Also (.) muss ich sagen, würde ich wieder machen. [12:37]

I: OK.

B: Ja.

I: Dann eine kurze Zwischenfrage. Du hast gesagt, bei dem Mutex, am Anfang, als sie damit konfrontiert wurden, mit der ersten Aufgabe, dass sie da geknobelt haben, erstmal. Das hast du gerade so gesagt. [13:39]

B: Ja.

I: Was genau, denkst du, worüber haben sie da geknobelt?

B: Ich glaube, die haben erstmal/ Es hat sich für mich so angefühlt, so wie ich davor gesessen habe. Dass es einfach erstmal, boah!, so komisch aussieht. Also, die haben also gerade diese Marken hier gehabt und die Transitionen in der Wiederholung und jetzt mussten sie erstmal sich da reindenken. Und ich denke, die sind wirklich/ Die haben ja eine ganze Weile davor gesessen und das sah für mich so aus, als wenn die wirklich (.) versucht haben, da irgendwie jetzt/ Vielleicht sogar erstmal (..) zu gucken, was hier eigentlich passiert, was hier sich verändern könnte, was/ (.) Wenn man jetzt nicht so leistungsstarke Schüler hätte, hätte man vielleicht nochmal Hilfen geben können. Brauchte ich jetzt nicht, weil nachher haben sie ja irgendwann [14:00]

angefangen. Hilfen geben können, dass man mal, was wir hier machen, aktive Transitionen (.) so heißen die nicht. Wie haben wir gesagt, heißen die?

I: Doch.

B: Doch, aktive Transitionen.

I: Aktiv, also aktivierte. Mhm.

B: Aktivierte Transitionen finden und mal sich überlegen, wie sich das verändert, wenn man die jetzt schalten lassen würde. (.) Und ich denke, genau das haben sie gemacht.

I: Mhm. OK.

B: Allerdings nicht irgendwie jetzt auf dem Papier, sondern nur im Kopf, weil auf dem Papier ist es ja auch schwierig. Das Problem hatten wir. Weil es hat jetzt keiner einen Radiergummi genommen und die Marken durch die Gegend geschoben. Aber als Hilfestellung hätte man das ja im Prinzip geben können. (3)

[15:24] **I:** Gab es Situationen, in denen die Schülerinnen und Schüler über- oder unterfordert waren?

B: (11) Nein. (.)

I: Weder das eine, noch das andere? Gar nicht?

[15:45] **B:** Nein. Also, wir haben die Aufgabe 2c beim wechselseitigen Ausschluss gehabt. Da haben sie wirklich alle nichts geschrieben. Bei Aufgabe 2b hat nur Richard was geschrieben. Die anderen hatten da gar nichts und haben nur gegrübelt, was da wohl stehen würde. (.) Aber das ist ja normal so, dass man einen Block Aufgaben hat, so sollte ja so eine Lernaufgabe letztendlich ja auch sein, und man Stück für Stück sich durch die Aufgaben durcharbeitet und jeder an dem Punkt hängen bleibt, wo er jetzt erstmal nicht mehr weiter weiß. Das ist ja genau der Sinn. Deswegen würde ich an der Stelle, auch wenn sie die Aufgabe c alle nichts hatten, nicht sagen, dass die Schüler an der Stelle überfordert waren. Weil eine Überforderung wäre für mich, wenn sie mit einer Aufgabe gar nicht anfangen. Aber die haben ALLE losgelegt. Sie haben ALLE was verstanden. Sie haben ALLE Aufgabe 1 ausgefüllt und sich eine Vorstellung davon gemacht und sind dann weitergegangen. Saskia hat, glaube ich, zu Aufgabe 2a noch gar nichts gehabt. Die hat mit Aufgabe 1 vollauf zu tun gehabt, aber ist ja auch eher die schwächere. Und die anderen sind eben jetzt bei Aufgabe 2b stehen geblieben. Das ist ja das, wo ich gesagt habe, dass man überlegen müsste, ob man 2c jetzt drin lässt, oder ob man das vielleicht sogar in einer alternativen Stunde ans Ende setzt. Es war aber auch nicht störend, dass es da war, weil (.) letztendlich, die Beispiele, die wir dann noch besprochen haben, und sei es bloß das Bad, was wechselseitig benutzt werden konnte, hat ja auch schon hier gezeigt, das man was dort bringen kann und was erkennen kann und die haben ja auch dort mitgeschrieben, was wiederum auch ein guter Hinweis dafür ist, das da stehen zu lassen und dann eben auch, zur Not im Lehrervortrag, nochmal was ergänzt, worauf ich jetzt nicht so gut vorbereitet war, aber im Endeffekt ist ja was gekommen. Das mit dem Lego-Beispiel mit dem geradeaus und um die Ecke fahren hätte ich besser vorbereiten können noch und ihnen noch besser zeigen können. (.) Aber es hat auch so gereicht. Als Beispiel hat es gereicht. (.) und wie gesagt, sie haben ja mit geschrieben. Und das heißt, sie haben irgendwas davon verstanden, was sie auch notierenswert fanden an der Stelle. So das war das Einzige, wo sie mal etwas nicht gemacht haben.

[17:53] Und (.) bei dem Modellieren, (.) dadurch, dass wir gesagt haben Partnerarbeit (.) war also auch ganz klar, (..) dass sie eben auch wirklich nicht jeder allein da irgendwie jetzt was bringen musste, sondern irgendwie sie zu zweit da sich auf das berufen konnten, was auch der Partner auf seinem Rechner hatte und dadurch war auch dieses/ (.) diese Identifikation mit dem, was

sie mit den beiden Beispielen, die sie sozusagen zu zweit dort an ihren zwei Rechnern hatten, so da, dass sie auch das Gefühl hatten, dass sie da etwas zustande gebracht haben. Also auch da waren sie zufrieden mit dem, was sie modelliert haben. Sie haben ganz klar gesagt: „Ja, ich habe nur den einen Philosophen hingekriegt, aber das muss man jetzt ja bloß fortsetzen“ und das war vollkommen in Ordnung. Nein, also es gab keine Situation, wo sie sich ÜBERfordert gefühlt haben und es war auch KEINE Situation, wo sie in irgendeiner Form UNTERfordert und gelangweilt waren. Weil sie hatten immer gut zu tun. Selbst bei dieser so trivial erscheinenden Aufgabe (deutet auf das Arbeitblatt mit der Wiederholung). Die war absolut nötig, um sich daran zu erinnern, weil das ja jetzt auch ein halbes Jahr her ist. Und absolut unvorbereitet war. Wir haben ja nicht gesagt, sie sollen sich mal die Petrinetze nochmal angucken. War ja absichtlich so, dass sie nichtmal wussten, dass du heute da bist. (2)

I: Mhm. Ok. (3) Gab es Probleme beim Umgang mit Versys? Mit dem Programm? Damit meine ich eher gerade didaktische Probleme. Also hattest du an einer Stelle irgendwo den Eindruck, wo du gedacht/ dass dir das Programm gerade den Unterricht schwerer macht? [19:04]

B: Überhaupt nicht.

I: Überhaupt nicht?

B: Nein.

I: Wie, würdest du sagen, sind die Schüler mit dem Programm zurecht gekommen?

B: (3) Mit Ausnahme der kleinen Probleme, die es gab, weil da irgendwas farblich anders angezeigt wurde als es eigentlich sein sollte, also so ein paar kleine Bugs, die vielleicht hier und da noch drin sind, war das vollkommen intuitiv. Und sie haben ja auch damit schon einmal gearbeitet. Das heißt, das war (.) ja, intuitiv zu benutzen. Da haben sie nicht einmal gezögert an irgendeiner Stelle. (.) Ah naja, das heißt, Moment! Das eine war/ Sie wussten nicht/ (..) Oder, ich weiß nicht, ob alle, aber mir ist aufgefallen bei Nils und Marco, und zwar vorne in der Nähe der Tafel, die beiden, denen hab ich das nochmal gesagt, aber (unv.) (habe?) ich es erst gesagt, dass sie die Übergänge mit den farbigen Marken, dass sie da das an die Kanten ranschreiben müssen, wenn eine Marke genommen wird. Das weiß ich nicht mehr, ob das damals mit drin war. Das wäre vielleicht eine Sache, die in der Wiederholung auch hätte/ Entweder in der Wiederholung sein müssen, oder ich hätte es einfach nochmal sagen müssen. Die Information bei den beiden hat ja gereicht. Oder es hätte auf dem Blatt stehen müssen, bei dem Hinweis mit den farbigen Marken, dass man nochmal irgendwie (.) so ein farbiges Bild hat, wo sie sehen, wie das genau gemacht wird. Dass also die farbige Marke sozusagen nicht automatisch genommen wird. Weil die haben ja im Prinzip durch die verschiedenen einzelnen Plätze, die haben ja die farbigen Marken benutzt, aber sie brauchten sie ja eigentlich gar nicht. So, wie sie es modelliert haben, wären sie ja auch mit schwarzen Marken hingekommen. Und hatten aber jetzt das Problem, dass die farbigen Marken gar nicht gezogen wurden, weil sie ja mit einem schwarzen Pfeil gearbeitet haben, weil sie nichts eingestellt hatten. Also, da so ein kleiner Punkt, den wir vorher nicht bedacht hatten. (.) Der sich aber ganz leicht beheben lässt. Ansonsten ist es einfach sehr schön anschaulich. (...) [19:32]

I: Wenn du es ganz kompakt zusammenfassen müsstest. Was haben die Schülerinnen und Schüler jetzt aus den heutigen Stunden mitgenommen? [21:27]

B: (..) Ich fange mal von hinten an. (.) Sie haben die Atomarität als Prinzip zum zweiten Mal jetzt im Unterricht gehabt. Haben also gemerkt, dass das irgendwie was Größeres ist. Sie haben (...) wahrgenommen, erlebt, (.) dass Blockaden auftreten können, Verklemmungen. (..) Sie haben gemerkt, dass man dieses Problem lösen kann, (.) indem man zum Beispiel (.) über (.)

solche atomaren Operationen geht oder indem man andere Strategien einbaut, aber zumindest bewusst wahrnimmt, dass die Blockade eintreten kann und dagegen steuert. Zum Beispiel auch durch die Zeit. (..) Das haben sie als Problem wahrgenommen und auch durch diese aktive Lösung gesehen, dass es da Strategien gibt, das zu vermeiden. (..) Sie haben, noch ein Schritt davor, den wechselseitigen Ausschluss als allgemeines Konzept wahrgenommen, weil wir uns darüber eben auch unterhalten haben, wo es noch überall vorkommt. Sowohl im allgemeinen Leben als auch in der Technik. Verschiedene Ebenen. (.) Die Prozesse haben wir nur genannt. Wir haben das Lego genannt und wir haben das Bad genannt, also wirklich ganz verschiedene Ebenen, ganz allgemeingültiges Konzept. (...) Sie haben letztendlich mit einer Modellierungsart gearbeitet, (.) die ihnen geholfen hat, diese Probleme zu erfassen. Ob sie jetzt schon darüber reflektieren würden, dass sie das immer mit Petrinetzen machen würden (.) wage ich jetzt hier nicht zu behaupten. Weiß ich nicht. Aber auf jeden Fall ist ja die Arbeit mit einem Modell immer die Erfahrung, dass ich mit diesem Modell zum Ziel kam und damit, denke ich, auch ein Stück auf dem Weg sind wir gegangen.

[23:44] **I:** Dann direkt daran anschließend: wenn du jetzt einmal Nostradamus spielen müsstest, was glaubst du, wie wird der Test zu Beginn der nächsten Stunde ausfallen?

B: Lass mich mal nochmal in die Aufgaben gucken. (.) Die Testaufgaben. Hast du die mal da?

I: Ehm, ja, natürlich. (sucht Aufgaben heraus) Die hier.

[24:13] **B:** Genau. Das sind die.(6) Plätze, Transitionen, Marke, wenn schaltet, dann verschwindet, das können sie. Das sollten sie können und das (.) definitiv fünf von sechs. Saskia sollte das auch hinkriegen. Gehe ich eigentlich von aus. (...) Baustelle. (6) Kriegen fünf von sechs hin. (.) Die Modellierung mit dem Platz auf jeden Fall (.) Bei Saskia weiß ich nicht. (.) Ich kann schwer einschätzen, wie viel sie verstanden hat von dem, was sie mit Richard mitgebaut hat. Das ist nicht ganz eindeutig geworden. Weil sie ja auch von Anfang an diese Probleme hatte, das zu erfassen. Das ist einfach ein Stück weit/ Sie ist da an der Grenze, aber diese Grenze hat sie auch in ANDEREN informatischen Gebieten. Sie quält sich einfach mit allem. Das hat also/ Wenn sie diese Schwierigkeiten hier hat und nur Anforderungsbereich eins hinbekommt, nur diese begriffliche Geschichte, (.) dann (.) ist das aber nichts, wo ich jetzt sagen würde, das ist ein Mangel an dem Material, sondern diese Probleme hat sie mit allem. (.) Also sie hat oft in Klausuren nur (.) fünf Punkte, gerade eine Vier, sechs Punkte, manchmal nur vier Punkte, also das ist immer so, dass man die Punkte suchen muss, bei dem, was sie schreibt. Dass sie sich sehr viel müht, sehr viel lernt, (..) nun ist die Frage, ob sie lernt. Wahrscheinlich nicht, weil ich es ja nicht direkt als Test angesagt habe und weil die ja auch genug andere Klausuren im Moment schreiben, aber/

I: Ja, das werden wir ja dann sehen.

[26:10] **B:** Genau. Aber ich denke, Aufgabe Baustelle diese Baustelle zu modellieren müssten alle anderen hinbekommen. Die gemeinsam genutzte Fahrbahn sollten sie erkennen. (.) Würde ich jetzt erwarten. Die Straßenkreuzung. Mit der habe ich mich ja auch schwer getan. (lacht) (4) Dass das wieder so symmetrisch ist, (...) sollten alle erkennen. Die Frage/ Also, wer jetzt da irgendwas anfängt, denke ich, fügt die Symmetrie ein. Das erwarte ich eigentlich bei allen. Ob sie anfangen ist die Frage. Also, ob sie diese Hürde nehmen, dass sie irgendwie eine Entscheidung treffen da. Aber eigentlich (..) gehe ich davon aus, dass sie wenigstens Teile der Punkte kriegen sollten. (4) Dass hier eine Verklemmung drin vorkommen kann, bei einer Straßenkreuzung, erkennen sie auch. So wie Tom das in der Stunde erkannt hat, so lesen die das und sollten das eigentlich auch hinschreiben. (...) Die Fußgängerampel. Gut, die haben wir auch Anforderungsbreich drei hingeschrieben (..) Richard kriegt das hin. (.) Nils kriegt das hin. (4) Tom und Lars vielleicht

teilweise, vielleicht auch ganz. Die sind immer so dazwischen. Aber das ist schon/ Ich denke die Aufgabe entspricht dem Anforderungsbereich drei. Die wird auch (.) die Gruppen dann spalten. Wer das schafft, wer da eine Idee hat und wer nicht. Also ja. (.)

I: Mhm Ok. Wenn du die Stunde nochmal halten müsstest, was würdest du verändern? [28:06]

B: (5) So, wie ich es vorhin gesagt habe. Ich würde das mit dem (..) dem Lego, mit den Motoren (...) mir da eine Veranschaulichung überlegen. Vielleicht mir überlegen, wie ich das aufmale oder was mitbringe, oder was zeige. Da würde ich (..) was in der Hinterhand haben, um dieses, wo es noch möglich ist, oder vielliecht auch diese Prozesse, diese Abwechslung der Prozesse, das zu visualisieren, denke ich, da könnte man schon nochmal ein bisschen was haben. Da hätte ich gedacht, bringen die Schüler von selbst mehr. Und da sie nicht mehr gebracht haben, ist es lohnenswert, da etwas da zu haben, weil es einfach interessant ist auch für die Schüler. Das zu sehen, dass es eben nicht nur um Waschmaschinen und Badbenutzung geht, die sie aus dem täglichen Leben, die Badbenutzung in der WG oder in der Familie mit Geschwistern, kennen sie wahrscheinlich zur Genüge. Aber dass es auch technisch einfach dieses Problem gibt, das haben sie noch nicht so auf dem Schirm gehabt. Das war tatsächlich neu für sie. Und das haben sie eben auch bei den Lego-Motoren nicht unbeding wahrgenommen. (Im ersten Moment?) Da sollte man was in der Hinterhand haben, um ihnen das noch deutlicher zu machen. (.) Da bin ich mir eben nicht sicher, wie viel wirklich davon hängengeblieben ist. (7) Jetzt muss ich mal überlegen. Und dann die Geschichte mit den Pfeilen, mit den Kanten, dass die eben, mit den farbigen Marken.

I: Ja, das hattest du ja schon angesprochen.

B: Ansonsten muss ich sagen, ist das eine runde und schöne Stunde. Schon. Das habe ich so empfunden.

I: Eine Sache noch, die mir jetzt gerade wieder einfällt, die mir aufgefallen ist auch in der Beobachtung. Das war bei dem einen Schüler, ich kenne seinen Namen jetzt nicht, der dann bei dem Problem der Philosophen angefangen hat mit der Gleichzeitigkeit und dann auch von den Methoden und Funktionen gesprochen hat, ... [30:02]

B: Richard.

I: ... da hatte ich den Eindruck, dass da sehr stark auf der Ebene Programmierung gedacht wurde.

B: Ja (zustimmend).

I: Ist das ein generelles Problem? Oder, also woran lag das an der Stelle? Also du hast ja dann wirklich versucht...

B: Nein, ich war einfach überrascht davon. Deswegen bin ich ja auch/ In der Situation bin ich ja auch nur im Prinzip stecken geblieben und hab nicht mehr so richtig gewusst, weil ich selbst noch nicht die Verknüpfung (.) von diesem wechselseitigen Ausschluss zu dem, was wir in der Programmierung eigentlich machen. (.) Diese Verknüfung habe ich noch nicht und die hat er jetzt in dem Moment gesucht. (4) Mhmm. Und deine Frage ist jetzt, ob das immer so ist. [30:35]

I: Ja, sind die Schüler das generell gewohnt, dass Dinge sofort mit Programmen verknüpft werden?

B: Ich denke schon, ja. Ich denke schon, dass sie das versuchen, weil das ja letztendlich immer die gemeinsame Basis ist. (..) Dass sie das veruschen ja, das glaube ich schon. Weil sie schon auch/ Letztendlich ist ja alles, was sie machen, sie setzen das alles in Bezug zu einem Kern. [31:16]

Und der Kern ist letztendlich das: da läuft ein Programm und das macht irgendetwas und das hat jetzt verschiedene Facetten. Und irgendwie, ob ich jetzt eine Datenbank anschaue oder die theoretische Informatik, letztendlich hat es immer irgendwo einen Verknüpfungspunkt. Und da sehen sie so ein bisschen/ Das bauen sie alles immer so ein bisschen um die Programmierung herum. Ja. Und das hat er natürlich hier auch versucht. Und da war ich dann in dem Moment auch etwas (unv.)

[32:05] **I:** Also ich hatte den Eindruck/ Du hast ja dann auch versucht, direkt mit der Frage: „Was ist denn jetzt hier in dem SACHVERHALT eigentlich die Ressource? Was wird denn da benutzt?“ und so. Dass du das ja wirklich auch versucht hast, wieder auf diese realweltliche Ebene zu heben. Da hatte ich auch das Gefühl, dass die Schüler da ein bisschen sich schwer getan haben. Aber das liegt vielleicht auch genau daran.

[32:24] **B:** Das Problem war aber auch, hier stand eben auch nichts von der realen Welt mehr. Hier war das PRINZIP des wechselseitigen Ausschlusses und sie haben versucht, das Prinzip allgemein zu erklären und haben versucht mit der Programmierung und da war ich jetzt irgendwie auf seine Begrifflichkeiten nicht vorbereitet. Also das stimmt, da war ein Bruch und deswegen/ Da habe ich ja auch ein bisschen rumgeeiert dann und im Endeffekt dann den Satz einfach genommen und an die Tafel geschrieben. (.) Also stimmt, da war ein Bruch drin und da müsste man nochmal überlegen, wie man das umformuliert, aber da fällt mir im Moment auch keine Lösung ein.

[32:52] **I:** Kein Problem. (.) Ja, gut. Dann bleibt mir nur noch zu fragen, ob es noch irgendetwas gibt, was du anmerken möchtest, ob es noch irgendwas gibt, was deiner Meinung nach wichtig ist, was ich jetzt mit meinen Fragen hier nicht abgedeckt habe.

B: Nein, das war es eigentlich. Also wie gesagt, eine schöne Stunde. Hat Spaß gemacht und würde ich wieder versuchen. (..)

I: OK.

B: Ja.

I: Dann vielen Dank bis dahin.

B.4 Schülertest

B.4.1 Aufgaben und Erwartungshorizont

Aufgabe 1: Plätze, Transitionen und Marken. Für die Aufgabe werden maximal sechs Punkte vergeben: ein Punkt für jedes korrekt dargestellte Item des Textes:

Das Netz hat die Plätze A und B **(1)**; und die Transitionen x und y **(1)**. Die Transition x ist aktiviert, wenn auf A eine Marke liegt **(1)**. Wenn x schaltet, bleiben die Marken auf A unverändert und es entsteht eine Marke auf B **(1)**. Die Transition y ist aktiviert, wenn auf A eine Marke liegt **(1)**. Wenn y schaltet, verschwindet eine Marke von A. Zu Beginn liegen zwei Marken auf A **(1)**.

Aufgabe 2: Baustelle. Für (a) werden maximal zwei Punkte vergeben. Ein Punkt wird vergeben, wenn erkannt wurde, welche Transitionen das Betreten und Verlassen des Baustellenbreichs repräsentieren. Ein zweiter Punkt wird vergeben, wenn die Mutex-Bedingung erfüllt ist.

Für (b) wird maximal ein Punkt vergeben: wenn die Knappe Ressource korrekt identifiziert wurde. Akzeptiert werden Begriffe wie „Fahrbahn“ oder „Platz im Baustellenbereich,“ nicht jedoch „der Baustellenbereich.“

Aufgabe 3: Straßenkreuzung. Für (a) werden maximal 3 Punkte vergeben. Ein Punkt wird vergeben, wenn für mindestens eine Transition Kreuzung überqueren eine korrekte Kante vom Platz Straße frei des rechten Nachbarn ergänzt wurde. Ein zweiter Punkt wird vergeben wenn für mindestens eine Transition Kreuzung überqueren auch eine korrekte Schlinge zum Platz Straße frei des rechten Nachbarn modelliert wurde. Ein dritter Punkt wird vergeben, wenn für alle Richtungen symmetrisch dieselbe Modellierung ergänzt wurde, egal ob diese korrekt ist oder nicht.

Für (b) wird maximal ein Punkt vergeben: wenn das Problem der Verklemmung korrekt beschrieben wurde.

Aufgabe 4: Fußgängerampel. Es werden maximal drei Punkte vergeben. Ein Punkt wird vergeben, wenn die korrekten Transitionen identifiziert wurden, die mit der Ampel interagieren müssen. Ein zweiter Punkt wird vergeben, wenn mindestens eine Transition modelliert wurde, d.h., wenn erkannt wurde, dass die Ampel selbst *schalten* muss. Ein dritter Punkt wird vergeben, wenn eine korrekte Ampelschaltung modelliert wurde, sodass von den beiden Transitionen Ampel passieren und Straße überqueren immer genau eine aktiviert ist.

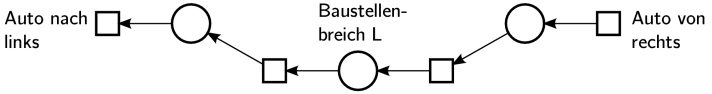
Aufgabe 1: Plätze, Transitionen und Marken (6p)

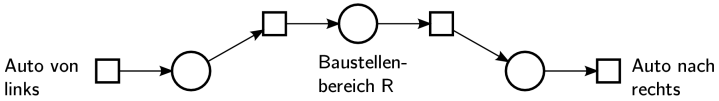
a) Zeichne das folgende Petrinetz:

Das Netz hat die Plätze A und B; und die Transitionen x und y. Die Transition x ist aktiviert, wenn auf A eine Marke liegt. Wenn x schaltet, bleiben die Marken auf A unverändert und es entsteht eine Marke auf B. Die Transition y ist aktiviert, wenn auf A eine Marke liegt. Wenn y schaltet, verschwindet eine Marke von A. Zu Beginn liegen zwei Marken auf A.

Aufgabe 2: Baustelle (3p)

Das folgende Petrinetz modelliert zwei sich entgegen kommende Fahrbahnen. Ihr Weg führt jedoch über eine enge Baustelle, die nur breit genug für ein Auto ist.





a) Ergänze das Petrinetz, sodass sich höchstens ein Auto im Baustellenbereich befinden kann. (Auf den Plätzen **Baustellenbereich L** und **Baustellenbereich R** darf zusammen höchstens eine Marke liegen.)

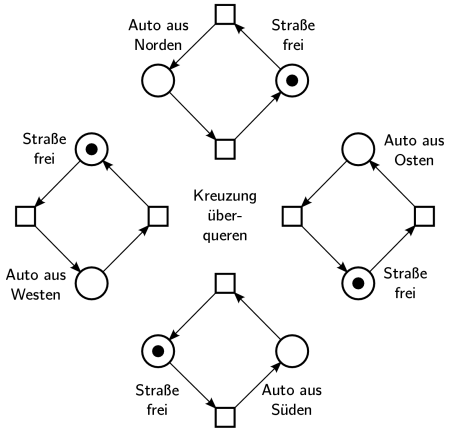
b) Die Baustelle erzeugt einen wechselseitigen Ausschluss. Was ist dabei die knappe Ressource?

_____

_____

Aufgabe 3: Straßenkreuzung (4p)

Das folgende Petrinetz modelliert eine Kreuzung zweier gleichberechtigter Straßen. Es gilt rechts vor links.



a) Ergänze das Petrinetz, sodass jedes Auto die Kreuzung nur überqueren kann, wenn die Straße rechts von ihm frei ist.

b) Beschreibe das Problem, das dabei auftreten kann.

_____

_____

_____

_____

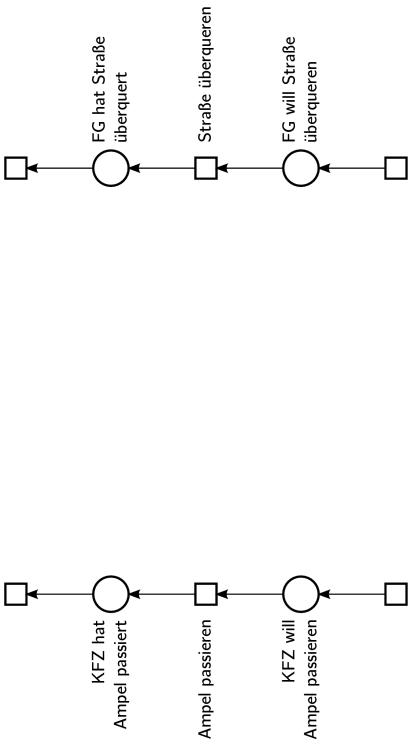
_____

_____

Aufgabe 4: Fußgängerampel (3p)

Eine Fußgängerampel regelt, wann Fußgänger (FG) über die Straße gehen dürfen und wann Autos (KFZ) die Fußgängerampel passieren dürfen. Beides darf nie zur selben Zeit möglich sein. Die Ampel *schaltet von alleine* in belibigen Intervallen.

Modelliere eine solche Fußgängerampel, die verhindert, dass die Fußgänger von den Autos überfahren werden.



B.4.2 Ergebnisse

Für vergessene Marken wurde in den Modellierungsaufgaben 2 und 4 nur einmal global ein Punkt abgezogen. 6 Schülerinnen und Schüler haben an dem Test teilgenommen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	$\tilde{x}$
									1	1		3			1		**12**

1. Plätze, Transitionen und Marken

0	1	2	3	4	5	6	$\tilde{x}$
					1	5	**6**

2. Baustelle

0	1	2	3	$\tilde{x}$
1	1	3	1	**2**

3. Straßenkreuzung

0	1	2	3	4	$\tilde{x}$
		1	3	2	**3**

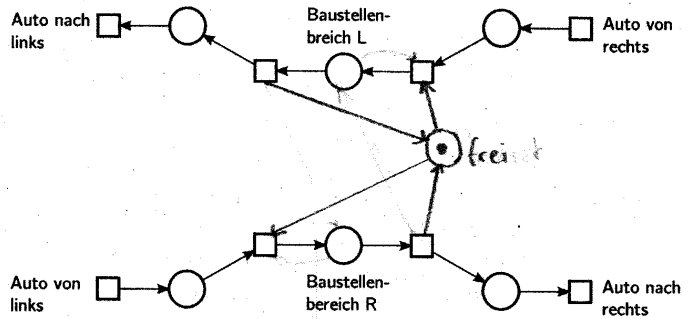
4. Fußgängerampel

0	1	2	3	$\tilde{x}$
4			2	**0**

Nils („Die knappe Ressource ist die eine Fahrbahn im Baustellenbereich, auf der sich aus zwei Richtungen nur ein Auto befinden kann.“)

Aufgabe 2: Baustelle (3p)

Das folgende Petrinetz modelliert zwei sich entgegen kommende Fahrbahnen. Ihr Weg führt jedoch über eine enge Baustelle, die nur breit genug für ein Auto ist.



a) Ergänze das Petrinetz, sodass sich höchstens ein Auto im Baustellenbereich befinden kann. (Auf den Plätzen **Baustellenbereich L** und **Baustellenbereich R** darf zusammen höchstens eine Marke liegen.)

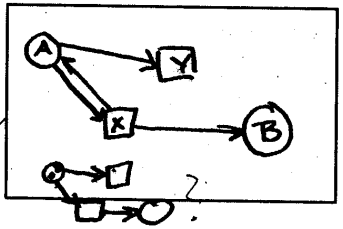
b) Die Baustelle erzeugt einen wechselseitigen Ausschluss. Was ist dabei die knappe Ressource?

Die knappe Ressource, ist die eine Fahrbahn im Baustellenbereich, auf der sich aus zwei Richtungen nur ein Auto befinden kann.

Saskia

Aufgabe 1: Plätze, Transitionen und Marken (6p) 5/6

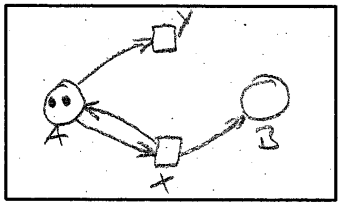
a) Zeichne das folgende Petrinetz:
Das Netz hat die Plätze A und B; und die Transitionen x und y. Die Transition x ist aktiviert, wenn auf A eine Marke liegt. Wenn x schaltet, bleiben die Marken auf A unverändert und es entsteht eine Marke auf B. Die Transition y ist aktiviert, wenn auf A eine Marke liegt. Wenn y schaltet, verschwindet eine Marke von A. Zu Beginn liegen zwei Marken auf A.



Marco

Aufgabe 1: Plätze, Transitionen und Marken (6p)

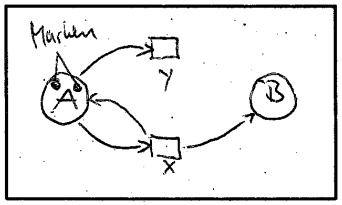
a) Zeichne das folgende Petrinetz:
Das Netz hat die Plätze A und B; und die Transitionen x und y. Die Transition x ist aktiviert, wenn auf A eine Marke liegt. Wenn x schaltet, bleiben die Marken auf A unverändert und es entsteht eine Marke auf B. Die Transition y ist aktiviert, wenn auf A eine Marke liegt. Wenn y schaltet, verschwindet eine Marke von A. Zu Beginn liegen zwei Marken auf A.



Tom

Aufgabe 1: Plätze, Transitionen und Marken (6p)

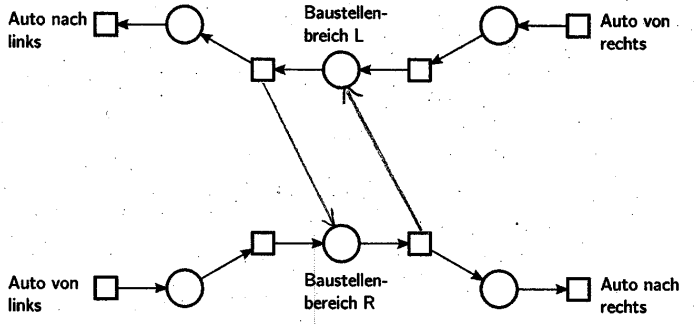
a) Zeichne das folgende Petrinetz:
Das Netz hat die Plätze A und B; und die Transitionen x und y. Die Transition x ist aktiviert, wenn auf A eine Marke liegt. Wenn x schaltet, bleiben die Marken auf A unverändert und es entsteht eine Marke auf B. Die Transition y ist aktiviert, wenn auf A eine Marke liegt. Wenn y schaltet, verschwindet eine Marke von A. Zu Beginn liegen zwei Marken auf A.



Marco („Der Platz beim Baustellenbereich.“)

Aufgabe 2: Baustelle (3p)

Das folgende Petrinetz modelliert zwei sich entgegen kommende Fahrbahnen. Ihr Weg führt jedoch über eine enge Baustelle, die nur breit genug für ein Auto ist.



a) Ergänze das Petrinetz, sodass sich höchstens ein Auto im Baustellenbereich befinden kann. (Auf den Plätzen **Baustellenbereich L** und **Baustellenbereich R** darf zusammen höchstens eine Marke liegen.)

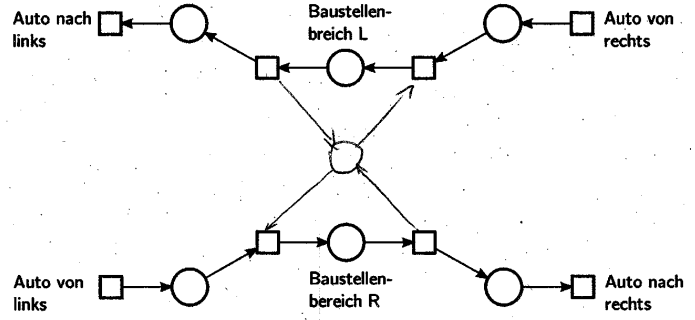
b) Die Baustelle erzeugt einen wechselseitigen Ausschluss. Was ist dabei die knappe Ressource?

Der Platz beim Baustellenbereich

Tom („Die knappe Ressource ist die Fahrbahn im Baustellenbereich.“)

Aufgabe 2: Baustelle (3p)

Das folgende Petrinetz modelliert zwei sich entgegen kommende Fahrbahnen. Ihr Weg führt jedoch über eine enge Baustelle, die nur breit genug für ein Auto ist.



a) Ergänze das Petrinetz, sodass sich höchstens ein Auto im Baustellenbereich befinden kann. (Auf den Plätzen **Baustellenbereich L** und **Baustellenbereich R** darf zusammen höchstens eine Marke liegen.)

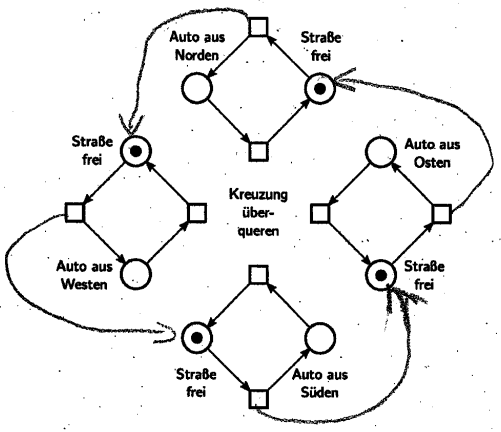
b) Die Baustelle erzeugt einen wechselseitigen Ausschluss. Was ist dabei die knappe Ressource?

Die knappe Ressource ist die Fahrbahn im Baustellenbereich.

Marco („Wahrscheinlich wird irgendwann niemand mehr fahren können, da alle Autos gleichzeitig über die Kreuzung wollen.“)

Aufgabe 3: Straßenkreuzung (4p)

Das folgende Petrinetz modelliert eine Kreuzung zweier gleichberechtigter Straßen. Es gilt rechts vor links.



a) Ergänze das Petrinetz, sodass jedes Auto die Kreuzung nur überqueren kann, wenn die Straße rechts von ihm frei ist.

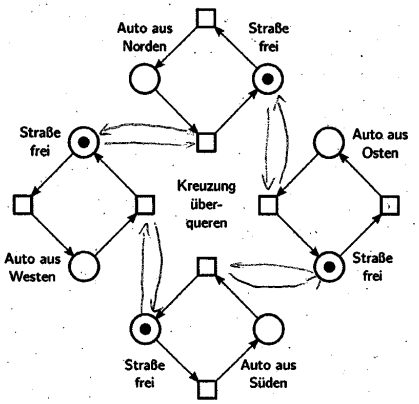
b) Beschreibe das Problem, das dabei auftreten kann.

Wahrscheinlich wird irgendwann niemand mehr fahren können, da alle Autos gleichzeitig über die Kreuzung wollen

Tom („Aus allen Richtungen kommen gleichzeitig Autos, nun hat jeder ein Auto rechts und somit gibt es keine weitere Transition die möglich ist.“)

Aufgabe 3: Straßenkreuzung (4p)

Das folgende Petrinetz modelliert eine Kreuzung zweier gleichberechtigter Straßen. Es gilt rechts vor links.



a) Ergänze das Petrinetz, sodass jedes Auto die Kreuzung nur überqueren kann, wenn die Straße rechts von ihm frei ist.

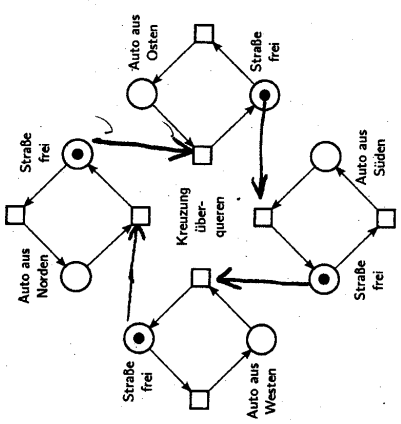
b) Beschreibe das Problem, das dabei auftreten kann.

Aus allen Richtungen kommen gleichzeitig Autos, nun hat jeder ein Auto rechts und somit gibt es keine weitere Transition die möglich ist.

Saskia („Es kann das Problem auftreten, dass an jeder Ecke ein Auto steht und wartet, dass rechts frei wird. Wenn jedoch überall ein Auto steht kann irgendwann rechts nicht mehr frei werden.“)

Aufgabe 3: Straßenkreuzung (4p)

Das folgende Petrinetz modelliert eine Kreuzung zweier gleichberechtigter Straßen. Es gilt rechts vor links.



a) Ergänze das Petrinetz, sodass jedes Auto die Kreuzung nur überqueren kann, wenn die Straße rechts von ihm frei ist.

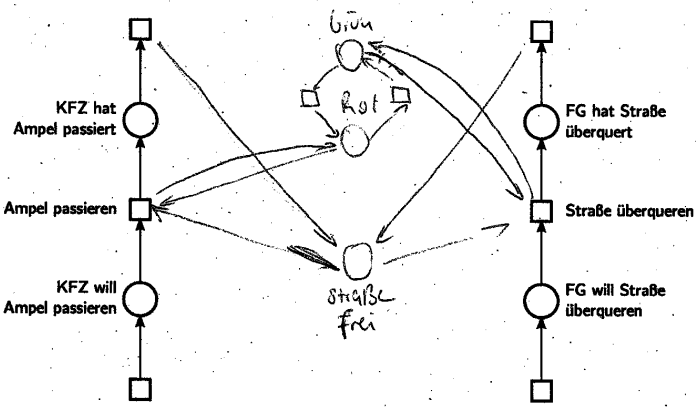
b) Beschreibe das Problem, das dabei auftreten kann.

Es kann das Problem auftreten, dass an jeder Ecke ein Auto steht und wartet, dass rechts frei wird. Wenn jedoch überall ein Auto steht kann irgendwann rechts nichtmehr frei werden.

Aufgabe 4: Fußgängerampel (3p)

Eine Fußgängerampel regelt, wann Fußgänger (FG) über die Straße gehen dürfen und wann Autos (KFZ) die Fußgängerampel passieren dürfen. Beides darf nie zur selben Zeit möglich sein. Die Ampel *schaltet von alleine* in belibigen Intervallen.

Modelliere eine solche Fußgängerampel, die verhindert, dass die Fußgänger von den Autos überfahren werden.



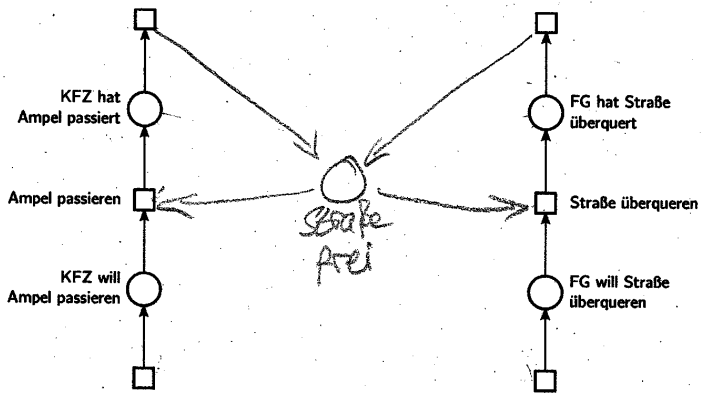
(Nur die Ampelschaltung (Grün, Rot) betrachten.)

Tom („Nur die Ampelschaltung (Grün, Rot) betrachten.“)

Aufgabe 4: Fußgängerampel (3p)

Eine Fußgängerampel regelt, wann Fußgänger (FG) über die Straße gehen dürfen und wann Autos (KFZ) die Fußgängerampel passieren dürfen. Beides darf nie zur selben Zeit möglich sein. Die Ampel *schaltet von alleine* in belibigen Intervallen.

Modelliere eine solche Fußgängerampel, die verhindert, dass die Fußgänger von den Autos überfahren werden.

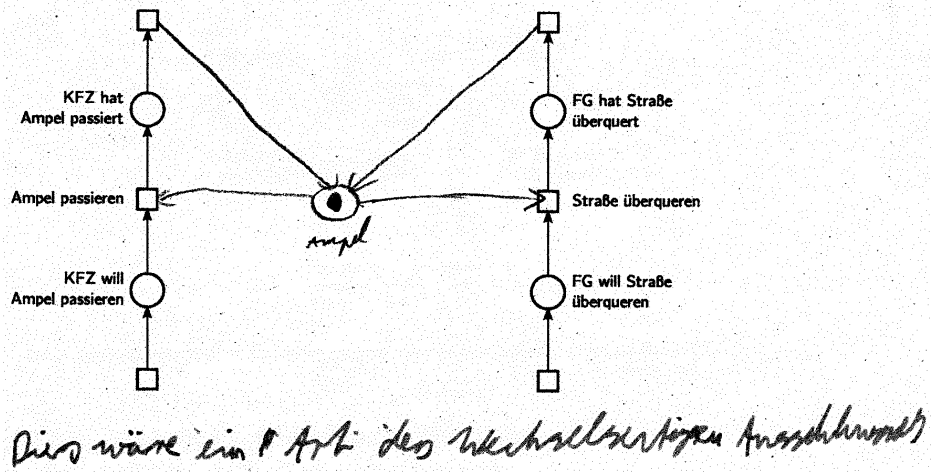


Marco

Aufgabe 4: Fußgängerampel (3p)

Eine Fußgängerampel regelt, wann Fußgänger (FG) über die Straße gehen dürfen und wann Autos (KFZ) die Fußgängerampel passieren dürfen. Beides darf nie zur selben Zeit möglich sein. Die Ampel *schaltet von alleine* in belibigen Intervallen.

Modelliere eine solche Fußgängerampel, die verhindert, dass die Fußgänger von den Autos überfahren werden.



Richard („Dies wäre ein[e] Art des Wechselseitigen Ausschlusses.“)