

**Petrinetze als Modellierungswerkzeug
auf dem Weg zur Implementation von
Steuerungsprozessen in der Sekundarstufe II**

Schriftliche Hausarbeit

im Rahmen der Zweiten Staatsprüfung für das Lehramt für die
Sekundarstufen II und I

Thorsten Roterling

Studienseminar Neuss

Erstgutachter: Tim Lethen, StD.

Inhaltsverzeichnis

Vorbemerkung.....	3
Motivation für diese Arbeit.....	3
Fachliche Vorbetrachtungen zu Petrinetzen.....	4
Geschichte der Petrinetze.....	4
Aufbau von Petrinetzen.....	5
Petrinetze im Informatik-Unterricht.....	6
Ziele des unterrichtspraktischen Teils dieser Arbeit.....	8
Die Lernumgebung.....	9
Planung und Durchführung einer Einführungsstunde.....	11
Evaluationsmethode.....	13
Phase 1.....	14
Phase 2.....	15
Phase 3.....	16
Phase 4.....	18
Schlussevaluation.....	20
Petrinetze als Modellierungswerkzeug.....	20
Projektbeschreibung.....	20
Einsatz von Petrinetzen innerhalb des Projekts.....	21
Evaluationsmethode.....	23
Gruppe 1.....	23
Gruppe 2.....	25
Gruppe 3.....	26
Schlussbetrachtungen.....	30
Literaturverzeichnis.....	33
Softwareverzeichnis.....	33
Anhang.....	34
Evaluierungsbogen.....	34
Schülerkommentare im Rahmen des Evaluierungsbogens.....	36
Begleit-CD.....	37

Vorbemerkung

Zum Zeitpunkt der Anfertigung dieser Arbeit wurde an den Gymnasien in Nordrhein-Westfalen gerade schrittweise die klassische Sekundarstufe mit 9 Jahren von der um ein Jahr verkürzten Sekundarstufe mit 8 Jahren (G8) abgelöst. Um Verwechslungen bei der Nummerierung der Jahrgangsstufen zu vermeiden, wird daher in dieser Arbeit stets von der *Einführungsphase* (alt: 11, neu: 10) und der *Qualifizierungsphase* (alt: 12+13, neu 11+12) gesprochen.

Motivation für diese Arbeit

Im Laufe meines Hochschulstudiums waren Petrinetze mehrfach Gegenstand von Vorlesungen der theoretischen Informatik. In den Informatik-Lehrplänen des Landes Nordrhein-Westfalens sowie in den Vorgaben für das Zentralabitur taucht dieser Gegenstand jedoch nirgends auf. In Standardwerken der Informatik-Didaktik widmen zum Beispiel Schubert und Schwill als auch Hubwieser Petrinetzen jeweils ein sehr kurzes Kapitel, doch eine umfassende Einbettung in die Unterrichtspraxis findet nicht statt. Vielmehr erwartet den Leser dort nicht viel mehr als eine kurze Einführung in den Gegenstand. Auch bei meinen Internetrecherchen bin ich über keine Arbeit gestolpert, die Petrinetze in der Unterrichtspraxis näher beleuchtet, während der Fundus an Einführungsliteratur auf Hochschulniveau fast unüberschaubar wirkt.

Dies verwunderte mich in sofern, als dass ich aus meinem Studium in Erinnerung behalten hatte, dass Petrinetze mit nur wenigen Elementen, vergleichsweise komplexe Prozesse veranschaulichen konnten. Ein geeignetes Software-Tool vorausgesetzt, konnte man mit einem Petrinetz zunächst dynamisch experimentieren und ein Gefühl für das betreffende Netz entwickeln, bevor man es anschließend theoretisch fundiert analysierte. Dies erinnerte mich stark an die Motivation beim Einsatz von dynamischer Geometriesoftware in der Mathematik, bei der Sätze der Geometrie zwar nicht streng-mathematisch bewiesen, aber ihre Gültigkeit anschaulich nachvollzogen werden kann.

Als innovativen Ansatz für die Schulpraxis möchte ich mit dieser Arbeit daher näher untersuchen, inwiefern sich Petrinetze auch im Informatikunterricht der gymnasialen Oberstufe bereits einsetzen lassen. Nach der Beleuchtung zweier verschiedener Dimensionen, die auf den knappen didaktischen Betrachtungen von Schubert und Schwill beruhen, habe ich den modellierenden Ansatz näher verfolgt und als Schwerpunkt dieser Arbeit eine konkrete unterrichtliche

Umsetzung entwickelt. Mit dem Ziel diese später auch geeignet, wenn auch aufgrund der kleinen Stichprobenmenge nicht repräsentativ, evaluieren zu können, habe ich die Unterrichtsreihe nicht nur theoretisch geplant, sondern auch praktisch in meinem eigenen Unterricht im Informatikkurs der Jahrgangsstufe 11 umgesetzt. Der Kurs setzt sich aus 6 Schülerinnen und 14 Schülern zusammen, wovon 2 Schüler jedoch über den gesamten relevanten Zeitraum nicht anwesend waren.

Fachliche Vorbetrachtungen zu Petrinetzen

In diesem Abschnitt sollen Petrinetze kurz auf fachlicher Ebene vorgestellt werden, um insbesondere die Terminologie, die in dieser Arbeit verwendet wird, darzulegen. Es sei ausdrücklich darauf hingewiesen, dass dies keine Basis dafür darstellt, wie Petrinetze im Unterricht eingeführt werden. Dies erfolgt umfassend im Hauptteil dieser Arbeit.

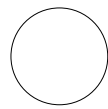
Geschichte der Petrinetze

Ende der 50er Jahre machte sich Carl Adam Petri Gedanken über eine beliebig oft erweiterbare Rechnerarchitektur. Er war zu dieser Zeit Oberassistent am Institut für Instrumentelle Mathematik der Universität Bonn und hielt es für überaus ineffizient rekursive Berechnungen stets von vorne beginnen zu müssen, wenn man auf eine neue Rechenmaschine wechselte, weil sich die vorige erst im Nachhinein als zu schwach für diese Berechnung erwies. Ihm war klar, dass sich eine solche Rechnerarchitektur nicht zentral steuern ließe, sondern die einzelnen Erweiterungen autark arbeiten müssten.

Um solche nebenläufigen Prozesse zu beschreiben und zu analysieren, entwickelte Petri die später nach ihm benannten Petrinetze. Er entwarf sie zunächst als rein mathematisches Modell und erweiterte sie erst später um eine grafische Repräsentation. Gerade diese einfache grafische Repräsentation verhalf Petrinetze aber wohl erst zu ihrer Popularität (*Reisig, S. 24*).

Aufbau von Petrinetzen

Die Netzstruktur eines Petrinetzes besteht aus drei Elementen:



Stellen (oft auch Plätze genannt, Symbol p) sind passive Elemente, die einen Zustand besitzen. Sie werden grafisch durch einen Kreis dargestellt.



Transitionen (Symbol t) sind aktive Elemente, indem sie die Zustände der benachbarten Stellen beeinflussen. Sie werden graphisch durch ein Rechteck oder einen Balken dargestellt.



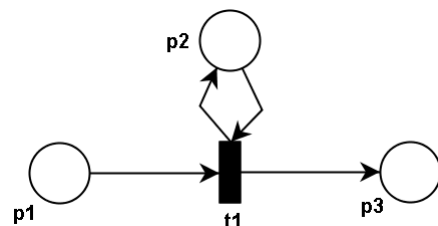
Kanten (Symbol f) stellen gerichtete Beziehungen zwischen Stellen und Transitionen her. Dabei sind Kanten zwischen Elementen gleichen Typs unzulässig. Sie werden grafisch durch einen Pfeil dargestellt.

Mathematisch lässt sich diese Netzstruktur mit einem 3-Tupel $N=(P, T, F)$ beschreiben, mit P die Menge aller Stellen, T die Menge aller Transitionen und F die Menge aller Kanten, also aller Relationen $(P \times T) \cup (T \times P)$. Alle genannten Mengen dürfen nicht leer sein. Damit gleicht die zugrundeliegende Netzstruktur eines Petrinetzes einem gerichteten Graphen mit zwei voneinander verschiedenen Knotentypen.

Die Netzstruktur eines Petrinetzes alleine ist demzufolge zunächst statisch. Seine Dynamik erhält das Petrinetz durch Hinzunahme einer **Markierung**, bei der eine Anzahl von **Marken** auf die Stellen des Netzes verteilt wird und somit den aktuellen Zustand der jeweiligen Stelle symbolisiert. Marken werden grafisch entweder durch eine entsprechende Anzahl von gefüllten Kreisen oder durch eine einbeschriebene Zahl dargestellt. Jedes Petrinetz besitzt eine Anfangsmarkierung (Symbol M_0).

Um die folgenden Definitionen zu vereinfachen, seien an dieser Stelle zwei allgemein geläufige Begriffe der Netzstruktur eines Petrinetzes näher definiert. Als **Vorbereich** einer Stelle oder einer Transition x bezeichnet man die Menge aller Elemente y , für die eine Relation yFx besteht, und als

Nachbereich die Menge, für die eine Relation xFy besteht. Im rechts dargestellten Beispiel befindet sich Stelle p_1 im Vorbereich der Transition t_1 , p_3 im Nachbereich dieser und p_2 befindet sich sowohl im Vor- als auch im Nachbereich der Transition.



Beim sogenannten *Feuern* einer Transition wird nun eine neue Markierung des Petrinetzes erzeugt, bei der alle Stellen im Vorbereich der Transition eine Marke verlieren, während die Stellen im Nachbereich entsprechend eine Marke hinzugewinnen. Voraussetzung dafür, dass eine Transition feuern kann, ist, dass sie aktiviert ist. Eine Transition heißt *aktiviert*, wenn alle Stellen im Vorbereich noch mindestens eine Marke besitzen.

Für dieses elementare Petrinetz gibt es mittlerweile eine kaum mehr überschaubare Anzahl an Erweiterungen. Zwei davon sind so weit verbreitet, dass sie häufig schon mit zu elementaren Petrinetzen gezählt werden. Da sie für die spätere Modellierung wichtig sind, möchte auch ich diese hier kurz einführen.

Jeder Kante kann ein numerisches *Kantengewicht* zugeordnet werden. Dadurch kann eine Stelle beim Feuern einer über die Kante benachbarten Transition in einem Schritt mehr als eine Marke verlieren bzw. hinzugewinnen, nämlich ebenso viele, wie das Kantengewicht es angibt. Entsprechend muss beim Kriterium für eine aktivierte Transition das Kantengewicht berücksichtigt werden.

Zudem kann jeder Stelle eine maximale *Kapazität* zugeordnet werden. Dies sorgt dafür, dass für die Aktivierung einer Transition auch der Nachbereich Beachtung finden muss, da gewährleistet sein muss, dass durch das Feuern die Markenanzahl einer Stelle nicht ihre maximale Kapazität übersteigt. (Reisig, S. 25 ff.)

Petrinetze im Informatik-Unterricht

In welcher Weise können solche Petrinetze nun den Informatikunterricht bereichern? Zwar werden in einigen Standardwerken der Informatik-Didaktik Petrinetze beiläufig erwähnt, lediglich bei Schubert und Schwill fand ich jedoch eine kurze didaktische Einschätzung dieser Fragestellung. Sie sehen grundlegend zwei verschiedene Dimensionen: zum einen können sie *„als Ergänzung zu programmiersprachlichen Darstellungen [...] das Verständnis für parallele Konzepte [unterstützen], indem sie der mehr formalen und relativ maschinennahen Beschreibung durch eine Programmiersprache eine mehr informelle und für die Schüler einprägsame Darstellung auf höherem Niveau gegenüberstellen.“* (Schubert & Schwill, S. 193)

Zum anderen seien Petrinetze in der Lage auch abseits einer konkreten programmiersprachlichen Umsetzung das Konzept paralleler Prozesse zu vermitteln und dabei *„den Unterricht dennoch attraktiv und lebendig zu gestalten“*, unter anderem weil *„der dynamische Aspekt, anders als bei*

Programmen, die man ‚trocken‘ nachvollzieht, deutlich sichtbar bleibt.“ (Schubert & Schwill, S. 193 f.) Sie heben dabei insbesondere hervor, dass sich die Simulation eines Petrinetzes besonders einfach gestalten und sich auch mit einfachen Hilfsmitteln per Hand auf dem Schreibtisch realisieren ließe.

Ich möchte diese beiden von Schubert und Schwill genannten Aspekte ein wenig umformulieren, um sie konkreter an die Vorgaben des Informatik-Lehrplans der gymnasialen Oberstufe heranzuführen. Zwar dienen Petrinetzen in beiden Fällen letztlich der Veranschaulichung paralleler Konzepte, doch besteht ein Unterschied, ob diese parallelen Prozesse den Unterrichtsgegenstand bilden, wie im zweiten Fall, oder ein programmiersprachliches (Teil-)Problem auf dem Weg von der Aufgabenstellung zur fertigen Implementierung als Programm darstellen, wie im ersten Fall.

Bilden parallele Konzepte den eigentlichen Unterrichtsgegenstand, so befinden wir uns ganz klar auf dem Gebiet der theoretischen Informatik, welches der Informatik-Lehrplan der gymnasialen Oberstufe als paradigmunenabhängige Sequenz für die Qualifizierungsphase vorsieht. Neben „klassische[n] Themenbausteine[n] wie die Theorie der endlichen Automaten, der Schaltnetze und Schaltwerke oder der formalen Sprachen“ (Informatik-Lehrplan, S. 70) können parallele Konzepte eine nahtlose Ergänzung dieser Themenreihe darstellen, die zusätzliche, interessante Fragestellungen in diesem Gebiet aufwirft.

Mehr als noch vielleicht reguläre Sprachen durch einen statischen Automaten veranschaulicht und erfahrbar gemacht werden können und so bislang auch im Informatikunterricht eine Symbiose bildeten, verlangen die hochgradig dynamischen, parallele Konzepte nach einer geeigneten und insbesondere dynamischen Visualisierung. Nach Schubert und Schwill können gerade dies Petrinetze leisten. In der folgenden Arbeit möchte ich diesen Ansatz kurz als *theoretisch-analytischen Ansatz* bezeichnen.

Schaut man sich dagegen den ersten Aspekt von Schubert und Schwill näher an, so treten die theoretischen Aspekte von parallelen Konzepten deutlich in den Hintergrund. Man könnte so formulieren, dass Petrinetze Schülerinnen und Schüler auf dem Weg zur abstrakten und formalen Programmierung als Hilfsmittel dienen können, die Idee paralleler Prozesse informell darzustellen, ohne dass man näher auf die theoretischen Natur dieser eingehen müsse. So lösen sich Petrinetze ein ganzes Stück weit vom eigentlichen Gegenstand der parallelen Prozesse ab und werden viel mehr zu einem Modellierungswerkzeug in einem Software-Entwicklungsprozess, das als (Teil-)Problem die Modellierung

eines parallelen Prozesses beinhaltet. Ich möchte diesen Ansatz im weiteren Verlauf dieser Arbeit daher kurz als *modellierenden Ansatz* bezeichnen.

Solche Modellierungsprozesse sieht der Informatik-Lehrplan der gymnasialen Oberstufe vor allem in der Einführungsphase vor, ganz besonders bei den objektorientierten Sequenzen, kann aber auch zum Einstieg der Qualifizierungsphase fortgesetzt werden (*Informatik-Lehrplan, S. 48 und 58*).

Ziele des unterrichtspraktischen Teils dieser Arbeit

Da zur Zeit der Anfertigung dieser Arbeit die 13. Jahrgangsstufe kurz vor den Abschlussprüfungen stand und der Kollege, der den Informatikkurs der 12. Jahrgangsstufe leitete, schwer erkrankt war, waren unterrichtspraktische Versuche in der Qualifizierungsphase nicht realisierbar. Daher habe ich mich dazu entschlossen, eine Sequenz in meinem eigenen Informatikkurs der 11. Jahrgangsstufe, also der Einführungsphase, umzusetzen. Gemäß diesen Voraussetzungen und der im vorigen Abschnitt getroffenen Einordnung in den Informatik-Lehrplan, bot es sich daher an, schwerpunktmäßig den modellierenden Ansatz unterrichtspraktisch näher auszuarbeiten. Dabei interessierten mich insbesondere folgende drei Fragestellungen:

Lässt sich der Umgang mit Petrinetzen als Modellierungswerkzeug leicht erlernen?

Nur wenn der Einsatz eines Werkzeugs leicht erlernbar ist, kann es auch als ein solches dienen, um komplexere Probleme damit zu lösen. Ist die Bedienung des Werkzeugs selbst das Problem, wird es nicht als die Hilfe wahrgenommen, die es eigentlich darstellen sollte, sondern als Belastung. Man vergleiche dies beispielsweise mit dem Einsatz von graphischen Taschenrechnern oder Computeralgebrasystemen im Mathematikunterricht. Ich durfte schon mehrfach miterleben, dass Schülerinnen oder Schüler den Rechner irgendwann frustriert beiseite legten und die betreffende Rechnung per Hand vornahmen, oder aber vor Klausuren mehr Fragen zur Rechnerbedienung als zum aktuellen mathematischen Themengebiet gestellt haben.

Auf der anderen Seite zeigt der doch vielfach sehr erfolgreiche Einsatz von graphischen Taschenrechnern oder Computeralgebrasystemen im Mathematikunterricht, dass offensichtlich die Einstellung gegenüber einem Werkzeug maßgeblich auch von einer geeigneten Einführung dessen abhängt. Im Rahmen meiner Arbeit habe ich mir daher ein Konzept für eine solche Einführungsstunde überlegt, die den Schülerinnen und Schülern Petrinetze als

Modellierungswerkzeug vertraut machen soll. Als Zeitrahmen habe ich eine 90-minütige Doppelstunde festgelegt, da alles darüber hinausgehende meines Erachtens in keinem Verhältnis zum späteren Nutzen stehen würde.

Kann die Modellierung eines parallelen Prozesses als Petrinetz Schülerinnen und Schülern helfen, ein besseres Verständnis dieser zu erlangen?

Dies ist die eigentliche These, die Schubert und Schwill aufstellen. Mir lag es also daran, diese auch für mich praktisch zu überprüfen. Insbesondere interessierte mich natürlich, für den Fall, dass sich ein besseres Verständnis der parallelen Prozesse einstellen sollte, welche konkreten Auswirkungen auf die spätere Implementierung dieses hat. Das zweite Ziel meiner Arbeit wird es daher sein, Petrinetze als Baustein eines Software-Entwicklungsprozesses zu verwenden. Um eine Vergleichbarkeit, wenn auch keine repräsentative, zu erreichen, war von vorne herein angedacht, verschiedene Gruppen zu bilden, die Petrinetze auf dem Weg zur fertigen Implementation in unterschiedlichem Umfang einsetzen.

Kann aus einem modellierten Petrinetz unmittelbar ein Grundgerüst für die Implementation abgeleitet werden?

Diese These geht über die vorige noch ein Stück weit hinaus. Ich wollte untersuchen, ob das fertig modellierte Petrinetz nicht nur zum besseren Verständnis der später erwarteten Programmabläufe dienen kann, sondern auch bereits eine Art Schablone für die konkrete Implementierung darstellen kann, vergleichbar etwa mit einem UML-Klassendiagramm, aus dem unmittelbar ein Grundgerüst für das spätere Programm abgeleitet werden kann. Wäre dem so, wären Petrinetze in der Lage bereits einen Teil der Programmierarbeit vorwegzunehmen.

Die Lernumgebung

Wie Schubert und Schwill ausführen, können Petrinetze mit einfachen Hilfsmitteln dargestellt und simuliert werden: *„Als Marken verwendet man [...] am besten Heftzwecken, die man umdreht und an der Nadel anfassend verschiebt, oder Geldstücke. Nichtdeterminismus realisiert man durch einen Würfel, mit dem man diejenige Transition auswürfelt, die unter mehreren gleichzeitig aktivierten den nächsten Schritt schalten soll.“* (Schubert & Schwill, S. 194)

Da ich die Erfahrung gemacht habe, dass Schülerinnen und Schüler, einmal im Computerraum angelangt, darauf brennen den Computer auch einzusetzen, und da mir aus dem Studium bekannt war, dass zahlreiche Software-Tools existieren,

die Petrinetze darstellen und simulieren können, wollte ich in meiner ersten Unterrichtsreihe zu Petrinetzen ein solches der händischen Variante vorziehen.

Großer Vorteil eines Software-Tools ist die Nachhaltigkeit: ein konstruiertes Petrinetz kann einfach abgespeichert und zu einem späteren Zeitpunkt immer wieder betrachtet oder genutzt werden. Weiterhin bieten viele Software-Tools bereits automatisierte Analysewerkzeuge an, die auch für den theoretisch-analytischen Ansatz einen Mehrwert gegenüber der von Schubert und Schwill vorgeschlagenen Variante bieten.

Als kleiner Nachteil erweist sich der Verlust der haptischen Erfahrung, der nach J. S. Bruner als enaktive Repräsentationsform eine wichtige Bedeutung im Lernprozess zukommt. Daher war ich angehalten dieses Element auf andere Weise in die Einführungsstunde einzubringen.

Anforderungen an ein Software-Tool

Der Erfolg einer Unterrichtsreihe, die Petrinetze als Modellierungswerkzeug einsetzt, hängt meines Erachtens nicht unwesentlich mit der Bereitstellung eines geeigneten Software-Tools zusammen, das in der Lage ist, Petrinetze grafisch darzustellen und interaktiv zu simulieren. Letzteres wird in der Terminologie der Petrinetz-Tools üblicherweise als *Markenspiel* bezeichnet. Erst durch dieses Markenspiel wird die Dynamik eines Petrinetzes für die Schülerinnen und Schüler erlebbar.

Die TGI-Gruppe der Universität Hamburg pflegt auf ihrer Website¹ eine Datenbank mit einer großen Auswahl an frei verfügbaren und kostenpflichtigen Petrinetz-Tools. Bei meinen Recherchen musste ich feststellen, dass viele der kostenlosen Petrinetz-Tools etwa um die Jahrtausendwende programmiert und seitdem nicht mehr weiterentwickelt wurden. Dadurch sind diese Programme entweder nicht mehr verfügbar oder laufen unter modernen Betriebssystemen gar nicht oder nur sehr instabil.

Fordert man zudem, dass das Software-Tool eine grafische Oberfläche besitzt und ein Markenspiel durchführen kann, reduziert sich die Auswahl deutlich, da viele professionelle Petri-Tools auf diese Komponenten verzichten und vielmehr eine Sammlung von Analysewerkzeugen für Petrinetze darstellen. Selbst wenn man den theoretisch-analytischen Ansatz verfolgt, sind sie für den Unterricht meines Erachtens ungeeignet, da sie die ikonische Repräsentationsform nach Bruner völlig ausklammern.

1 <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html>

Unter Berücksichtigung der soeben genannten Anforderungen erwies sich die Software *NetLab*, entwickelt vom Institut für Regelungstechnik an der RWTH Aachen, die für Lehrzwecke kostenlos verwendet werden darf, als bester Kandidat. NetLab verspricht einfache Bedienung durch eine komplett deutschsprachige Oberfläche, einen leicht zu bedienenden grafischen Editor und ein komfortables Markenspiel. Darüber hinaus bietet NetLab aber auch zahlreiche Werkzeuge für tiefer greifende Analysen des modellierten Petri-Netzes an.

Leider erwies sich der Einsatz auf den Schülerrechnern meiner Ausbildungsschule als unmöglich, da sich die Software spätestens ab Windows 6.0 nicht unter eingeschränkten Benutzerkonten starten ließ. Verschiedene Workarounds, insbesondere das Setzen zusätzlicher Schreibrechte, stellten keine Lauffähigkeit in unserer Umgebung her.

Als bestmöglicher Ersatz stellte sich die leider nur in englischer Sprache verfügbare Software *Pipe2* heraus. Die Software wurde im Jahr 2002 von mehreren Studenten des Imperial College London als Abschlussarbeit programmiert, unter eine Open-Source-Lizenz gestellt und seitdem von verschiedenen Personen weiterentwickelt, bevor Ende 2007 die bislang letzte Version von Pipe2 erschien. Die Software ist NetLab in der Bedienung sehr ähnlich und die englischsprachige Oberfläche fällt bei der grafischen Modellierung kaum ins Gewicht. Erst bei der Nutzung der Analysewerkzeuge könnten englische Fachbegriffe selbst für Oberstufenschüler noch eine Hürde darstellen.

Im Vorgriff auf die Evaluation der Einführungsstunde sei hier erwähnt, dass 17 von 18 Schülerinnen und Schülern eine deutliche Zustimmung zur These äußerten, dass sich NetLab leicht bedienen und einsetzen ließe; nur einer stimmte der These lediglich eingeschränkt zu. Neben der Erkenntnis, dass Pipe2 von den Schülerinnen und Schülern demnach allgemein angenommen wurde und somit offensichtlich ein geeignetes Werkzeug zur Erstellung und Simulation von Petrinetzen darstellt, zeigt das Ergebnis insbesondere, dass Pipe2 offensichtlich so beschaffen ist, dass die spätere Evaluation zu Petrinetzen nicht durch eine ungeeignete Lernumgebung verfälscht wurde.

Planung und Durchführung einer Einführungsstunde

Sofern Schülerinnen und Schüler noch nie mit Petrinetzen gearbeitet haben, müssen diese geeignet eingeführt werden. Abhängig von der Zielsetzung der Unterrichtsreihe können hier bereits theoretisch-analytische oder modellierende

Aspekte besonders betont werden. Wird der theoretisch-analytische Ansatz verfolgt, so ist es meines Erachtens von entscheidender Wichtigkeit, dass Schülerinnen und Schüler die Terminologie von Petrinetzen sicher beherrschen und neben der grafischen auch die formale Darstellung kennenlernen. Wählt man dagegen den modellierenden Ansatz, so treten diese Aspekte eher in den Hintergrund. Viel wichtiger ist hier, dass Schülerinnen und Schüler das Werkzeug „Petrinetz“ beherrschen lernen und richtig einsetzen können, d.h. insbesondere eine reale Situation als Petrinetz modellieren können.

Da ich wie erwähnt in meiner Unterrichtsreihe Petrinetze als Modellierungswerkzeug einsetzen wollte, betont die von mir geplante Einführungsstunde also vor allem diese Elemente. An verschiedenen Stellen sollte allerdings klar werden, dass sich diese Planung auch als Ausgangsbasis für den theoretisch-analytischen Ansatz verwirklichen lässt.

Obwohl es an schönen Einführungsbeispielen in der Literatur nicht mangelt, wie die Flaschenabfüllanlage bei Schubert und Schwill, der Keksautomat bei Reisig oder die Netzwerkübertragung bei Hubwieser, so mangelt es meines Erachtens all diesen Beispielen an der Möglichkeit sie im Rahmen des Unterrichts direkt erlebbar zu machen. Allenfalls kann ein Modell solcher Maschinen oder Mechanismen im Unterricht präsentiert werden, was jedoch letztlich bedeutet, dass man das Modell eines Modells modelliert. Da ich diese enaktive Repräsentationsform jedoch auch für Oberstufenschüler als ganz entscheidendes Kriterium für den Lernerfolg und insbesondere die Nachhaltigkeit des Erlernen empfinde, habe ich ein eigenes Einführungsbeispiel gewählt, das mit ganz einfachen Materialien des Schulalltags auskommt. Mir ist bewusst, dass mein Einführungsbeispiel auf den ersten Blick wenig spektakulär wirkt, die dahinter liegenden Prozesse sind es jedoch nicht, wie die Schülerinnen und Schüler spätestens in der vierten Phase feststellen mussten.

Nach einem kurzen Lehrervortrag zur Zielsetzung der Unterrichtsstunde und einer dreiminütigen Orientierungsphase, in der die Schülerinnen und Schüler selbstständig die Oberfläche des Software-Tools erkunden können (was nebenbei auch gewährleistet, dass die Software nach Ende dieser Phase auf allen Schülerrechner stabil läuft), gliedert sich der Hauptteil der Einführungsstunde in vier Phasen, die einem gemeinsamen Rhythmus folgen:

- Zunächst beschreibt der Lehrer einen Prozess, der in dieser Phase modelliert werden soll. Nur so ist gewährleistet, dass die einzelnen Phasen sinnvoll aufeinander aufbauen und sich der Schwierigkeitsgrad der Modellbildung langsam erhöht.

- Anschließend wird dieser Prozess durch ein bis zwei Akteure mit realen Gegenständen nachgespielt. Durch die Begrenzung der benötigten Materialien geschah dies im Plenum, was gleichzeitig sicherstellte, dass alle Schülerinnen und Schüler den gleichen Prozess vor Augen hatten, bevor sie mit der Modellierung begannen.
- Darauf aufbauend wird dann mit Hilfe des Software-Tools von allen Schülerinnen und Schülern ein Petrinetz modelliert, das den soeben gesehenen Prozess geeignet darstellen kann. Dadurch, dass die Schülerinnen und Schüler durch die beschränkte Anzahl von Computern partnerweise sitzen, ergab sich automatisch eine Partnerarbeit für diesen Schritt, die ich allerdings hier auch als die optimale Sozialform empfinde.
- Zuletzt wird neben der Ergebnispräsentation eines möglichen Modells ausgewertet, welche Erkenntnisse die Schülerinnen und Schüler bei der Modellierung des Prozesses gewonnen haben. Als gemeinsame Auswertung der Phase geschieht dies stets in einem Unterrichtsgespräch. Dadurch kann auch ansatzweise gewährleistet werden, dass alle Schülerinnen und Schüler mit dem gleichen Lernzuwachs in die nächste Phase starten.

Mit diesen Phasen folge ich, wie schon zuvor kurz erwähnt, grob den Repräsentationsformen nach Bruner. Das Nachspielen der Prozesse bedient die enaktive Komponente, die grafische Darstellung des Prozesses in Form eines Petrinetzes führt uns auf die ikonische Ebene und das Unterrichtsgespräch wechselt zum Schluss auf eine sprachliche Metaebene.

Im Folgenden werde ich nun die konkreten Aufgabenstellungen der einzelnen Phasen und die jeweilige Zielsetzung beschreiben. Da ich die Einführungsstunde auch praktisch in meinem Informatikkurs der Jahrgangsstufe 11 umgesetzt und mit den Schülerinnen und Schülern evaluiert habe, findet sich im Anschluss an jede Phase eine Reflexion dieser.

Evaluationsmethode

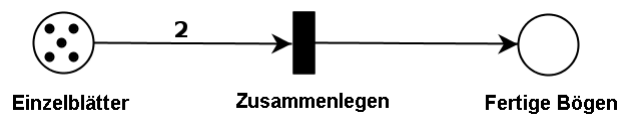
Um möglichst viele erste Eindrücke von den Schülerinnen und Schülern, die sich im Rahmen der Einführungsstunde ergaben, festzuhalten, habe ich mich bei der Evaluation der Einführungsstunde für eine jederzeit spontan einsetzbare Methode entschieden.

Zu Beginn der Stunde erhielt jede Schülerin und jeder Schüler jeweils zwei grüne und zwei rote Karten. Durch kurzes Hochhalten von zwei grünen Karten konnten die Schülerinnen und Schüler ihre starke Zustimmung, mittels einer grünen Karte

ihre schwache Zustimmung, mit einer roten eine schwache Ablehnung und mit Hilfe von zwei roten Karten ihre starke Ablehnung zu Thesen äußern, die ich in verschiedenen Phasen der Unterrichtsstunde äußerte. Auf eine neutrale Einstellung gegenüber einer These habe ich bewusst verzichtet, um zumindest in jedem Fall noch eine Tendenz ablesen zu können. Nachdem einige Schülerinnen und Schüler durch Hochhalten aller möglichen anderen Kartenkombinationen doch versuchten diese Rückzugsmöglichkeit zu wählen, habe ich diese Stimmen konsequenterweise als Enthaltungen gewertet, wodurch sich manche Ergebnisse nicht zur vollen Teilnehmerzahl ergänzen. Um eine gegenseitige Beeinflussung der Lerngruppe zu minimieren, sollten alle Schülerinnen und Schüler auf ein Signal hin gleichzeitig ihre Karten hochhalten, was jedoch nicht immer absolut zuverlässig funktionierte.

Phase 1

Aufgabenstellung: Für eine Klausur sollen einzelne Blätter zu Klausurbögen² zusammengelegt werden. Dabei soll jede Schülerin und jeder Schüler zwei (weiße) Blätter erhalten. Es stehen insgesamt fünf weiße Blätter zur Verfügung.



Die erste Phase unterscheidet sich insofern von den übrigen Phasen, als dass sowohl das Nachspielen des sehr simplen Prozesses, als auch die Darstellung des Prozesses in Form eines Petrinetzes vom Lehrer übernommen wird. Dadurch soll den Schülerinnen und Schülern veranschaulicht werden, was in den folgenden Phasen von ihnen erwartet wird. Gleichzeitig können so die wesentlichen Bedienelemente des Software-Tools erläutert werden.

Während das Markenspiel am Petrinetz vorgeführt wird, bleibt eine konkrete Beschriftung der Elemente zunächst aus. Unter der Fragestellung „Welche Elemente aus der Realwelt werden wie im Petrinetz modelliert?“ sollen die Schülerinnen und Schüler im folgenden Unterrichtsgespräch eine erste Transferleistung zwischen Realwelt und Modell erbringen. Sie erkennen, dass sich die verwendeten Gegenstände im Petrinetz als Marken wiederfinden, Tätigkeiten in Form von Transitionen dargestellt werden und Stellen einer Art Lagerplatz entsprechen. Zwar können die Schülerinnen und Schüler die Elemente aufgrund ihrer geometrischen Form bezeichnen, doch es wird schnell klar, dass hier geeignete Fachbegriffe fehlen, sodass sich an dieser Stelle auch die korrekte

² Der Begriff „Klausurbogen“ beschreibt hier nicht die doppelseitigen Bögen, sondern eine Zusammenstellung mehrerer Blätter für eine Klausur.

Terminologie von Petrinetzen problemorientiert einführen lässt. Zum Abschluss dieser Phase schlagen die Schülerinnen und Schüler geeignete Bezeichnungen für die unbeschrifteten Elemente vor.

Zwischenreflex und -evaluation

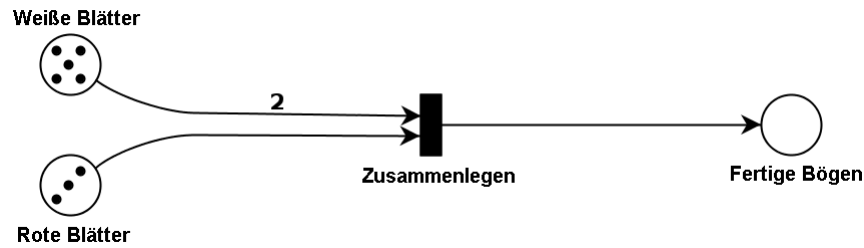
Man kann sich an dieser Stelle natürlich fragen, ob eine solch starke Lehrerzentrierung in dieser Phase stattfinden muss. Außer Frage steht jedoch, dass die Konstruktion eines Petrinetzes zumindest einmal von Lehrerseite aus vorgeführt werden muss, da den Schülerinnen und Schülern zu Anfang die Bedienung der Software schlicht unbekannt ist. Man könnte jedoch überlegen, ob man die Reihenfolge insofern vertauscht, als dass zuerst das Petrinetz vom Lehrer konstruiert wird, dann ausschließlich die Materialien zur Verfügung gestellt werden und schließlich der Prozess schon von Schülerseite nachgespielt wird. Dadurch könnte man eine höhere Aktivierung im folgenden Unterrichtsgespräch erzielen, da zumindest schon die Akteure eine entsprechende Transferleistung hätten erbringen müssen.

Direkt im Anschluss an diese Phase habe ich die Schülerinnen und Schüler meines Informatikkurses gefragt, ob der kurze Einstieg bereits ihr Interesse respektive ihre Neugier an Petrinetzen geweckt hätte. Ziel war es, festzustellen, ob der erste Eindruck, den die Schülerinnen und Schüler von Petrinetzen erhalten hatten, geeignet ist, die natürliche Neugier an dem neuen Themenfeld zu fördern oder ob sich nicht vielmehr direkt von Beginn eine ablehnende Haltung oder Blockaden einstellen. Im Gesamtbild zeigte sich eine positive, aber noch vorsichtige Einstellung gegenüber dem neuen Thema. Zur These „Das Thema hat mein Interesse geweckt“ äußerte nur ein Schüler starke Zustimmung, zwölf eine bedingte Zustimmung, zwei eine schwache Ablehnung und drei eine deutliche Ablehnung.

Man könnte die geringfügige Ablehnung darauf zurückführen, dass es keine Problematisierung zu Beginn der Einführungsstunde gibt und den Schülerinnen und Schülern noch nicht klar wird, wozu sie Petrinetze einsetzen können. Dies wird leider erst dann vollends klar, wenn Petrinetze als Modellierungswerkzeug in einem größeren Projekt eingesetzt werden.

Phase 2

Aufgabenstellung: Zusätzlich zu den zwei weißen Blättern soll jeder Schüler noch ein rotes Blatt erhalten, auf denen er sich Notizen anfertigen kann. Dafür werden ergänzend drei rote Blätter zur Verfügung gestellt.



Gegenüber der Aufgabenstellung in Phase 1 treten keine wesentlichen neuen Problemstellungen auf. Es ist daher zu erwarten, dass sowohl das Nachspielen des Prozesses als auch die Modellierung mit Hilfe eines Petrinetzes keine ernsthaften Schwierigkeiten für die Schülerinnen und Schüler erzeugt. Dies ist gewollt, damit sie das soeben nur passiv miterlebte nun auch aktiv an einem weiteren, einfachen Beispiel anwenden können und um gleichzeitig ein erstes Erfolgserlebnis für möglichst viele Schülerinnen und Schüler zu garantieren.

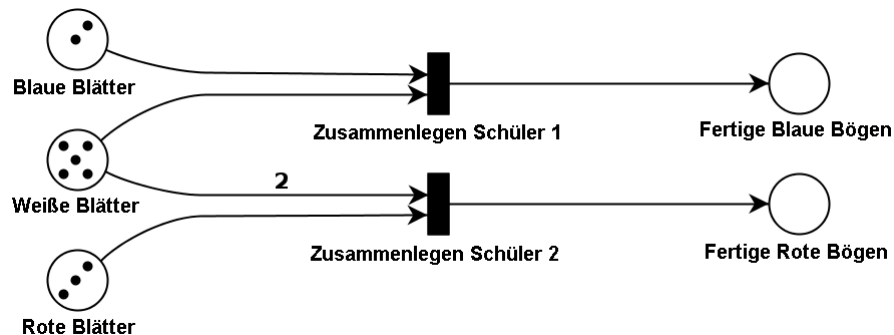
Gleichwohl können die Schülerinnen und Schüler hier einen ersten Nutzen in der Modellierung erkennen. Durch Erstellen eines geeigneten Petrinetzes und Simulation dessen kann vorausgesagt werden, wie viele fertige Bögen maximal erstellt werden können und ob gegebenenfalls Reste übrig bleiben, obgleich diese einfache Erkenntnis natürlich auch ohne Petrinetz nachvollziehbar wäre. Der Austausch über den absehbaren Nutzen von Petrinetzen sollte daher die Grundlage für das abschließende Unterrichtsgespräch der zweiten Phase bilden.

Zwischenreflexion

Wie erwartet, stellte diese Aufgabe die Schülerinnen und Schüler meines Informatikkurses vor keine ernsthaften Probleme, gab aber insbesondere auch schwächeren Schülerinnen und Schülern noch einmal Gelegenheit sich aktiv mit der Bedienung des Software-Tools auseinanderzusetzen. Im Unterrichtsgespräch wurden von Schülerseite erste (naive) Ideen für den Nutzen von Petrinetzen genannt.

Phase 3

Aufgabenstellung: Gleichzeitig soll ein zweite Person Klausurbögen für eine andere Klausur zusammenlegen. Hier werden als Notizzettel jedoch blaue Blätter verwendet. Außerdem wird jeweils nur ein einzelnes weißes Blatt pro Klausurbogen benötigt. Als Ressource stehen zusätzlich zwei blaue Blätter zur Verfügung.



In der dritten Phase sind zum ersten Mal zwei Akteure gleichzeitig beschäftigt, was in der Terminologie der Informatik einer Nebenläufigkeit entspricht. Zudem entsteht eine Konkurrenzsituation zwischen zwei Akteuren, da sie beide auf eine gemeinsame Ressource, die weißen Blätter, zugreifen. Ganz abhängig davon, in welcher Geschwindigkeit die beiden Akteure arbeiten, entstehen völlig verschiedene Endresultate. Arbeitet zunächst ausschließlich Schüler 1, so kann er insgesamt zwei blaue Bögen zusammenlegen bevor ihm die blauen Blätter ausgehen, während Schüler 2 aus den Resten noch einen roten Bogen zusammenlegen kann, bevor für seinen Arbeitsprozess nicht mehr genügend weiße Blätter vorrätig sind. Ist dagegen zunächst nur Schüler 2 aktiv, so kann er zwei rote Bögen zusammenlegen, während Schüler 1 nur noch einen blauen Bogen zusammenlegen kann.

Während die eigentliche Modellierung des Petrinetzes wieder keine größeren Probleme bereiten sollte, sind die Schülerinnen und Schüler hier zum ersten Mal aufgefordert, das Markenspiel zielgerichtet einzusetzen. Während im Realspiel nur ein möglicher Prozessablauf aufgezeigt werden kann, können mittels des Markenspiels viele verschiedene Abläufe simuliert werden. Die Schülerinnen und Schüler sind daher neben der eigentlichen Modellierung dazu angehalten, sich Gedanken darüber zu machen, welche Fragestellungen mit Hilfe der Simulation des Petrinetzes beantwortet werden können. Mögliche Fragestellungen wären:

- Wie viele verschiedene mögliche Prozessabläufe gibt es?
- Welcher Prozessablauf erzeugt die meisten roten bzw. blauen Bögen?
- Welcher Prozessablauf hinterlässt die geringsten Reste an Blättern, respektive gibt es einen Ablauf, der überhaupt keine Reste hinterlässt?

Zwischenreflexion

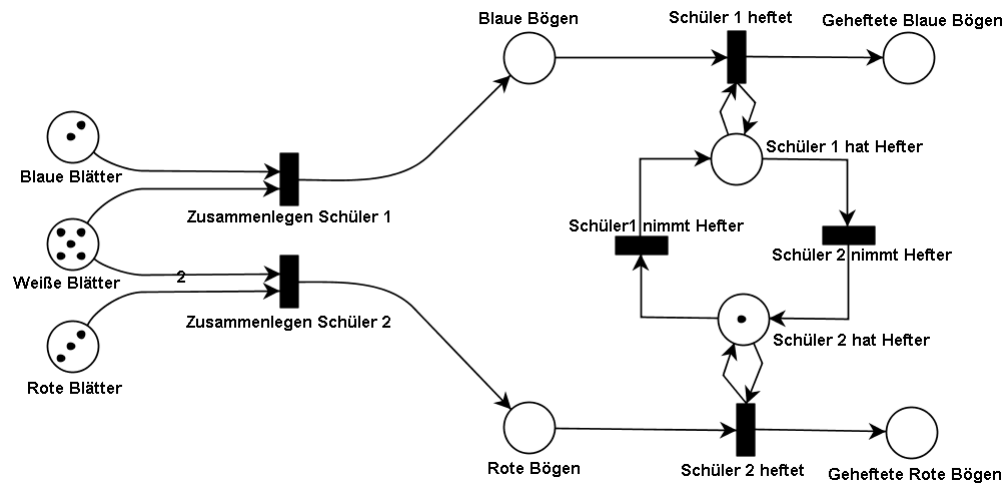
Schon beim Nachspielen des Prozesses zeigten sich die Akteure über die Konkurrenzsituation kurzzeitig überrascht und versuchten dann instinktiv Strategien zu entwickeln, um ihre Abläufe insofern möglichst effizient zu

gestalten, als dass sie möglichst viele Bögen herstellen und wenig Blätter übrig behalten wollten. Da die Diskussion der beiden Akteure bereits die oben genannten Fragen aufwarf, habe ich kurzfristig das Unterrichtsgespräch direkt an das Spiel angeschlossen, um die Gruppe an dem Streitgespräch der beiden Akteure zu beteiligen.

Da die anschließende Modellierung nun kaum Neues mit sich brachte und viele Fragestellungen bereits im vorgeschalteten Unterrichtsgespräch vorweggenommen wurden, war die Motivation insbesondere bei den stärkeren Schülerinnen und Schülern hier etwas eingebrochen. Da von der dritten zur vierten Phase ein deutlicher Anstieg der Modellkomplexität stattfindet, würde ich daher im Nachhinein in Kursen mit höherem Leistungsniveau eher die zweite Phase ersatzlos streichen, um mit der dritten Phase genügend Neuerungen einzuführen.

Phase 4

Aufgabenstellung: Damit die zusammengelegten Klausurbögen nicht später unbeabsichtigt auseinanderfallen, sollen die Bögen nach dem Zusammenlegen durch die gleiche Person geheftet werden. Dazu steht jedoch nur ein Hefter für beide Personen zur Verfügung.



Die vierte und letzte Phase soll den Schülerinnen und Schülern nun Gelegenheit bieten das soeben Erlernete an einem herausfordernden Beispiel selbstständig umzusetzen. Obwohl der eigentliche Prozess nur um das Heften erweitert wurde, erhöht sich die Komplexität des Modells deutlich, da nun eine Ressource existiert, die nicht verbraucht werden kann, aber nur einmal innerhalb des Systems zur Verfügung steht. Hält ein Schüler den Hefter in Händen, so kann der

andere nicht mehr heften, wohl aber noch Blätter zusammenlegen, es sei denn die benötigten Blätter stehen nicht mehr in ausreichendem Maße zur Verfügung. Der andere Schüler wiederum kann natürlich heften, aber auch den Hefter einfach nur festhalten und ebenfalls Blätter zusammenlegen.

Wichtigstes, neues Element dieses Modells ist die Schleife, das heißt, eine Transition hat die selbe Stelle sowohl im Vorbereich, als auch im Nachbereich. Dadurch kann die Transition einerseits nur dann schalten, wenn sich in der Schleifenstelle eine Marke befindet, die entsprechende Marke wird andererseits aber im gleichen Schritt verbraucht als auch erzeugt, sodass sie letztlich nicht verloren geht. Ich erwartete, dass das selbstständige Auffinden der Schleifenidee das größte Problem dieses Modells darstellt und hoffte, dass die Schülerinnen und Schüler ein zielgerichtetes Trial-and-Error-Verfahren anwenden würden, um sich einer geeigneten Lösung schrittweise anzunähern.

Zwischenreflexion

Schon beim Nachspielen des Prozesses begannen die Akteure wieder den Prozess auf eine bestmögliche Strategie zu analysieren, aber auch der Begriff der Fairness stand bereits durch den „Streit“ um den Hefter im Raum. Ein Schüler merkte an, dass ein Akteur den Prozess nie zu Ende bringen könnte, wenn ihm der andere Akteur den Hefter vorenthalten würde.

Das Modellieren des zugehörigen Petrinetzes stellte wie erwartet eine große Herausforderung für alle Schülerinnen und Schüler gleichermaßen dar. Hier zeigte sich erstmals eine deutliche Differenzierung der Herangehensweise. Einige Schülerinnen und Schüler setzen direkt ihre Vorstellungen als Petrinetz um, simulierten nur sehr halbherzig und gaben sich mit der Lösung zufrieden, auch wenn diese bei näherer Betrachtung den Prozess nicht hinreichend genau beschrieb. Andere Schülerinnen und Schüler waren zwar motiviert, simulierten viele verschiedene Modelle durch und konnten gute Ansätze finden, kamen aber letztlich nicht auf die Idee der Schleifen. Um so mehr waren sie aber zum Abschluss der Unterrichtsstunde an den Lösungen der Anderen interessiert. Den Schülerinnen und Schüler, die ein vollständiges Modell fanden, (etwa ein Viertel des Kurses) stand das Erfolgserlebnis dagegen deutlich ins Gesicht geschrieben.

Hervorzuheben von meinen Beobachtungen während dieser Modellierungsphase ist insbesondere, dass die Schülerinnen und Schüler, die am Ende ein funktionierendes Modell vorweisen konnten, vorwiegend aus dem mittleren Leistungsbereich stammten, während die Schülerinnen und Schüler aus dem oberen Leistungsspektrum eher zu der Gruppe gehörten, die sich vorschnell mit ihrem modellierten Petrinetz zufrieden gaben.

Schlussevaluation

In der Schlussevaluation der Unterrichtsstunde wollte ich von den Schülerinnen und Schülern wissen, ob sie nun im Gesamtbild der Stunde betrachtet, das Thema spannend finden würden. Das Ergebnis fiel noch etwas positiver als das der Zwischenevaluation aus (3x starke Zustimmung, 9x schwache Zustimmung, 4x schwache Ablehnung), sodass ich die Einführungsstunde zumindest aus motivatorischer Sicht der Schülerinnen und Schüler als erfolgreich bezeichnen möchte. Außerdem frug ich, ob die Schülerinnen und Schüler das Modellieren von Petrinetzen aus ihrer persönlichen Sicht als einfach empfänden. Obwohl viele Schülerinnen und Schüler an der letzten Herausforderung (zumindest dem Endresultat, nicht dem Lernzuwachs nach) gescheitert sind, wurde diese Frage überaus positiv beantwortet (11x starke Zustimmung, 1x schwache Zustimmung, 3x schwache Ablehnung).

Petrinetze als Modellierungswerkzeug

Nachdem mir mit der Einführungsstunde offenbar eine erfolgreiche Einführung des Modellierungswerkzeugs „Petrinetz“ gelungen war, will ich in diesem Abschnitt den zwei noch offenen Fragestellungen näher nachgehen. Dazu bettete ich Petrinetze kurze Zeit später in ein größeres Projekt ein.

Projektbeschreibung

In der Abschlussevaluation des ersten Halbjahres äußerten die Schülerinnen und Schüler meines Kurses, dass sie sich für das kommende Halbjahr ein Programmierprojekt wünschten, an dem sie selbstständig über mehrere Unterrichtsstunden hinweg arbeiten könnten. Außerdem hatte ich im ersten Halbjahr beobachtet, dass viele Schülerinnen und Schüler Schwierigkeiten damit hatten, sich auf die objektorientierte Programmierung einzulassen. Das war nur allzu verständlich, da die meisten von ihnen in der Differenzierungsstufe maßgeblich die prozedurale Programmierung erlernt hatten und die Miniprojekte des ersten Halbjahres keine Vorteile der objektorientierten Programmierung aufzeigen konnten. Ich wollte daher den Wunsch der Schülerinnen und Schüler gleichzeitig dafür nutzen, die Kernidee der Objektorientierung etwas greifbarer zu machen. Mein Gedanke war daher innerhalb eines Projektes wesentliche Elemente eines auf Objektorientierung aufbauenden Software-Engineering-Prozesses zu durchlaufen.

Ziel des Projektes war die Simulation einer Fußgänger-Bedarfsampelanlage, bestehend aus zwei Kraftfahrzeugampeln, zwei Fußgängerampeln, zwei Anforderungstastern für die Fußgänger und einer zentralen Ampelsteuerung. An einem Realbeispiel arbeiteten die Schülerinnen und Schüler die soeben genannten Einzelobjekte heraus. Anschließend beschrieben sie die Objekte durch ihre Attribute näher und legten Methoden fest, die die einzelnen Objekte zur Verfügung stellen müssten. Daraus erstellten sie schließlich ein geeignetes UML-Klassendiagramm.

Nachdem das Klassendiagramm einige Revisionen erfahren hatte, wurden die Objekte (mit Ausnahme der Ampelsteuerung) arbeitsteilig implementiert. Eine Gruppe übernahm die Fußgängerampel, da sie besonders leicht zu implementieren war, und jeweils zwei Gruppen die KFZ-Ampel und den Anforderungstaster. Dabei stellten sich jedoch erhebliche Schwierigkeiten ein, die ich deshalb an dieser Stelle darlege, da sie für diverse Anfangsprobleme bei der Realisierung der Ampelsteuerung sorgten. So bereitete beim Anforderungstaster eine Restriktion der verwendeten Programmiersprache Delphi 5 Schwierigkeiten, da man mit dem gewöhnlichen Klassenkonzept aus eingebundenen Units nicht auf Methoden von Klassen zugreifen kann, die in der Eltern-Unit implementiert sind. Nachdem ein Kollege einen Workaround dafür gefunden hatte, konnte eine Gruppe die Klasse mit erheblicher Verzögerung, die andere Gruppe schließlich gar nicht fertigstellen. Zwei Gruppen mit unterschiedlichen Klassen speicherten diese versehentlich unter dem gleichen Unit-Namen, was erst beim Einbinden in die Ampelsteuerung auffiel. Eine Gruppe hatte Bilder in ihre Klasse eingebunden, aber nicht als Ressource zur Verfügung gestellt.

Einsatz von Petrinetzen innerhalb des Projekts

Nach der Erstellung der drei oben genannten Klassen, musste nun noch eine geeignete Ampelsteuerung entworfen werden, die die übrigen Klassen geeignet ansteuert. Wie ich bei der Ausarbeitung einer Musterlösung der Ampelsteuerung feststellen musste, ist dieser Steuerungsprozess keinesfalls trivial, da ab einem genau zu definierenden Zeitpunkt ein nebenläufiger Prozess einsetzen muss, der später wieder zusammengeführt wird. Die im nächsten Kapitel vorgestellte Musterlösung zeigt die erwarteten Schwierigkeiten an diesem Steuerungsprozess auf.

Da sich solche parallelen Prozesse wie zuvor festgehalten optimal mit Petrinetzen modellieren lassen, konnten diese hier sinnvoll im Entwicklungsprozess eingebunden werden. Um eine differenzierte Evaluation und Reflexion abzielend auf die zuvor aufgeworfenen Fragestellungen zu ermöglichen, bildete ich drei

Gruppen, die Petrinetze in ganz unterschiedlichem Umfang einsetzen: Gruppe 1 sollte Petrinetze überhaupt nicht einsetzen, sondern nach eigenem Ermessen die Ampelsteuerung implementieren. Gruppe 3 sollte die Ampelsteuerung zunächst mit Hilfe eines Petrinetzes modellieren und anschließend auf dieser Grundlage den Steuerungsprozess selbstständig implementieren.

Gruppe 2 schließlich sollte ähnlich wie Gruppe 3 zunächst ein Petrinetz modellieren. Anschließend gab ich dieser Gruppe jedoch konkrete Tipps, wie sich daraus ein Grundgerüst für die folgende Delphi-Implementierung ableiten ließe. In schülergerechter Form zeigte ich ihnen auf, dass Stellen als Variablen aufzufassen sind und Transitionen die zuvor programmierten Methoden repräsentieren. Ein regelmäßiger Prozess (in Delphi das Timer-Objekt) müsse in regelmäßigen Abständen für jede Transition respektive Methode prüfen, ob die Stellen im Vorbereich, sprich die Variablen, im erforderlichen Maße markiert sind, und im positiven Fall die Methode ausführen und anschließend Marken bei den Stellen im Nachbereich erzeugen, sprich die Variablen neu setzen. Man merkt schnell, dass diese Programmierung nicht sonderlich effizient ist, doch ich erwartete so die bestmögliche Übertragung von Petrinetz auf die fertige Implementierung.

Die jeweilige Zuteilung der Schülerinnen und Schüler habe ich ihnen selbst überlassen. Sie sollten selbst einschätzen, von welcher Gruppe sie sich am ehesten Erfolg versprechen. Nicht unerwähnt sollte jedoch bleiben, dass die Wahl in nicht geringem Umfang auch von gruppenspezifischen Kriterien abhing.

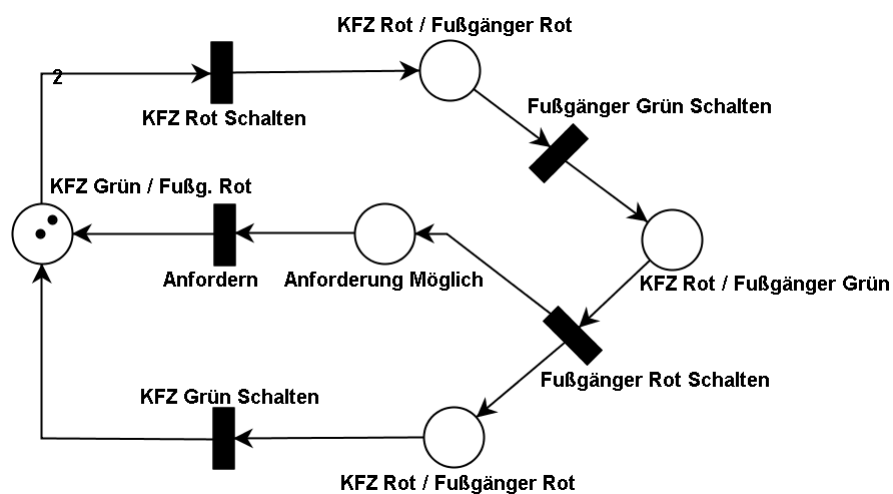
Musterlösung

Das zu entwerfende Modell besitzt zwei grundlegende Kernprobleme. Im Gegensatz zu den Beispielen in der Einführungsstunde gibt es bei der Ampelsteuerung keine konkreten Gegenstände, die „umgelagert“ werden. Es mag nahe liegen die Stellen des Petrinetzes mit den einzelnen Lampen der Ampel zu identifizieren und die Marken als Ein-/Aus-Zustand zu interpretieren, doch würde dies zum einen das Modell erheblich aufblähen, zum anderen wäre dieses Modell nicht mit dem festgelegtem Klassendesign vereinbar, bei dem die Ampelsteuerung keinen direkten Zugriff auf die einzelnen Lampen der Ampeln besitzt. Vielmehr sind die Stellen hier abstrakter als Phase respektive Zustand zu interpretieren, in denen sich die Ampelsteuerung gerade befindet. Das Petrinetz bekommt dadurch zunächst den Charakter eines Automaten.

Als zweites Kernproblem ist jedoch die dann nicht mehr mit einem Automaten modellierbare, zeitweise Nebenläufigkeit des Prozesses hervorzuheben, die durch den Anforderungstaster provoziert wird. Haben die Fußgänger gerade grün oder

ist die Grünphase bereits eingeleitet, müssen Anforderungen ignoriert werden. Sobald die Fußgänger wieder rot erhalten, muss eine Anforderung wieder registriert, darf aber noch nicht umgesetzt werden, bevor die Kraftfahrzeuge keine Grünphase erhalten haben. In der Grünphase der Kraftfahrzeuge sollte eine Anforderung dagegen unmittelbar umgesetzt werden. Generell gilt zudem, dass bei einer gesetzten Anforderung, die noch nicht umgesetzt wurde, weitere Anforderungen zu ignorieren sind.

Ein mögliches Modell, das beide Kernprobleme geeignet umsetzt, zeigt die folgende Musterlösung.



Evaluationsmethode

Nach Abschluss der gesamten Projektarbeit erhielt jede Schülerin und jeder Schüler einen Fragebogen, der im Anhang zu finden ist. Der Fragebogen war anonym auszufüllen, ließ aber eine genaue Zuordnung zur Gruppe zu. Der Fragebogen wurde von insgesamt 17 Schülerinnen und Schülern ausgefüllt und abgegeben.

Gruppe 1

In Gruppe 1 befand sich der Kursbeste, der unmittelbar bei Projektstart mit der Implementierung der Ampelsteuerung begann, ohne irgendwelche Vorüberlegungen anzustellen. Die anderen Gruppenmitglieder verfielen zusehend in eine passive Haltung und übernahmen allenfalls Hilfstätigkeiten während der Projektarbeit. Die Implementierung zeichnete sich demnach dadurch aus, dass sie maßgeblich auf Ideen und Vorstellungen einer Einzelperson aufbaute. Insbesondere passte die Implementierung aber nicht mehr zu den in der vorherigen Gruppenarbeit angefertigten Objekten. Um die Funktionsfähigkeit

der Ampelsteuerung doch noch herzustellen, wurden kurzerhand die bereits fertigen Objekte den eigenen Vorstellungen angepasst, was auch gelang, aber mit der Idee der objektorientierten Arbeitsteilung nichts mehr gemein hatte. Die fertige Simulation war demzufolge mehr als eine komplette Neuprogrammierung zu betrachten.

Obwohl diese Gruppe am schnellsten mit ihrer Implementierung fertig war und das Endergebnis schlussendlich die einzig umfassend funktionierende Simulation war, scheiterte die Simulation genau an den Problempunkten, dessen sich die anderen Gruppen, die Petrinetze zur Modellierung einsetzten, wie später gezeigt, stets bewusst waren. Das Simulationsprogramm funktionierte nur solange, wie der Anwender die erwarteten Aktionen durchführte, also den Anforderungstaster erst dann betätigte, wenn sich die Ampelsteuerung im Grundzustand (KFZ-Ampel grün, Fußgängerampel rot) befand. Drückte man den Anforderungstaster in einem anderen Zustand, so entstanden entweder ungewollte Zustände (z.B. KFZ-Ampel und Fußgängerampel gleichzeitig grün) oder das Programm stürzte einfach ab.

Nachdem ich der Gruppe diese Probleme aufgezeigt hatte, dachten die Gruppenmitglieder zwar kurzfristig über eine geeignete Modifizierung der Ampelsteuerung nach, empfanden dies schließlich aber als kein gravierendes Problem der Ampelsteuerung und stellten sich so mit der bisherigen Lösung zufrieden, obwohl noch genügend Zeit bestanden hätte, sich mit diesen Problemen näher zu befassen.

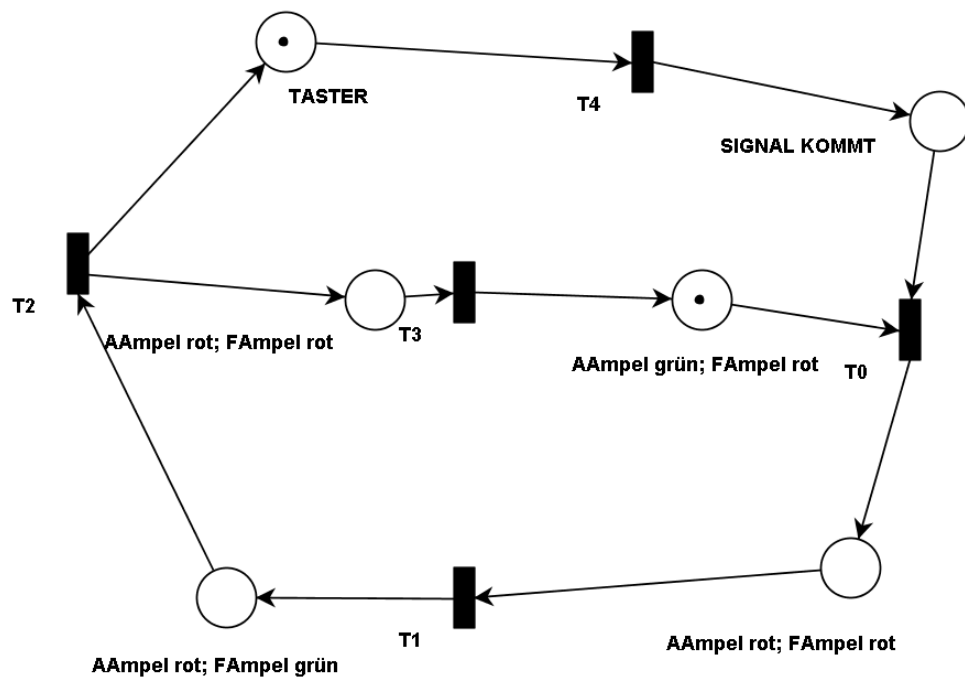
Evaluation der Gruppenarbeit

In der Abschlussevaluation wurden die fünf Gruppenmitglieder gefragt, ob sie diese Gruppe insbesondere deswegen gewählt hätten, weil sie sich nicht vorstellen konnten, dass ihnen Petrinetze bei der Implementierung der Ampelsteuerung hilfreich sein könnten. Diese These wurde klar abgelehnt, viermal schwach, einmal stark. Außerdem wollte ich wissen, ob sie im Anschluss an die Projektarbeit das Gefühl gehabt hätten, dass ihnen Petrinetze bei der Bewältigung der Aufgabe doch hätten dienlich sein können. Zwei äußerten eine schwache Zustimmung, einer eine schwache Ablehnung und zwei eine starke Ablehnung dieser These.

Diese Reaktion ist bemerkenswert. Einerseits glaubte die Gruppe anscheinend eine ungefähre Vorstellung zu haben, wie sie Petrinetze hätten einsetzen können, andererseits glaubten sie nicht, dass ihnen die Modellierung mit Petrinetzen weitergeholfen hätte und das obwohl am Ende der Projektarbeit die Simulation Probleme aufwies, für die sie ohne Weiteres keine geeignete Lösung fanden.

Gruppe 2

Gruppe 2 gelang es äußerst schnell und selbstständig die Realsituation in ein geeignetes Petrinetz zu übertragen. Der Abstraktionsschritt von konkreten Gegenständen der Einführungsstunde zu Zustandstoken gelang überraschend intuitiv. Das Problem der zeitweisen Nebenläufigkeit der Anforderung und der Ampelphasen wurde alsbald erkannt und durch ein zielgerichtetes, größtenteils selbstständiges Trial-and-Error-Verfahren gelöst, indem verschiedene Varianten vollständig durchgespielt wurden.



Das Endergebnis des Petrinetzes entspricht annähernd der Musterlösung, mit dem Unterschied einer zusätzlichen Stelle, sodass die Transitionen T3 und T4 hier nicht den gleichen Nachbereich besitzen, sondern ihre ausgehenden Kanten zu zwei verschiedenen Stellen führen. Das führt dazu, dass der Zustand der Ampelphase deutlicher von dem Zustand der Anforderung abgegrenzt wird und man zudem auf gewichtete Kanten in diesem Modell verzichten kann. Daher muss ich gestehen, dass dieses Petrinetz die Realsituation noch geeigneter und einfacher modelliert, als meine zuvor angefertigte Musterlösung.

Nachdem die Gruppe nun in kurzer Zeit ein hervorragendes Petrinetz modelliert und obwohl sie die vorgesehenen Hilfen auf dem Weg vom Petrinetz zur Delphi-Implementation erhalten hatte, war die Gruppe ratlos, wo und wie sie mit der Realisierung beginnen sollte. Als man nach einiger Zeit immer noch keinen

geeigneten Ansatz gefunden hatte, ignorierte die Gruppe das Petrinetz größtenteils und begann damit die Ampelsteuerung nach ihren eigenen Vorstellungen zu implementieren. Auch das gelang jedoch aufgrund von Schwächen in der Delphi-Programmierung nicht, sodass die Gruppe nur einen halbfertigen Ansatz der Implementierung am Ende des vorgegebenen Projektzeitraumes vorweisen konnte.

Evaluation der Gruppenarbeit

Alle vier Gruppenmitglieder stimmten in der abschließenden Evaluation der These zu, die Gruppe insbesondere deshalb gewählt zu haben, weil sie anfangs überzeugt waren, dass Petrinetze ihnen bei der Bewältigung der Aufgabe weiterhelfen könnten (1x starke Zustimmung, 3x schwache Zustimmung). Auch der These, dass es einfach gewesen sei ein passendes Petrinetz für die Ampelsteuerung zu entwerfen, stimmten die Gruppenmitglieder überwiegend zu (1x starke Zustimmung, 2x schwache Zustimmung, 1x schwache Ablehnung).

Die These, dass der Schritt vom Petrinetz zur Delphi-Implementierung dann leicht gefallen wäre, lehnten die Schüler wie zu erwarten überwiegend ab (1x schwache Zustimmung, 3x schwache Ablehnung). Ob die Hilfen durch den Lehrer in diesem Zusammenhang geeignet gewesen wären, wurde dagegen von jedem Mitglied der Gruppe anders bewertet, sodass alle vier Wahlmöglichkeiten auf dem Fragebogen abgedeckt waren.

Die Evaluation spiegelt meines Erachtens genau das Bild wieder, das ich auch bereits während der Projektarbeit beobachten konnte. Der Schritt von der Realsituation zum Modell fiel leicht, der Schritt vom Modell zur Implementation schien der Gruppe auf dem vorgeschlagenen Weg dagegen unmöglich.

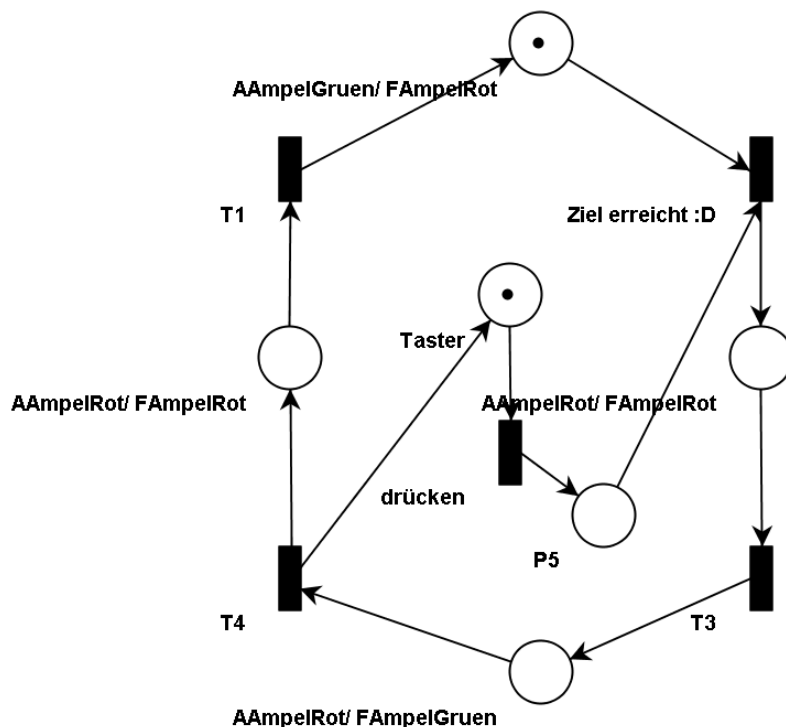
Gruppe 3

Da sich sehr viele Schülerinnen und Schüler spontan für Gruppe 3 entschieden hatten, habe ich diese Gruppe in zwei eigenständige Teilgruppen aufgeteilt, wobei die genaue Gruppeneinteilung von Schülerseite bestimmt wurde.

Gruppe 3a

Gruppe 3a hatte zunächst kleinere Schwierigkeiten beim Abstraktionsprozess, kam aber mit kleinen Tipps ebenfalls auf den richtigen Weg und erkannte das Problem der zeitweisen Nebenläufigkeit schließlich ohne weitere Hilfen. Beim Erarbeiten einer geeigneten Lösung dafür hatten sie zwar viele Ideen, waren aber nicht in der Lage diese eigenständig umfassend anhand des Markenspiels zu

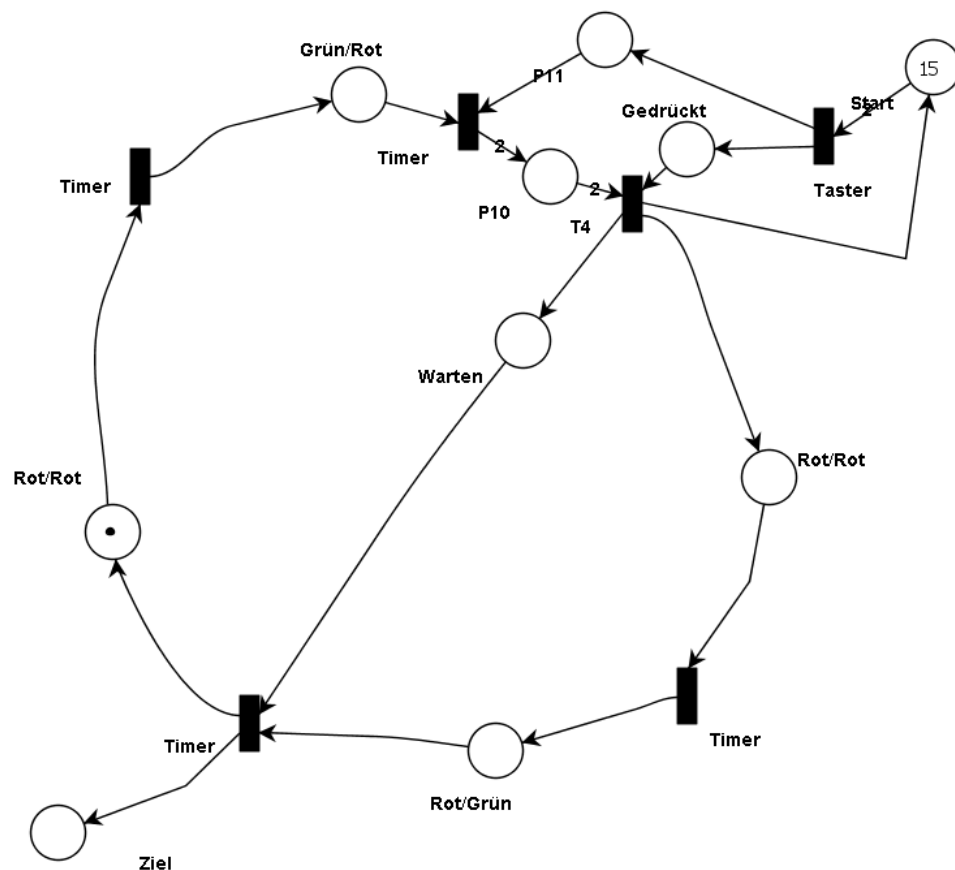
validieren. Funktionierte das Modell für eine bestimmte Abfolge von Aktionen, so betrachtete die Gruppe die Ampelsteuerung als zufriedenstellend modelliert. Zeigte ich dann eine Aktionsreihenfolge auf, die die Ampelsteuerung in einen ungeeigneten Zustand führte, begann die Gruppe mit durchaus großer Motivation nach einer neuen Lösung zu suchen. Erst mit der Zeit begann diese Gruppe die Netze in ausreichendem Maße selbstständig zu validieren. Insgesamt ähnelt das Trial-and-Error-Verfahren dem Vorgehen von Gruppe 2, wenngleich diese Gruppe anfangs weniger selbstständig und weniger zielgerichtet vorging.



Auch das Endresultat des Petrinetzes ist bis auf die Anordnung und die Beschriftung der Knoten identisch mit dem von Gruppe 2. Aufgrund von technischen Problemen gelang es der Gruppe nicht das Petrinetz in eine funktionierende Ampelsteuerung umzusetzen. Beim Versuch die Klassen der Mitschüler in die Ampelsteuerung zu integrieren, stürzte Delphi ab, was hingegen bei den anderen Gruppen kein Problem darstellte. Die Gruppe unternahm den Versuch das Projekt noch einmal komplett neu aufzusetzen, allerdings war das Problem stets reproduzierbar. Die Gruppe konnte am Ende zwar ein theoretisches Konzept der Implementierung aufzeigen, doch ohne dieses auszutesten konnten die Gruppenmitglieder Schwachpunkte dieser Implementation nicht selbstständig erkennen.

Gruppe 3b

Gruppe 3b hatte bereits mit der Modellierung erhebliche Probleme, da sie ganz offensichtlich versuchten die Realsituation so genau und so vollständig wie irgend möglich abzubilden, anstatt eine geeignete Auswahl dessen vorzunehmen, was für das Modell der Ampelsteuerung relevant ist, wie es den anderen Gruppen mühelos gelang. So begannen sie zunächst ein Petrinetz zu modellieren, das veranschaulicht wie eine Gruppe von Menschen die Fahrbahn überquert. Die Stellen repräsentierten hier die jeweilige Position, an denen sich die Menschen befanden, jede Marke stand für einen Menschen. Nachdem die Modellierung mehr und mehr in die falsche Richtung geriet, versuchte ich den Fokus der Gruppe wieder auf Elemente zu fokussieren, die für die Ampelsteuerung entscheidend waren. Mir gelang es jedoch nicht die Gruppe davon zu überzeugen, ein neues Modell zu erstellen; stattdessen versuchten die Schülerinnen und Schüler das Modell zu ergänzen, anscheinend in dem irrigen Glauben, dass ein komplexeres Modell eine bessere Leistung darstelle. Zwar arbeitete die Gruppe weiterhin sehr selbstständig und erkannte zielsicher die jeweiligen Schwächen in ihrem Modell, doch war dieses mit der Zeit so komplex



geworden, dass sie es eigenständig gar nicht mehr vollständig überblicken und validieren konnten.

Die Gruppe erfuhr immer mehr Frustration dadurch, dass für sie Petrinetze sie zu sehr in ihrer Modellbildung einschränkten, da sie damit ihnen wichtige Details nicht darstellen konnten. Dies äußerte sich insbesondere in den fortwährenden Nachfragen an mich, ob es nicht geeignete Erweiterungen eines Petrinetzes gäbe.

Obwohl diese Gruppe den Großteil ihrer Projektarbeitszeit an dem Petrinetz verbrachte, war das Endergebnis nicht zufriedenstellend. Das Modell kann auf verschiedene Weisen in eine ungewollte Markierung geführt werden und manche Elemente scheinen redundant modelliert worden zu sein. Auch bemerkenswert ist, dass das Modell den Eindruck vermittelt, jede Person müsse den Anforderungstaster separat betätigen und würde dann im Raum vor der Fußgängerampel „zwischengelagert“, da mit jeder Ampelphase nur eine Person die Fahrbahn überqueren kann. Eine Implementierung des Petrinetzes in ein Delphiprogramm wurde zwar begonnen, war aber aufgrund der nur noch sehr beschränkten Zeit nur im Ansatz fertig geworden.

Evaluation der Gruppenarbeit

Auch von diesen insgesamt acht Gruppenmitgliedern wollte ich wissen, ob sie die Gruppe deshalb gewählt hatten, weil sie sich vorstellen konnten, das ihnen Petrinetze auf dem Weg zur fertigen Implementierung helfen könnten, was ausschließlich bejaht wurde (1x starke Zustimmung, 7x schwache Zustimmung). Auf die Frage, ob sie die Modellierung des Petrinetzes als einfach empfanden, äußerten sich alle Schülerinnen und Schüler äußerst positiv, was vor dem Hintergrund der Schwierigkeiten insbesondere in Gruppe 3b überrascht (2x starke Zustimmung, 6x schwache Zustimmung).

Weniger überraschend ist die Tatsache, dass sämtliche Gruppenmitglieder die These ablehnten, dass ihnen die Modellierung eines Petrinetzes für die konkrete Implementierung des Delphiprogramms geholfen hätte (6x schwache Ablehnung, 2x starke Ablehnung). Scheinbar konträr dazu äußerten sich die Schülerinnen und Schüler überwiegend zustimmend zur These, dass auch im Nachhinein Petrinetze sinnvoll für die Entwicklung der Ampelsteuerung erschienen (5x schwache Zustimmung, 1x schwache Ablehnung, 2x starke Ablehnung). Dies ist vermutlich so zu werten, dass Petrinetze durchaus als Hilfe bei der Implementierung empfunden wurden, aber nicht eine direkte Schnittstelle zur Programmierung herstellen konnten.

Schlussbetrachtungen

Ob sich Petrinetze erfolgreich im Informatikunterricht einsetzen lassen, kann sicherlich nicht global mit Ja oder Nein beantwortet werden. Daher möchte ich vielmehr zunächst kurz eine Einschätzung des in dieser Arbeit nicht näher betrachteten theoretisch-analytischen Ansatzes geben und dann detailliert auf die Thesen eingehen, die ich zu Beginn des unterrichtspraktischen Versuchs im Zusammenhang mit dem modellierenden Ansatz aufgestellt habe.

In wie fern sich die Symbiose aus parallelen Konzepten veranschaulicht durch Petrinetze als Ergänzung der theoretischen Informatik in der Qualifizierungsphase einsetzen lässt, war wie erwähnt nicht Schwerpunkt dieser Arbeit und kann daher nur ansatzweise beantwortet werden. Während der Einführungsstunde war ich jedoch ziemlich überrascht, dass die vorgestellten Beispiele zumindest einige Schülerinnen und Schüler spontan zu ersten über die Modellierung hinausgehenden, analytischen Untersuchungen einluden. In der Schlussevaluation frug ich die Schülerinnen und Schüler, ob sie sich auch eine kurze Unterrichtsreihe vorstellen könnten, bei der man Petrinetze weitaus tiefergehend und theoretisch fundiert analysieren würde. Obgleich viele Schülerinnen und Schüler per se eine Abneigung gegenüber allzu theoretischen Inhalten haben (vergleiche Schülerkommentar 3 im Anhang), gab es auch zahlreiche positive Stimmen zu einer solchen Unterrichtsreihe (6x starke Zustimmung, 3x schwache Zustimmung, 5x schwache Ablehnung, 3x starke Ablehnung).

Natürlich muss man hier etwas vorsichtig mit allzu konkreten Aussagen sein, da sowohl ich, aber noch vielmehr die Schülerinnen und Schüler, ohne eine solche Unterrichtsreihe praktisch durchgeführt zu haben keine differenzierte Erwartungshaltung abgeben können. Doch es scheint mir, als wäre eine solche Unterrichtsreihe durchaus einen Versuch wert.

Lässt sich der Umgang mit Petrinetzen als Modellierungswerkzeug leicht erlernen?

Diese Frage lässt sich nach der erfolgreichen Einführungsstunde sowohl aus der Sicht der Schülerinnen und Schüler als auch aus meinen Beobachtungen heraus deutlich mit Ja beantworten. Das gesetzte Zeitlimit von 90 Minuten reichte aus, um die Verwendung von Petrinetzen als Modellierungswerkzeug in ausreichendem Maße zu erlernen. Dies zeigt sich insbesondere darin, dass mit Ausnahme einer Gruppe die spätere Modellierung des Steuerungsprozesses der Ampelsteuerung als Petrinetz sehr gut gelang.

Auch wenn sich sicherlich noch einzelne Verbesserungspunkte an der Einführungsstunde ergeben haben, so hat sich die grundlegende Konzeption als geeigneter Einstieg in die Modellierung mit Petrinetzen herausgestellt.

Kann die Modellierung eines parallelen Prozesses als Petrinetz Schülerinnen und Schülern helfen, ein besseres Verständnis dieser zu erlangen?

Auch wenn durch die zahlreichen Implementierungsprobleme bei den Gruppen, die Petrinetze im Entwicklungsprozess nutzten, das Bild sehr unscharf geworden ist, so ist zumindest die Eigenheit hervorzuheben, dass die Gruppe, die ohne Petrinetze arbeite, die Schwierigkeiten der parallelen Prozesse weder erkannte noch geeignet lösen konnte. Trotz aller Implementierungsprobleme war den übrigen Gruppen gerade dieser Umstand durch ihre Modellierung der Ampelsteuerung als Petrinetz überaus präsent. Ich wage daher die Aussage, dass Petrinetze hier zumindest ein deutliches Problembewusstsein geschaffen haben. Diese Meinung teilen im Übrigen auch viele Schülerinnen und Schüler aus der dritten Gruppe mit mir, wie man gleich anhand mehrerer Kommentare im Anhang nachlesen kann.

Kann aus einem modellierten Petrinetz unmittelbar ein Grundgerüst für die Implementation abgeleitet werden?

Auch wenn natürlich diverse technische Umstände für Probleme bei der Implementierung sorgten, die man im Nachhinein bei der Reflexion dieser Fragestellung in Betracht ziehen muss, so verleitet mich die Tatsache, dass wirklich sämtliche Gruppen an einer direkten Übertragung vom Petrinetz auf die konkrete Programmierung gescheitert sind, egal ob sie dafür Hilfen erhielten oder nicht, diese Fragestellung (mindestens für diesen Kurs) eher mit Nein zu beantworten, was sich ebenfalls mit Schülerkommentaren im Anhang deckt.

Worin könnten die Probleme liegen, dass die Übertragung so schwer fällt? Zunächst muss man berücksichtigen, dass anders als beispielsweise UML, die unmittelbar an ein programmiersprachliches Konzept angebunden ist, Petrinetze vielmehr an ein theoretisches Konzept angebunden sind. Von daher ist bei Petrinetzen die Schnittstelle zwischen Realität und Modell stark ausgeprägt, aber nicht die zur Programmierung. Dadurch lässt sich ein Petrinetz nur sehr ineffizient auf eine Implementierung übertragen.

Zum anderen sind die verbreiteten Programmiersprachen in den Schulen von ihrem Design her nicht auf parallele Konzepte ausgelegt. Auch wenn sich zum Beispiel in Delphi die Nebenläufigkeit mittels des Timer-Objekts realisieren lässt, hat man es immer noch maßgeblich mit einer imperativen Programmiersprache zu tun, in der Befehle nacheinander abgearbeitet werden. Auch Schubert und

Schwill weisen bereits darauf hin, dass in der Schule typischerweise „*keine Programmiersprache mit parallelen Konzepten zur Verfügung [steht].*“ (Schubert & Schwill, S. 193)

Ob vor dem Hintergrund, dass eine direkte Übertragung von Petrinetz zur fertigen Implementierung kaum möglich scheint und somit „nur“ ein verbessertes Verständnis von parallelen Prozessabläufen durch Petrinetze erreicht werden kann, es lohnend erscheint, diese im Unterricht der gymnasialen Oberstufe gemäß des modellierenden Ansatzes einzuführen, muss schließlich jeder für sich entscheiden. Ich möchte aber noch einmal ausdrücklich hervorheben, dass insbesondere Schülerinnen und Schüler die bislang nur mittelmäßige Programmierqualität in meinem Informatikunterricht zeigten, in Petrinetzen ein motivierendes Thema vorfanden, in dem sie hervorragendes geleistet haben. Zumindest im Zuge der Binnendifferenzierung erachte ich Petrinetze daher in jedem Fall als lohnende Alternative, für Schülerinnen und Schüler, die Abstraktionsfähigkeit in Form einer direkten programmiersprachlichen Implementierung nicht von vorne herein mitbringen.

Literaturverzeichnis

- **Hubwieser**, Peter: *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. Springer, Heidelberg 2007, Seiten 248-249.
ISBN 978-3-540-72477-3
- **Reisig**, Wolfgang: *Petrinetze – Modellierungstechnik, Analysemethoden, Fallstudien*. Webpublikation, 2010, Seiten 3-29.
http://www.informatik.hu-berlin.de/top/pnene_buch/
- **Informatik-Lehrplan** - Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen (Hrsg.): *Richtlinien und Lehrpläne für die Sekundarstufe II – Gymnasium/ Gesamtschule in Nordrhein-Westfalen – Informatik*. Ritterbach Verlag, Frechen 1999. Seiten 45-71.
ISBN 3-89314-612-1
- **Schubert**, Sigrid und **Schwill**, Andreas: *Didaktik der Informatik*. Spektrum Akademischer Verlag, Heidelberg 2004, Seiten 182-194.
ISBN 3-8274-1382-6

Softwareverzeichnis

- **NetLab**, Institut für Regelungstechnik der rheinisch-westfälischen technischen Hochschule Aachen (RWTH), Version 1 (2K) von 2007-06.
<http://www.irt.rwth-aachen.de/downloads/petrinetz-tool-netlab.html>
- **Pipe2**, Imperial College DoC Msc Group and Msc Individual Project, Version 2.5 (RC5) von 2007-12-24.
<http://pipe2.sourceforge.net/>

Anhang

Evaluierungsbogen

Evaluierungsbogen zu Petri-Netzen				
Allgemeine Fragen				
[1] Geschlecht:	() weiblich	() männlich		
[2] Bitte beurteile, wie dich die Themen des letzten Halbjahres, im Nachhinein betrachtet, begeistert haben:				
Kleine Programmierprojekte	++	+	-	--
Programmstrukturen (Rekursion etc.)	++	+	-	--
Datenrepräsentation (Binärzahlen etc.)	++	+	-	--
Aussagenlogik (Und-, Oder-Verknüpf. etc.)	++	+	-	--
[3] Bewerte bitte die folgenden Aussagen:				
→ Die Einführung in Petri-Netze hat mir Spaß gemacht.				
	++	+	-	--
→ Ich würde mir eine ganze Unterrichtsreihe (ca. 3 bis 4 Doppelstunden) zu Petri-Netzen als Teilgebiet der theoretischen Informatik (d.h. ohne direkten Programmierbezug) wünschen.				
	++	+	-	--
Der folgende Abschnitt ist nur von Gruppenmitgliedern auszufüllen, die die Ampelsteuerung ohne Petri-Netze entworfen haben.				
[A] Bewerte bitte die folgenden Aussagen:				
→ Ich habe diese Gruppe gewählt, weil ich mir nicht vorstellen konnte, wie Petri-Netze mir beim Umsetzung der Ampelsteuerung nützlich sein könnten.				
	++	+	-	--
→ Im Nachhinein betrachtet, habe ich das Gefühl, dass mir Petri-Netze bei der Bewältigung der Aufgabe doch hätten dienlich sein können.				
	++	+	-	--

Der folgende Abschnitt ist nur von Gruppenmitgliedern auszufüllen, die die Ampelsteuerung mit Petri-Netzen entworfen haben.

Hat eure Gruppe Hilfen auf dem Weg vom Petri-Netz zur fertigen Delphi-Implementation erhalten? () ja () nein

[B] Bewerte bitte die folgenden Aussagen:

→ *Ich habe diese Gruppe gewählt, weil ich mir vorstellen konnte, dass Petri-Netze mir bei der Umsetzung der Ampelsteuerung helfen könnten.*

++ + - --

→ *Ich fand es einfach, ein passendes Petri-Netz zur Ampelsteuerung zu modellieren.*

++ + - --

→ *Nachdem wir das Petri-Netz modelliert hatten, fand ich es einfach, die Ampelsteuerung in Delphi zu implementieren.*

++ + - --

→ *Im Nachhinein betrachtet, fand ich den Einsatz von Petri-Netzen für die Entwicklung der Ampelsteuerung sinnvoll.*

++ + - --

Der folgende Abschnitt ist nur von Gruppenmitgliedern auszufüllen, die die Ampelsteuerung mit Petri-Netzen mit weiteren Hilfen entworfen haben.

→ *Die Hilfe, die wir erhalten haben, hat mir auf dem Weg vom Petri-Netz zur fertigen Delphi-Implementation der Ampelsteuerung weitergeholfen.*

++ + - --

Platz für weitere Kommentare (ggf. auch Rückseite):

Schülerkommentare im Rahmen des Evaluierungsbogens

Gruppe 1

„Ich finde solche Projekte im Allgemeinen gut, jedoch auf einen kürzeren Zeitraum bezogen.“

„Ich mag Petri-Netze.“

„Mehr Praxis, weniger Theorie. :)“

Gruppe 2

Aus dieser Gruppe lagen keine Kommentare vor.

Gruppe 3

„Programm leicht bedienbar, jedoch nach kurzer Zeit langweilig.“

„Das gesamte Projekt – inklusive Verwendung der Petri-Netze – war mal eine sinnvolle und neue Variante eigenständig zu arbeiten.“

„Die Petri-Netze haben einem ein besseres Verständnis ermöglicht, um das ganze Projekt zu verwirklichen.“

„Ich fand die Petrinetze sinnvoll, um besser zu verstehen, was die Ampelsteuerung machen müsste, fürs Programmieren hat es gar nicht geholfen.“

„Das Erstellen eines Petri-Netzes war sinnvoll, um sich über die Schaltung der Ampel klar zu werden, allerdings war es schwierig, die Funktion des Petri-Netzes für Delphi zu übernehmen.“

„Ich fand die Petri-Netze gut, aber sie haben mir nicht geholfen die Ampelsteuerung zu entwerfen.“

„Petri-Netze haben nicht unbedingt dazu beigesteuert, die Ampelsteuerung in Delphi zu erleichtern. Petri-Netze sind einfach umzusetzen (also geeignet für eine II. Stufe)“

Begleit-CD

Sollten Sie diesen Text lesen können,
so fehlt an dieser Stelle die CD.

Ich versichere, dass ich die Arbeit eigenständig verfasst, keine anderen Quellen und Hilfsmittel als die angegebenen benutzt und die Stellen der Arbeit, die andern Werken dem Wortlaut oder Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe. Das Gleiche gilt auch für beigegebene Zeichnungen, Kartenskizzen und Darstellungen. Anfang und Ende von wörtlichen Textübernahmen habe ich durch An- und Abführungszeichen, sinngemäße Übernahmen durch direkten Verweis auf die Verfasserin oder den Verfasser gekennzeichnet.

Ich bin damit einverstanden, dass diese Hausarbeit nach Abschluss meiner Zweiten Staatsprüfung wissenschaftlich und pädagogisch interessierten Personen oder Institutionen zur Einsichtnahme zur Verfügung gestellt wird und dass zu diesem Zweck Ablichtungen dieser Hausarbeit hergestellt werden, sofern diese keine Korrektur- oder Bewertungsvermerke enthalten.

Duisburg, 2010-05-26