

Abgabe: 25.2.2007

Seminararbeit zum Thema

Konzeption von Aufgaben für ein Unterrichtsprojekt mit dem
Schwerpunkt „Verstehen von Informatiksystemen“

Daniel Graf

Inhalt

Über diese Arbeit	3
Aufgabensammlung zum Thema Zustand	4
Aufgabensammlung zum Thema Beobachter	6
Erläuterungen zu den Aufgaben des Themengebiets Zustand	8
Erläuterungen zu den Aufgaben des Themengebiets Beobachter	10
Musterlösung	
Zustandsmuster	12
Beobachtermuster	16
Schlusswort	19
Quellen	20

Über diese Arbeit

Thema meiner Seminararbeit, die ich im Rahmen meines Hauptstudiums des Lehramts Informatik angefertigt habe, ist die Konzeption und Gestaltung von insgesamt zwölf Aufgaben zu den Themenbereichen Zustand (mit dem gleichnamigen Entwurfsmuster) und Schichten (mit dem Entwurfsmuster Beobachter) jeweils inklusive Musterlösung. In jedem Themenbereich soll eine Aufgabe speziell für den Einsatz in einer schriftlichen Arbeit unter Aufsicht geeignet sein.

Im Folgenden stelle ich daher zunächst die Aufgabensammlungen zu den beiden Themengebieten vor, erkläre danach zu jeder Aufgabe was im Schulunterricht durch ihren Einsatz erreicht werden soll und stelle zum Schluss Musterlösungen zu jeder Aufgabe vor. Ich habe mich für diese Struktur entschieden, um einen Einsatz dieser Arbeit im Schulalltag zu erleichtern. Durch die Zusammenfassung der Aufgaben ohne Lösungsskizzen oder Kommentare kann die Aufgabensammlung leicht auch separat vom Rest der Arbeit betrachtet oder für den Unterrichtseinsatz kopiert werden. Die Bereiche Diskussion der Aufgaben und Musterlösung sind ebenfalls in Blocks zusammengefasst, sodass auch hier eine separate Betrachtung erleichtert wird.

Die Konzeption geschah in Anlehnung an die Konzepte von Peer Stechert und Sigrid Schubert, insbesondere unter Anwendung der Ideen der Strukturierung von Lernprozessen in die Bereiche „Beobachtbares Verhalten“, „Innere Struktur“ und „Implementierungsdetails“ (siehe [Ste]) sowie deren Kombinationen und der vernetzten fundamentalen Ideen in der Informatik.

An dieser Stelle möchte ich mich besonders bei Peer Stechert bedanken, der mich jederzeit kompetent beraten und mit Informationen versorgt hat und der mich während meiner Arbeit sehr engagiert betreut hat.

Die Konzepte die ich mit meinen Aufgaben zu vermitteln suche, sind größten Teils aus [DP] übernommen, beziehungsweise an die dort beschriebenen Ideen angelehnt.

Die Zeiteinschätzung für die Aufgaben kann selbstverständlich nicht jede Schülergruppe beliebig gut abwägen; in der Tat wird es auch innerhalb einzelner Lerngruppen große Unterschiede im Bearbeitungszeitraum geben. Um diese Schwankungen zu minimieren sind einige der Aufgaben mit besonderen Differenzierungsmöglichkeiten ausgestattet, die es ermöglichen sollen, ein breites Spektrum an Schülertypen geeignet zu bedienen. Insgesamt ist eine Vorhersage der benötigten Bearbeitungsdauer jedoch unmöglich, weshalb ich, mit gewissen Kompromissen, die bewährte Formel aus der Schulpraxis verwendet habe: „Schülerbearbeitungszeit = Lehrerbearbeitungszeit * 3“, wobei mit der Lehrerbearbeitungszeit schlichtweg meine eigene Bearbeitungsdauer gemeint ist.

Die Punktevergabe erfolgt sodann auf ähnlich pragmatische Art. Je 5 Minuten Bearbeitungszeit soll ein Punkt vergeben werden. Diese Verteilung setzt die zu investierende Arbeit in Relation zum Punktwert, was immerhin den Anspruch vertreten kann, eine faire Bewertung zu sein. Alternative Bewertungsmaßstäbe die den Schwierigkeitsgrad der Aufgaben in das Schema mit einbeziehen, halte ich für nicht verallgemeinerbar. Man sollte es als Lehrer von der aktuellen Schülergruppe abhängig machen, ob beispielsweise besonders anspruchsvolle Aufgaben mit weniger oder mehr Punkten als Durchschnittsaufgaben bewertet werden sollten. Je nach Motivationslage können Schüler auf bestimmte Punktesysteme unterschiedlich gut reagieren, deshalb sollte der Lehrer die Punktevergabe den Schülern anpassen. So sind erfolgsmotivierte Schüler eher mit höheren Punkten für schwere Aufgaben zu bedienen, während Misserfolgsvermeider gerade durch geringe Verlustaussichten bei schweren Aufgaben motiviert sind, diese anzugehen.

Aufgabensammlung zum Thema Zustand

1) Selbstmodellierung

- a) Definieren Sie den Begriff Zustand!
- b) Modellieren Sie sich informatisch. Achten Sie dabei insbesondere darauf, wie ihr Verhalten sich bei gleicher Eingabe (eine konkrete Handlungsanweisung) aber verschiedenen inneren Zuständen unterscheidet. Mögliche innere Zustände können sein: schlafend, konzentriert, müde... Überlegen Sie, welche Aspekte des Begriffs Zustand hier verwendet werden.

2) Textverarbeitung

Analysieren Sie ein beliebiges Textverarbeitungsprogramm vor dem Hintergrund des Zustandsmusters! Beobachten Sie dabei also insbesondere, wann sich das System bei gleicher Eingabe unterschiedlich verhält.

3) Zustandsbedingtes Verhalten im Alltag

Wo finden Sie in Ihrem eigenen Umfeld ebenfalls zustandsbedingtes Verhalten wieder?

4) Rollenspiel Callcenter

Entwickeln Sie Regeln für ein Rollenspiel, das die Arbeit in einem Callcenter für Produktwerbung simuliert. Bedenken Sie, dass es für die Betreiber solcher Callcenter meist vorrangig ist, die potentiellen Kunden möglichst lange in das Gespräch zu verwickeln. Deshalb haben die Telefonisten einen Codex von Verhaltensregeln, die genau vorschreiben, wie in einer gegebenen Situation des Gesprächs auf welche Kundenaussage reagiert werden soll. Spielen Sie nach diesem Plan einen Anruf mit einem Kunden durch, der Kunde sollte von einem Mitschüler gespielt werden, der Ihren Callcenterplan noch nicht kennt.

5) TCP/IP in UML

Schicken Sie Daten über eine TCP/IP-Verbindung und formulieren Sie das Zustandsverhalten der Verbindung in einem UML-Klassendiagramm. Versuchen Sie, das Zustandsbedingte Verhalten komplett außerhalb des Verbindungsobjektes, also in einer anderen Klasse, zu realisieren.

6) Ampel-Blitzer

In manchen Städten gibt es neben Radarkontrollen, die die Geschwindigkeit von Verkehrsteilnehmern messen, auch „Blitzer“, deren Aufgabe die Überwachung roter Ampeln ist. Sie kennen den Zustand der Signalanlage und prüfen, ob sich ein Fahrzeug in einem vordefinierten Bereich jenseits der Haltelinie der Ampel befindet. Hält sich ein Fahrzeug während der Rotphase im „verbotenen Bereich“ auf, wird es fotografiert. Pro Rot-Phase kann aus Gründen, die hier nicht näher diskutiert werden sollen, nur ein Bild gemacht werden.

Beschreiben Sie das Verhalten der Anlage in einem Zustandsdiagramm.

Aufgabensammlung zum Thema Beobachter

1) Newstickerredakteur

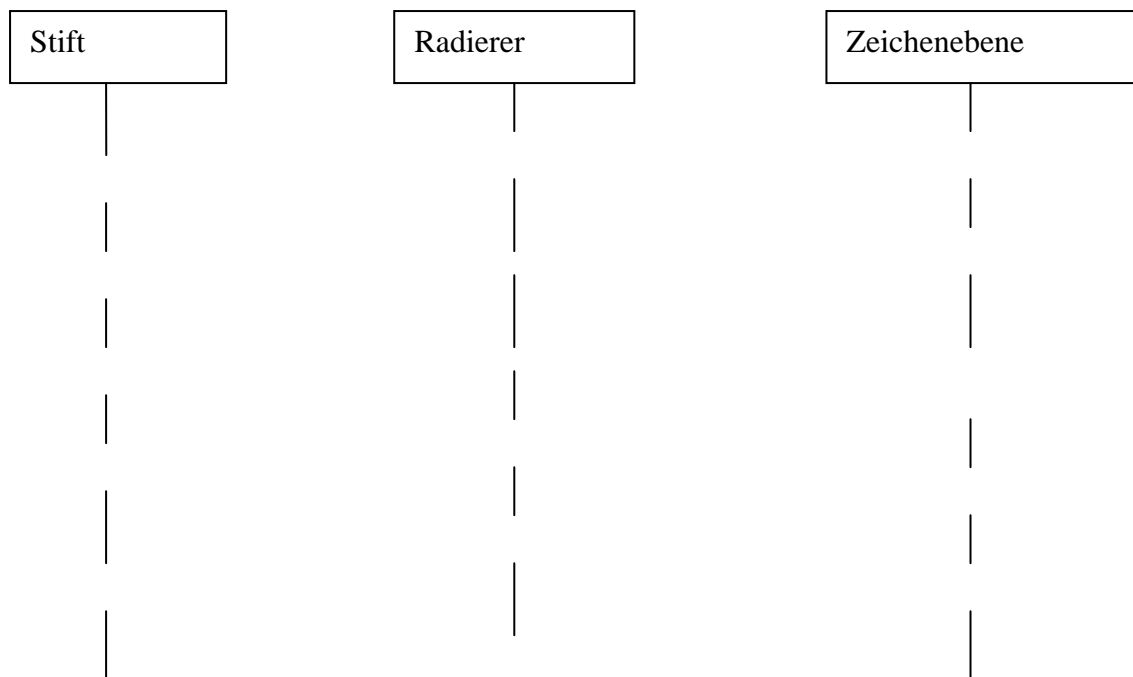
Entwickeln Sie einen Rollenspielplan für die Arbeit in einer Zeitungsredaktion! Modellieren Sie dabei insbesondere die Arbeit eines Redakteurs, der den Newsticker überwacht und eingehende Nachrichten auf ihre Bedeutung für die Redaktion überprüft. Dieser Redakteur soll jede Information des Tickers an die entsprechende Spartenredaktion des Verlages weiterleiten. Des Weiteren soll es möglich sein, dass sich neue Abteilungen beim Tickerüberwacher anmelden bzw. geschlossene Abteilungen dort abmelden.

2) Buttonabhörer

Die Klasse „Ampel_gui“ aus dem jar-file „Ampel_Afg.jar“ soll so verändert werden, dass die Ampel auch die Rot-Gelb Phase anzeigen kann.

3) Zeichenprogramm

Analysieren Sie ein Zeichenprogramm Ihrer Wahl. Suchen Sie nach Objekten auf der Anwenderebene und stellen Sie die Informationsverteilung im angegebenen Sequenzdiagramm dar. Beschränken sie sich auf die Fälle, dass der Stift und der Radierer benutzt werden.



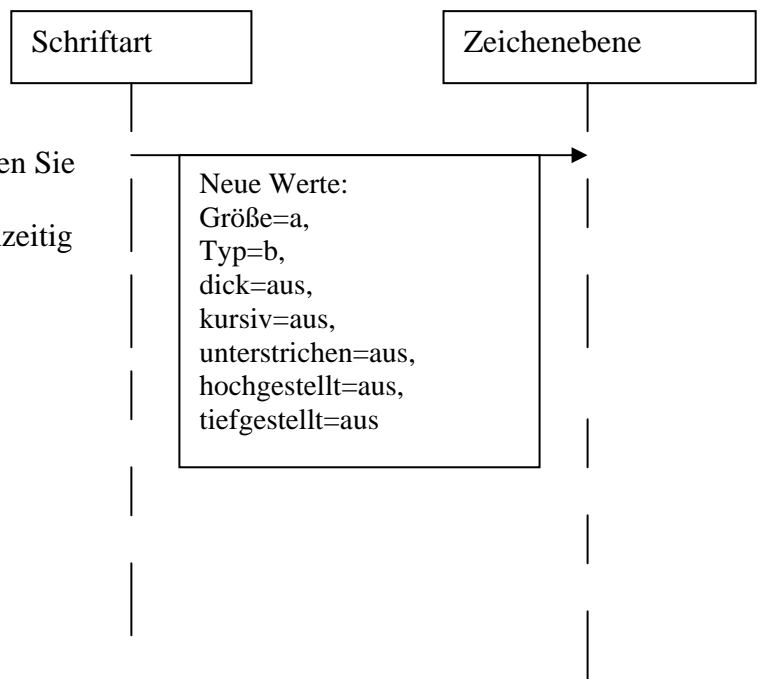
4) Proxy ⇔ Beobachter

Vergleichen Sie das Proxy-Muster mit dem Beobachtermuster; erarbeiten Sie vor allem Gemeinsamkeiten.

5) Push → Pull - Zeichner

Bei einem Textverarbeitungsprogramm kann man im Allgemeinen zwischen mehreren Stilvarianten der Schriftart wählen. Dabei besteht eine Variante aus mehreren Komponenten, wie Schriftgröße, Schriftart, Dicke usw.

Damit die Zeichenebene die eingegeben Zeichen im gewünschten Stil ausgibt, muss bei einem Klick auf ein entsprechendes Symbol ein Informationsupdate stattfinden. Überlegen Sie sich eine alternative Variante zu dieser Situation, bei der nicht alle Werte gleichzeitig übergeben werden müssen:



6) Spionagesatellit

Analysieren Sie die folgende Situation:

Eine aufstrebende Industrienation möchte ihre Macht untermauern. Dazu entwickelt sie einen Plan zur Autonomisierung von Wirtschaft und Politik. Als Teil dieses Plans soll ein Spionagesatellit mit sonnensynchronem Orbit¹ entwickelt werden. Durch verschiedene Effekte, wie die Anziehungskraft des Mondes und die schwankende Strahlungsdichte der Sonne wird die Umlaufbahn ständig gestört werden, sodass so oft wie möglich Korrekturdaten an den Satelliten übermittelt werden sollen.

Da das Zeitfenster zur Datenübertragung bei Satelliten in sonnensynchronem Orbit nur sehr knapp bemessen ist, sollen sämtliche Daten in einem einzigen Paket verschickt werden. Danach können Rückfragen an die Basisstation gestellt werden, wenn Teile der Daten verloren gegangen sind.

Entwickeln Sie nun einen Plan, um die Daten von der Empfangseinheit des Satelliten an die einzelnen Bauteile wie Solarsegel, Triebwerke, Kameras und Speichermedien zu verteilen. Bedenken Sie, dass diese Bauteile nicht genau wissen, wann eine Datenlieferung eintritt, und dass ein Datenpaket sehr groß sein kann.

¹sonnensynchroner Orbit: Der Satellit befindet sich jeden Tag zur gleichen Zeit an einem bestimmten Punkt über der Erde; z.B. überfliegt der Satellit „Landsat“ jeden Tag Mitteleuropa um etwa 10:15.

Erläuterungen zu den Aufgaben des Themengebiets Zustand

Der Duden Informatik ([DI], Seite 728) definiert einen Zustand als eine „Situation, in der sich ein System oder ein Systemteil zu einem Zeitpunkt befinden kann. Hierbei bezieht man sich auf das Innere des Systems und ignoriert die externen Einflüsse...“

Die Ideen, die hier vermittelt werden sollen, sind (siehe [DP]):

- Mithilfe des Zustandsmusters kann man z.B. bei der Implementierung regulärer Ausdrücke mehr Strukturklarheit erreichen.
- Fallunterscheidungen lassen sich mit dem Zustandsmuster leicht realisieren, indem die zustandsbedingte Aktion an das Zustandsobjekt delegiert wird.
- Ein Zustand ist eine Situation in der sich das System befinden kann. Dabei ignoriert man äußere Einflüsse und beschränkt sich allein auf das Innere des Systems.

Diskussion der einzelnen Aufgaben

1) Aufgrund der Einbindung eines theoretischen Begriffs in ein Experiment, beziehungsweise durch dessen experimentelle Anwendung kann diese Aufgabe dem Bereich S_{AB} zugeordnet werden.

a) Hier soll eine alltagsbezogene Definition des Begriffes vorausgeschickt werden, um später genauer analysieren zu können, an welchen Punkten es Überschneidungen gibt, und inwiefern sich das landläufige Verständnis des Begriffes von der informatisch-wissenschaftlichen Sichtweise unterscheidet.

Ich veranschlage für diese Aufgabe **5 Minuten** Bearbeitungszeit, da lediglich ein intuitiv bekannter Begriff in einen anwendbaren Formalismus überführt werden soll. Dementsprechend ist diese Aufgabe **1 Punkt** wert.

b) Ziel dieser Aufgabe ist es, einen intuitiven und damit lebensweltbezogenen Begriff von Zustand zu entdecken. Zusammen mit Teilaufgabe a) soll dann auch die Varianz des Begriffes diskutiert und ein Kern herausgearbeitet werden, der oben erwähnter Duden-Definition inhaltlich nahe kommt.

Die Bearbeitung diese Aufgabe sollte etwa mit **20 Minuten** veranschlagt werden und dementsprechend **4 Punkte** wert sein.

2) Diese Aufgabe lässt die Schüler im Wesentlichen das beobachtbare Verhalten eines Systems analysieren; da lediglich auf die potentielle innere Struktur rückgeschlossen werden kann, ordne ich diese Aufgabe dem Bereich S_A bei, allerdings kann man je nach erzielter Tiefe der Strukturforschung der Schüler diese Aufgabe auch in der Bereich S_{AC} einordnen.

Um die Aufgabe bearbeiten zu können, müssen die Schüler zunächst die Frage klären, was eine Eingabe für ein Textverarbeitungsprogramm sein kann. Dann kann das erkennbare Verhalten des Systems entsprechend verschiedener Eingaben klassifiziert werden, wobei auffällt, dass eine aktuelle Eingabe zu verschiedenen Verhaltensänderungen führen kann.

Diese Aufgabe thematisiert den Zustandsbegriff im Kontext der informatischen Realität anhand des Beispiels eines Textverarbeitungsprogramms, da davon auszugehen ist, dass dies die Lebenswelt der Schülerinnen und Schüler so berührt, dass ein exemplarisches Lernen möglich wird.

Um diese Aufgabe zu bearbeiten müssen die Schüler zunächst klären, was eine Eingabe überhaupt ist, was bis zu 5 Minuten dauern kann. Der Rest der Aufgabe ist mit etwa 20 Minuten zu veranschlagen, was insgesamt eine Bearbeitungszeit von **25 Minuten** ergibt. Dementsprechend sind **5 Punkte** zu verteilen.

3) Auch hier geht es hauptsächlich um beobachtbares Verhalten, daher die Zuordnung zum Gebiet S_A .

Hier sollen weitere Anknüpfungspunkte zur Lebenswelt erstellt werden; die Schülerinnen und Schüler reflektieren über den Begriff des Zustands. Teil dieser eher offenen Aufgabe ist es, nachzuweisen, inwieweit die bereits befundenen Eigenschaften von Zuständen auch auf die neuen Beispiele angewandt werden können, wodurch die Grenzen des Begriffs erfahren werden können.

Da die Aufgabenstellung den erforderlichen Umfang der Bearbeitung nicht vorgibt, kann man hier sehr unterschiedliche Zeitspannen zur Verfügung stellen, ich würde jedoch zunächst von **5 Minuten** ausgehen. Die Bewertung sollte sich ebenfalls auf diesen Grundwert beziehen, also gibt es **1 Punkt**.

4) In dieser Aufgabe ist eine Untersuchung einer inneren Struktur, nämlich der des Callcenters, gefragt. Die Untersuchung geschieht über eine Analyse des nach außen sichtbaren Verhaltens, also gehört diese Aufgabe in den Bereich S_{AB} .

Es soll erkannt werden, dass man bei der Modellierung oft unnötige Details weglassen muss, um ein sinnvolles, weil nicht zu komplexes, Modell zu erhalten.

Da die Aufgabenstellung recht umfangreich ist, sollten 5 Minuten für das Verstehen der Aufgabe veranschlagt werden, die eigentliche Bearbeitung beläuft sich wohl auf etwa **30 Minuten**, also gibt es **6 Punkte**.

5) Die Delegation von Aufgaben auf Zustandsobjekte ist ein wichtiger Aspekt des Zustandsmusters. Dabei handelt es sich um Implementierungsdetails, was für eine Beordnung zu Bereich S_C spricht. Durch das Umsetzen in der Modellierungssprache UML wird der Aufbau eines Systems mit diesem Muster deutlich. Die Umsetzung in einem UML Diagramm ermöglicht, das nach außen hin sichtbare Verhalten zu studieren, weshalb insgesamt Bereich S_{AC} die Kategorie für diese Aufgabe bildet.

Die Schüler können je nach eigenen Fähigkeiten und Interesse das Klassendiagramm untersuchen, wo die Vorteile der Aufgaben Delegation liegen und warum es ohne diese Delegation umständlicher wäre, ein geeignetes System zu entwerfen. Hier gehe ich von **10 Minuten** Bearbeitungszeit aus, also gibt es für diese Aufgabe **2 Punkte**.

6) Auch in dieser Aufgabe geht die Analyse des Sachverhalts von einer Beobachtung des äußeren Verhaltens des Informatiksystems aus. Durch die Darstellung in einem Zustandsdiagramm wird obendrein die Sicht auf das äußere Verhalten gelenkt, allerdings unter starker Berücksichtigung der vermuteten inneren Struktur. Daher sortiere ich diese Aufgabe in Bereich S_{AB} . Die Darstellungsform des Zustandsdiagramms verknüpft mehrere Bereiche der Informatik, wodurch vernetztes Lernen ermöglicht wird. Falls das Thema „Endliche Automaten“ noch nicht behandelt worden ist, können hier bereits erste Erfahrungen gesammelt werden, anderen Falls eignet sich diese Aufgabe auch, um die Thematik noch einmal aufzufrischen, beziehungsweise von einem anderen Standpunkt aus zu betrachten. Die Aufgabe sollte in etwa **15 Minuten** zu bearbeiten sein. Dafür gibt es **3 Punkte**.

Erläuterungen zu den Aufgaben des Themengebiets Beobachter

Die folgenden Ideen haben bei der Konzeption der Aufgaben eine Rolle gespielt (siehe [DP]):

- Das Beobachtermuster wird benutzt, um die Zustandsänderung von Objekten zu verwalten und die Konsistenz des Systems zu gewährleisten
- Beobachter können Informationsupdates für Objekte bewirken
- Beobachter wahren das Geheimnisprinzip: Das Subjekt kennt seine Beobachter nicht genau (es genügt zu wissen über die Interfaceklasse Bescheid zu wissen)
- Der Zeitpunkt des Informationsversands ist wichtig, sonst kann es zu inkonsistenten Zuständen beim Subjekt und seinen Beobachtern kommen
- Der Unterschied von „Push-“ und „Pull-Modell“

Diskussion der einzelnen Aufgaben

1) In dieser Aufgabe sollen die Schüler einen Rollenspielplan entwickeln, der die Thematik „Newsticker“ behandelt. Dazu soll eine Anleitung erarbeitet werden, die die Arbeit eines Redakteurs simuliert. Auch hier wird die innere Struktur eines Systems aufgrund des äußeren Verhaltens nachempfunden. Dabei sollen Aspekte der Struktur herausgearbeitet werden, die für ein bestimmtes Verhalten notwendig sind, demnach passt diese Aufgabe in den Bereich S_{AB} .

Als Erweiterung der Aufgabe kann die Frage geklärt werden, was passiert, wenn eine neue Redaktionssparte im Hause eröffnet wird, oder eine alte geschlossen.

Diese Aufgabe soll zunächst der Motivation für die Beobachter Thematik dienen. Die Schüler können erkennen, dass für ein geregeltes Zusammenwirken von Objekten ein Informationsaustausch notwendig ist, um bezüglich aktueller Ereignisse zu Informieren. Besonders wichtig ist dabei, dass Information nicht wahllos verstreut, sondern ausgewählten Objekten, gezielt zugestellt wird.

Als Anhaltspunkte werden gegeben, dass der Redakteur den Newsticker überwachen soll und bei Eingang einer neuen Nachricht prüft, ob sie für die Redaktion von Interesse sein könnte. Außerdem wird den Schülern der Hinweis gegeben, dass sie überlegen sollen, wie die Nachricht dann verarbeitet wird.

Außerdem soll die Frage geklärt werden, was passiert, wenn eine neue Redaktionssparte im Hause eröffnet wird, oder eine alte geschlossen. Als Bearbeitungszeit werden **25 Minuten** angenommen, also sind **5 Punkte** erreichbar.

2) Die Schüler können auf einem Niveau am Quellcode des Programms arbeiten, das es ermöglicht mit geringem Zeitaufwand die Funktionsweise des Abhörers nachzuvollziehen, indem nach der Anmeldung eines weiteren Knopfes mit entsprechender Konfiguration direkt eine Veränderung des Programmverhaltens sichtbar wird.

Demnach geht es im Wesentlichen um Implementierungsdetails, die zur Realisierung eines gewünschten Verhaltens des Systems notwendig sind, was dem Bereich S_{AC} entspricht. **15 Minuten** Bearbeitungszeit, daher **3 Punkte**.

3) Es wird zunächst die Analyse eines Systems gefordert, von dem man ausschließlich das äußere Verhalten bestimmen kann, da der Quellcode des untersuchten Programms in den meisten Fällen nicht vorliegt. Die Überführung der gesammelten Information in ein UML

Diagramm spricht den Bereich der inneren Struktur von Informatiksystemen an. Insgesamt ordnet sich diese Aufgabe also dem Bereich S_{AB} zu.

Durch die Einbindung in ein aus dem Alltag bekanntes Programm, werden Verknüpfungen zwischen Alltagsdenken und dem theoretischen Modell gefördert. So wird eine Heuristik gefördert, die die Schüler befähigt, das hier behandelte Beobachtermuster, in Anwendungsbeispielen wiederzufinden. Die Bearbeitung sollte ungefähr **10 Minuten** dauern. Für die Aufgabe gibt es **2 Punkte**.

4) Diese Aufgabe erfordert den Vergleich zweier Strukturmodelle. Dieser Vergleich kann je nach Schülerneigung die Bereiche äußeres Verhalten (der mit diesen Mustern konstruierten Systeme), Implementierungsdetails oder innere Struktur umfassen, daher ist auch die Zuordnung zu den Oberkategorien variabel. Natürlich gibt es eine gewisse Präferenz dieser Aufgabe zum Bereich S_B , da die Entwurfsmuster zunächst auf dieser Ebene ansetzen, aber die anderen beiden Bereiche spielen ebenfalls eine wichtige Rolle. Jeder Schüler kann dabei individuell seinen bevorzugten Schwerpunkt setzen. Es sei noch erwähnt, dass diese Differenzierung in den meisten Fällen nicht bewusst erfolgt, was dem individuellen Freiraum aber keinen Abbruch bereitet.

Hier wird explizit auf ein bereits behandeltes Thema zurückgegriffen. Durch den Vergleich der Gemeinsamkeiten der beiden Entwurfsmuster wird ein vernetztes Denken im Informatikunterricht angestoßen und ein vertieftes Verständnis der beiden Themenschwerpunkte ermöglicht.

Eine Bearbeitung dieser Aufgabe setzt die Behandlung der Themen Beobachter und Proxy bis zum einem gewissen Grad voraus, der den Schülern überhaupt die Wissensgrundlage für diese Reflexionsarbeit bietet. Die Aufgabe sollt in etwa **15 Minuten** zu bearbeiten sein. Dafür gibt es **3 Punkte**.

5) In Aufgabe 5 geht es hauptsächlich um Implementierungsdetails, die in einem UML Diagramm dargestellt werden, wodurch das nach außen hin sichtbare Verhalten nachvollziehbar wird. Diese Aufgabe ist durch die Einführung einer Differenzierung innerhalb des Beobachtermusters motiviert. Es ist wichtig ein Thema im Schulunterricht von vielen Seiten zu betrachten, eine Möglichkeit dies für das Beobachtermuster zu tun, ist eine Aufteilung des Musters in die Unterstrukturen Push- und Pull-Modell zu vollziehen.

Für einen Einstieg in diese differenzierte Sichtweise ist nun die genannte Aufgabe konzipiert. Das Ziel dieser Aufgabe ist es, klar zu machen, dass es Sinn machen kann, nicht gleich sämtliche Informationen bei einem Zustandswechsel an alle Beobachter zu versenden, sondern nur eine Nachricht abzuschicken, dass sich der Zustand überhaupt geändert hat. In etwa **10 Minuten** sollte die Aufgabe fertig gestellt werden können, **2 Punkte**.

6) In dieser Aufgabe wird die innere Struktur eines Informatiksystems aufgrund seines gewünschten Verhaltens konstruiert. Der Schwerpunkt liegt dabei auf der Sicht der inneren Struktur; insgesamt lässt sich die Aufgabe in den Bereich S_{AB} einordnen. Bei dieser Aufgabe soll das Pull-Modell eingeübt werden. Am Beispiel des Satelliten wird deutlich, warum es in der Praxis einen echten Bedarf für dieses Modell gibt, es wird ein Lebensweltbezug aufgebaut und ein tatsächliches Problem, nicht nur eine Schulaufgabe beschrieben. Voraussetzung dafür ist allerdings eine gewisse Technikaffinität der Schüler, die man aber bei Teilnehmern eines Informatikkurses in den meisten Fällen annehmen kann. Die Schülerbearbeitungszeit beträgt hier etwa **30 Minuten**. Dafür gibt es **6 Punkte**.

Musterlösung

Zustandsmuster

1) a) Hier wird zunächst eine alltägliche Arbeitsdefinition des Zustandsbegriffs angestrebt. Die Schüler werden womöglich Aspekte wie „Eigenschaftensammlung“ oder „Zeitabhängigkeit“ in ihren Definitionen einbauen. Die „Unterscheidbarkeit von anderen“ wird meiner Erfahrung nach oft vergessen.

Außerdem halte ich ihn für durchaus verzichtbar. Deshalb denke ich, dass folgende Formulierung realistisch ist:

„Ein Zustand ist die Gesamtheit aller zeitabhängigen Eigenschaften eines Objektes.“

Um den Begriff zu präzisieren, kann man zur Besprechung dieser Aufgabe die Definition aus [DI] mit den Schülern (siehe oben) diskutieren.

b) Da die Darstellungsform der Modellierung nicht vorgegeben war, gehe ich davon aus, dass die meisten Schüler eine verbale Umschreibung wählen.

Eine sinnvolle Variante wäre eine tabellarische Darstellung (*Grafik 1*). Ein Zustand ist im Alltag oft mit der Frage: „Wie geht es dir?“, verbunden. Auch in der Informatik möchte man solch eine Frage gelegentlich stellen, um das innere „Befinden“ eines Objektes herauszubekommen. Der informatische Zustand ist jedoch gänzlich objektiv, und wird nicht von äußeren Faktoren beeinflusst. Er determiniert noch viel stärker, als man es im „normalen Leben“ erwartet, das Verhalten des Objektes, so sind Zustand und Eingabe die einzigen relevanten Größen, wenn es darum geht, wie sich ein Objekt verhalten wird.

Zustand	Eingabe		
	lesen	arbeiten	entspannen/erholen
Schlafend	Nicht möglich	Nicht möglich	Gute Erholung
Müde	Langsames Lesen	Sehr langsam	Prüfen, ob Schlaf möglich
Wach	Normales Lesen	Arbeiten normales Tempo	Entspannungsmöglichkeit suchen
Konzentriert	Sinnbegreifendes Lesen	Arbeit geht besonders leicht	Nicht möglich

Grafik 1

Ein Zustand ist im Alltag oft mit der Frage: „Wie geht es dir?“, verbunden. Auch in der Informatik möchte man solch eine Frage gelegentlich stellen, um das innere „Befinden“ eines Objektes herauszubekommen. Der informatische Zustand ist jedoch gänzlich objektiv, und wird nicht von äußeren Faktoren beeinflusst. Er determiniert noch viel stärker, als man es im „normalen Leben“ erwartet, das Verhalten des Objektes, so sind Zustand und Eingabe die einzigen relevanten Größen, wenn es darum geht, wie sich ein Objekt verhalten wird.

2) Die Herangehensweise an diese Aufgabe ist offen gehalten, was den Schülern die Möglichkeit gibt, sich bei der Bearbeitung frei für eine Form zu entscheiden. Denkbar wäre ein Zustandsdiagramm, oder falls die Schüler dieses noch nicht kennengelernt haben eine intuitive Vorstufe dazu. Eine andere Variante ist natürlich die verbale Umschreibung in einem Fließtext, oder in Stichworten.

Beispielhaft könnte folgendermaßen eine Lösung im prosaischen Stil aussehen:

„Ich habe das Textverarbeitungsprogramm Microsoft Word 2003 untersucht. Dabei habe ich festgestellt, dass es viele Objekte gibt, die den Zustand des Arbeitsplatzes beeinflussen. Eine große Gruppe bilden die Icons in der Symbolleiste. Wenn ein Icon aktiviert ist, dann ändert sich der Zustand des Arbeitsplatzes entsprechend der Funktion des Icons. Zum Beispiel kann man den Fettdruck aktivieren, indem man auf das „F“ drückt. Dabei verändert sich dann der innere Zustand des Arbeitsbereichs von Normal- auf Fettdruck, was dazu führt, dass Text, den man, genau wie vorher, über die Tastatur eingibt, nun in breiten Buchstaben auf dem Schreibfeld erscheint.

Eine weitere Möglichkeit den Zustand des Arbeitsbereichs zu verändern, ist die Skalierung zu verändern. Der Unterschied zum Fettdruck ist, dass sich nicht der geschriebene Text verändert, sondern nur seine Darstellung.

Beim drücken auf das Speicher-Icon ändert sich der Zustand von ungespeichert auf abgespeichert. Dabei wird auch der Zustand des internen Speichers verändert, sodass man nun auf der Festplatte (oder eben dem gewählten Speichermedium) auch die erzeugte Datei finden kann.“

Bei der Bearbeitung, werden die Schüler sicherlich hauptsächlich den Zustand des Arbeitsbereichs im Blick haben, obwohl natürlich auch einzelne Komponenten differenzierte Zustände haben können. Zu vorderst sei da die aktuelle Schrift erwähnt, deren Zustand sich auf denkbar variantenreiche Art und Weise verändern lässt. Außerdem gibt es noch weitere Objekte, die in das Layout eingebunden werden können, und die ebenfalls einen eigenen Zustand besitzen, beispielsweise haben alle Objekte eine skalierbare Größe; Tabellen haben Zeilen und Spalten, deren Zellen haben einen Inhalt. Wie tief die Schüler in ihrer Betrachtung vordringen, wird mit Sicherheit großen Schwankungen unterliegen, jedoch bietet die Aufgabe für jedes Bearbeitungsniveau eine sinnvolle Möglichkeit.

3) Voraussetzung für die Bearbeitung dieser Aufgabe ist es, dass bereits eine Definition des Zustandsbegriffs erarbeitet wurde, oder zumindest die Vorstellung dessen, was ein Zustand ist, bei den Schülern verfügbar ist. Die Musterlösung geht dabei von der Kenntnis der eingangs zitierten Definition des Informatik Dudens aus.

Dinge aus dem Umfeld der Schüler, die sich zustandsabhängig verhalten können beispielsweise andere Menschen sein, dazu verweise ich auf Aufgabe 1. Des Weiteren könnten demzufolge auch Tiere, aber auch Fernseher, Computer oder Autos genannt werden. Ich führe nun die Betrachtung für die unbelebten Objekte noch einmal weiter:

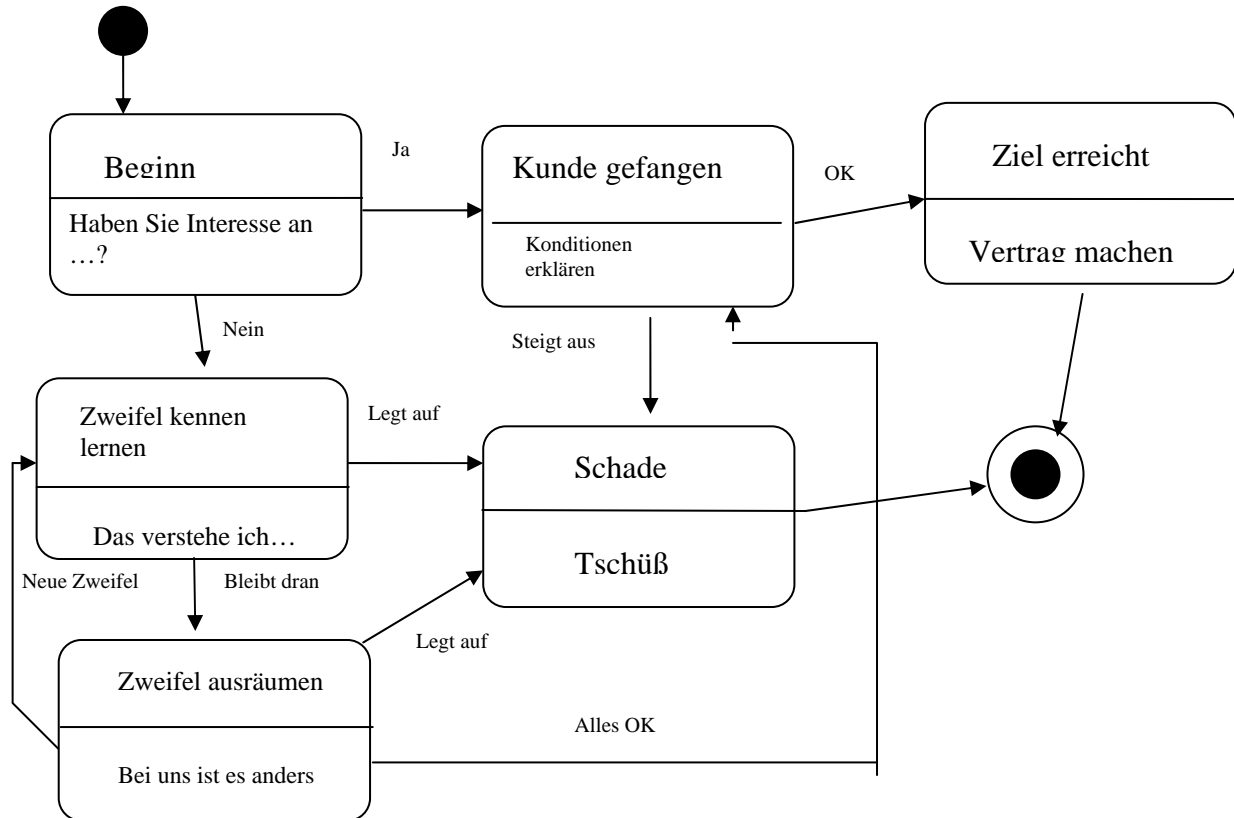
„Die Zustände eines Fernsehers lassen einmal unterteilen in eingeschaltet, Standby und ausgeschaltet. Der Apparat nimmt im ausgeschalteten Zustand nur den Mechanischen Aktivierungsbefehl an, im Standbybetrieb kann man den Befehl über die Fernbedienung geben und im eingeschalteten Zustand verarbeitet der Fernseher die Befehle zur Programmauswahl, Sendereinstellung, Lautstärkewahl und Farbeinstellung so, wie man es allgemein kennt. Bei diesen Befehlen wird abermals der Zustand des Fernsehers verändert. Schaltet man ein Programm weiter, so wird das Frequenzband nun an einer anderen Stelle abgelesen. Die Zustandswechsel und die externe Eingabe über das Antennenkabel bestimmen eindeutig das Verhalten des Fernsehers.

Ein Auto kann man ebenfalls in die Grobzustände Ein Aus und Standby unterteilen, wobei der Standbymodus für die aktivierte Zündung bei inaktivem Motor stehen soll. Die verschiedenen Gänge stellen dann weitere Zustandseinteilungen dar. Das Verhalten des Autos ist dann genau über die Eingabe des Fahrers via Lenkrad, Gaspedal, Bremse und Kupplung, sowie den Zustand des Fahrzeugs bestimmt.“

Das Zustandsbedingte Verhalten eines Computers ist denkbar vielseitig. Ein Ansatz wäre, sich nur das Verhalten unter einem bestimmten Anwendungsprogramm zu beschreiben, hierzu verweise ich auf Aufgabe 2. Eine andere Option wäre, die Grobzustände Ein Aus Standby und

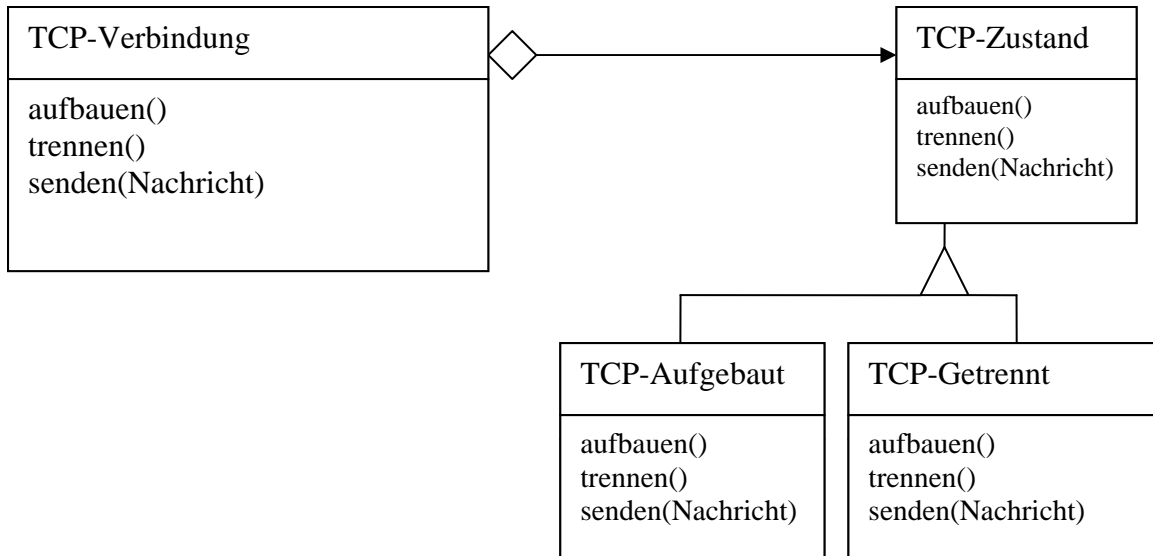
Ruhezustand zu betrachten. Diese Herangehensweise verläuft dann analog zu der für die Modelle Auto und Fernseher beschriebenen.

4) Der Rollenspielplan kann frei formuliert, oder in Form eines Zustandsdiagramms dargestellt werden. Der besseren Übersicht halber verwende ich das Diagramm (*Grafik 2*).

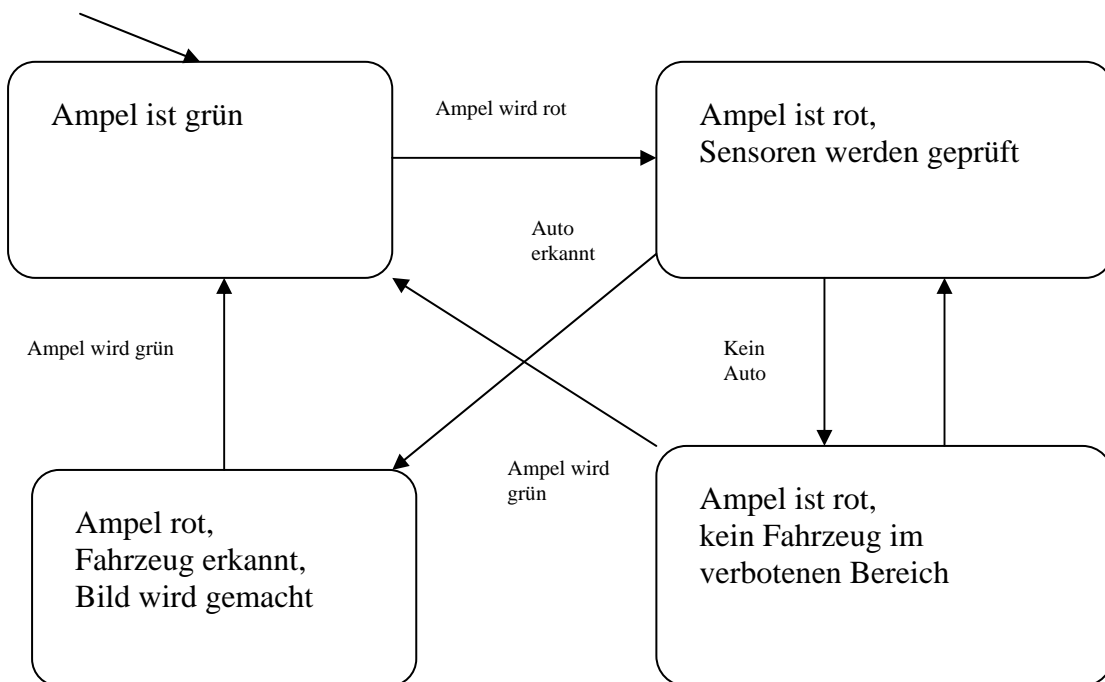


Grafik 2

5) Der Verbindungsaufbau ist nur rudimentär zu denken. Das heißt, es muss lediglich eine Black-Box-Methode zum setzen der Verbindung existieren. Der Zustand der Verbindung kann entweder „Aufgebaut“ oder „Getrennt“ sein. Die Methoden aufbauen und trennen werden vom TCP-Verbindungs Objekt an das TCP Zustandsobjekt weitergeleitet. Die Art der Ausführung des Befehls hängt dann davon ab, von welcher Sub-Klasse das Zustandsobjekt im entsprechenden Moment ist. *Grafik 3* zeigt die Umsetzung in UML.



Grafik 3
6)



Grafik 4

Anstelle des vierten Zustands unten rechts könnte man auch einen Zustandswechsel von Zustand 2 (Ampel ist rot) in sich selbst einbauen. Die Bedingung für diesen Zustandswechsel müsste dann natürlich „Kein Auto“ sein. Der Startzustand ist beliebig zu setzen.

Beobachtermuster

1) Die Bearbeitung dieser Aufgabe lässt in der Wahl der Methodik wieder ausreichend Freiheit, damit auch Schüler, die sich im Umgang mit dem Formalismus etwas schwerer tun, die Aufgabe sinnvoll bearbeiten können. Es ist eben auch gut denkbar, dass man dieses Problem rein verbal löst.

Wenn das Zustandsmuster schon bearbeitet wurde, werden die Schüler sicherlich geneigt sein, es auch hier ansatzweise einzubauen. Dies ist ein erklärtes Ziel der Aufgabe und begründet sich in der Vermittlung der vernetzten fundamentalen Ideen in der Informatik. Die Schüler sollen in der Tat erkennen, dass der Stoff, den sie vermittelt bekommen nicht isoliert dasteht, sondern Anknüpfungspunkte in der weiteren Informatik hat:

„Der Redakteur beobachtet im Grundzustand den Eingang des Newstickers. Sobald er eine neue Nachricht sieht geht er über in den Auswertungszustand.

Im Auswertungszustand liest er den Text, prüft auf der Liste der Redaktionsabteilungen, wer für dieses Thema verantwortlich ist und leitet die Nachricht an die entsprechende Abteilung weiter. Ist die Nachricht nicht von Belang wird sie verworfen.

Wird eine neue Sparte eröffnet, sagt ein Vertreter dem Tickermanager Bescheid, dass nun auch Nachrichten mit der Thematik X an die Redaktionsabteilung Y gesendet werden sollen.

Wird eine Abteilung geschlossen, meldet sie sich beim Tickerbeobachter ab. Der Manager merkt sich diese Informationen in einer Liste.“

Der Rollenspielplan kann in der Klasse exemplarisch durchgeführt werden, dazu sind die Rollen des Tickerbeobachters, des Nachrichtentickers sowie der einzelnen Abteilungen zu verteilen.

Zur Vorbereitung sollte man sich vorab ein paar Nachrichten ausdenken, dies kann sinnvoller Weise im Plenum geschehen. Außerdem müssen ein paar Namen für Redaktionsabteilungen gefunden werden.

Seien beispielsweise {1,2,3,4,5,6} die Schülernamen, dann kann dies der Ablauf sein:

„1 spielt den Newsticker, 2 den Beobachter, 3 spielt die Redaktionsabteilung Sport, 4 die Abteilung Wetter, 5 spielt Finanzen und 6 ist die Abteilung für Immobilien.

Am Anfang ist keine Redaktionsabteilung angemeldet.

1 gibt 2 die Nachricht ‚Heute 15 Uhr; Orkan Stärke 7 in Frankfurt‘. Dieser liest sie durch und verwirft sie, weil er keine Passende Abteilung finden konnte.

Dann bekommt er von 3 und 5 die Nachricht, dass eine neue Abteilung mit dem jeweiligen Namen und der passenden Zuständigkeit erzeugt wurde.

Es kommen ein paar passende und ein paar nicht passende Nachrichten herein, die entsprechend abgearbeitet werden. Dabei muss darauf geachtet werden, dass der Beobachter immer versucht die Liste der zuständigen Abteilungen abzuarbeiten.

Dann meldet sich eine Abteilung ab und die restlichen beiden Abteilungen melden sich an.

Bei den nun folgenden Nachrichten, sollte zumindest für jeden der folgenden Fälle eine passende Botschaft dabei sein: neu angemeldete Abteilung ist zuständig; keine Zuständigkeit zu den angemeldeten Bereichen, Thematik passt auch nicht zu abgemeldeten Bereichen.“

2) Die jar-file „Ampel.jar“ stellt die Musterlösung dar, es handelt sich dabei um eine Abwandlung von „Ampel_Afg.jar“ in der Klasse Ampel_gui.

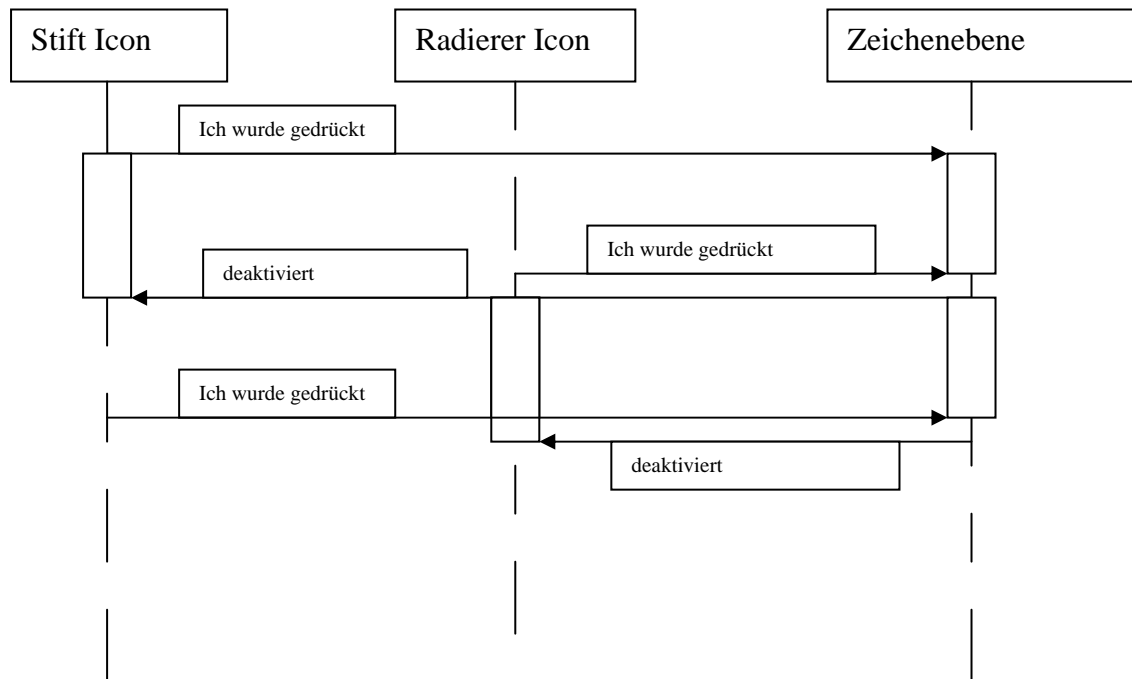
3) Für diese Aufgabe ist es nicht nötig ein besonders aufwändiges Programm zu untersuchen. Die Musterlösung orientiert sich daher am weit verbreiteten Microsoft Programm „Paint“:

„Objekte auf der Anwenderebene sind die verschiedenen Icons. Beispielsweise die Sprühdose oder der Stift. Das Programm muss ständig überprüfen, ob die Icons angeklickt worden sind,

dann verändert sich der Zustand der Zeichenfläche und eine mit der Maus gezogene Linie wird entsprechend als Radieren oder Sprühen usw. interpretiert und dargestellt.“

Grafik 5 stellt eine Musterlösung im Sequenzdiagramm dar.

Natürlich kann man nicht genau sagen, wie das Programm intern realisiert ist, aber diese Lösung ist eine plausible Annäherung. Ebenso kann es natürlich auch sein, dass sich Stift und Radierer gegenseitig beobachten und die Deaktivierungsnachricht nicht von der Zeichenfläche gesendet werden muss.



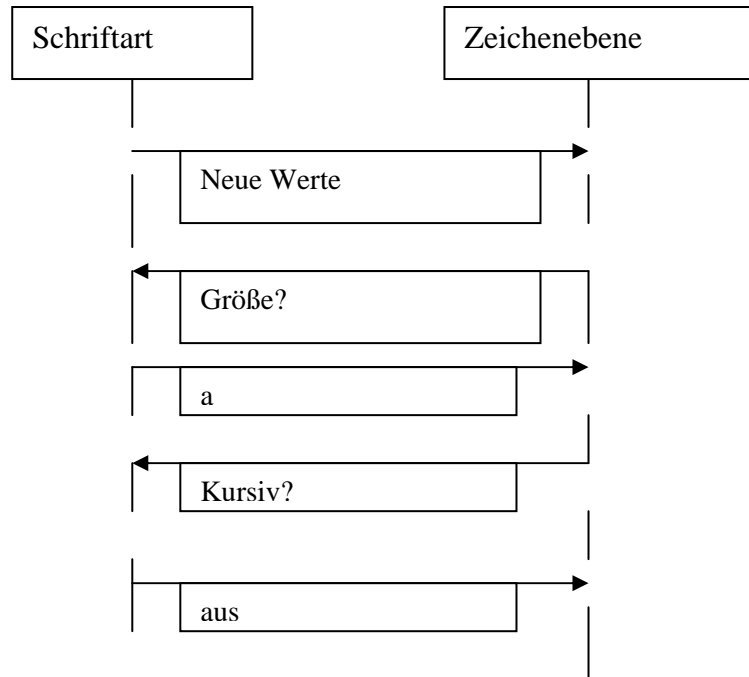
Grafik 5

4) Auch diese Aufgabe ist durch die Vermittlung des Vernetzungsgedankens in der Informatik motiviert.

Wesentliche Aspekte, die herausgefunden werden sollen sind, dass es sich natürlich in beiden Fällen um Entwurfsmuster handelt. Beide regeln die Informationsverteilung und wahren das Geheimnisprinzip. Besonders der zweite Punkt ist wichtig, deshalb sollte die Idee des Geheimnisprinzips bereits bekannt sein, damit sichergestellt ist, dass die Schüler ihn auch wiederfinden. Ist das Prinzip noch nicht eingeführt worden, so lässt es sich am Vergleich der beiden Muster gut verdeutlichen. Allerdings benötigen die Schüler dann noch eine Hilfestellung, wie zum Beispiel eine Diskussion der Frage: „Was muss ein Objekt über einen Proxy wissen, um damit arbeiten zu können und welche Information braucht der Kontext von einem angemeldeten Abhörer, um ihm Botschaften schicken zu können?“

5) Auf den ersten Blick scheint diese Variante umständlicher zu sein, aber wenn man dem Beobachter freistellt, wann er die genauen Veränderungen mitgeteilt bekommt, lassen sich Prozesse zum Teil angenehmer organisieren. Zur weiteren Vertiefung gibt es dann Aufgabe 6.

Grafik 6 zeigt eine mögliche Lösung.



Grafik 6

6) Falls Aufgabe 5 aus diesem Themenblock bereits von den Schülern bearbeitet wurde, oder falls das Pull-Modell mit den Schülern bereits auf andere Weise thematisiert wurde, kann man erwarten, dass die meisten Schüler einen Ansatz wählen, der im wesentlichen das Pull-Modell realisiert. Beispielsweise kann ein Plan folgendermaßen aussehen:

„Die zu übermittelnden Daten werden an der Basisstation in einen Sendestrom geschrieben, der über Funk an den Satelliten gesendet wird. Die Empfangseinheit des Satelliten aktiviert in einem vordefinierten Zeitfenster den Eingabestrom und schreibt, sobald Daten eintreffen, diese in den Strom, der wiederum in einen Arbeitsspeicher geschrieben wird.

Die Empfangseinheit gibt dann allen Peripheriegeräten des Satelliten Bescheid, dass neue Daten eingetroffen sind. Sie sendet nur diese kurze Nachricht, um die einzelnen Bauteile anzustoßen, dass sie sich die nötigen Daten holen können. Jedes Bauteil stellt dann eine Anfrage an den Speicher der Empfangseinheit, ob relevante Daten angekommen sind. Zum Beispiel fragt die rechte Steuerdüse nach, ob der Kurs korrigiert werden soll. Diese Daten werden dann gezielt zurückgegeben.

Das hat den Vorteil, dass Einheiten, die zum Zeitpunkt des Dateneingangs beschäftigt sind nicht unnötig von der Datentransferkapazität der Empfangseinheit gebrauch machen müssen, sondern erst ‚on demand‘ die jeweilige Information übermittelt wird. So kann der gesamte Datenversand innerhalb des Satelliten schneller ablaufen, wodurch Rückfragen an die Basisstation komfortabler innerhalb des Zeitfensters gestellt werden können.“

Schlusswort

In Ermangelung einer Inspiration für eine ergreifende Schlussrede, möchte ich ein paar Worte über die Probleme und Schwierigkeiten verlieren, die während des Ideenfindungsprozesses für die Aufgaben aufgetreten sind.

Wie man sich leicht denken kann, mangelte es mir nicht allgemein an Ideen zu den vorgegebenen Themenbereichen, denn immerhin hatte ich den größten Teil meines Informatikstudiums bereits abgeschlossen und die mir zur Verfügung gestellte Literatur, die ich an dieser Stelle noch einmal sehr empfehlen möchte, waren mir eine Quelle großzügiger Inspiration. Wie sich jedoch herausstellte, waren die Anforderungen, die meine Aufgaben erfüllen sollten, nicht so leicht einzuhalten, wie ich zunächst angenommen hatte. Im Kern der Entwicklung stand der Gedanke, dass in jeder Aufgabe auf eine gewisse Weise das Ausgabeverhalten eines Informatiksystems zu untersuchen sei, damit Schüler, die diese Aufgabe bearbeiten, ihre Erfahrungen auf einer sehr direkten und nachvollziehbaren Ebene machen können.

Üblicherweise hatte ich bis dato hauptsächlich Aufgaben des Typs S_C oder S_B kennengelernt. Sowohl während der eigenen Schulzeit, als auch (insbesondere) an der Hochschule. Es ging dort oft im Wesentlichen darum, zu programmieren, um irgendein in der Ferne liegendes Ziel zu erreichen; für den Schulunterricht soll aber, wie Stechert fordert, der Arbeitsaufwand der eine sichtbare Verhaltensänderung des Systems ändert, überschaubar bleiben. Dies erforderte von mir eine bis dahin unbekannte Einstellung zum Thema Aufgaben im Informatikunterricht. Zwar hatte ich bereits während meines Tagespraktikums in Informatik solche Aufgaben kennengelernt, aber die Konstruktion interessanter Probleme für meine Themenbereiche zu finden wurde dadurch nicht wesentlich leichter.

Einige meiner ersten Versuche sahen dann so aus, dass ich schließlich ein Muster gefunden hatte, das den Anforderungen entsprach und dieses sehr weit ausreizte. Ich schrieb zunächst einige Aufgaben, bei denen es jeweils darum ging, ein bestimmtes Programm zu untersuchen. Zu den Beispielen Zeichen- und Textverarbeitungsprogramm gesellten sich bald Varianten für Internetbrowser, Rechner eine fiktive Ferienanlagenverwaltung, wie sie gelegentlich im Programmierpraktikum an der Uni Siegen implementiert wird. Jedoch entschied ich, dass etwas mehr Abwechslung sein müsse, wenn die Aufgaben den Anspruch haben sollen, dass man auch mehrere von ihnen zum gleichen Themengebiet denselben Schülern vorsetzen können soll; wengleich auch die vorliegende Arbeit nicht als Konzeption der Unterrichtsreihen zu den Themen Zustand oder Beobachter zu verstehen sind.

Ich überlegte, wo mir im Alltag Situationen begegnen, die mit meinen Themengebieten zutun haben. Ich ging bei dieser Vorgehensweise davon aus, dass die gefundenen Probleme besonders lebensnah sein würden und damit echte Probleme und nicht bloß Aufgaben darstellen. Ich notierte mir einige Ideen und nach Rücksprache mit meinem Betreuer überarbeitete ich noch mal einige davon, beziehungsweise ersetzte ich einen Teil der Aufgabe durch neue. Das Hauptproblem war zuvor abermals die Einhaltung des S_A Kriteriums gewesen, die Aufgaben hatten sich oftmals zu sehr an Implementierungsdetails oder Strukturfragen orientiert, ohne erkennbares Verhalten des jeweiligen Systems zu berücksichtigen. Eine Möglichkeit Struktur bzw. Implementierungsfragen mit Beobachtungsexperimenten zu verknüpfen war nun der vermehrte Einsatz von UML. Es erwies sich als praktikabel, Situationsanalysen in Diagrammform zu protokollieren, da hiermit auf schülernahe Weise auch ohne Kenntnis eines komplexen Programmcodes die Idee hinter dem jeweiligen System erforscht werden kann.

Insgesamt empfand ich die Arbeit an diesem kleinen Projekt recht anstrengend, da neue Denkweisen von mir erwartet wurden, die meiner Erfahrung zum Teil diametral entgegenstanden. Doch gerade das ist es, was mir für meine künftige Berufspraxis besonders wertvoll erscheint, denn insbesondere im Schulfach Informatik ist die Konstruktion von Aufgaben aufgrund sehr dünner Modellaufgabendichte ein besonders zentraler Bestandteil des Lehrerberufs.

Quellen

- [Sde] Das Ampel-Programm ist teilweise angelehnt und zitiert von „Schule.de“:
OSZ-Handel und I BICS, Johann Penon, 2000 Berlin
- [DP] SoftwareErich Gamma, Richard Helm, und Ralph E. Johnson, John Vlissides: Design
Patterns. Elements of Reusable Object-Oriented. Addison-Wesley Longman, 1995
Amsterdam
- [Ste] Stechert, P.; Schubert, S.: A Strategy to Structure the Learning Process Towards
Understanding of Informatics Systems. Working / Joint IFIP-Conference Informatics,
Mathematics and ICT (IMICT2007): A golden triangle. 2007 Boston
- [DI] Bibliographisches Institut, Volker Claus und Andreas Schwill: Duden Informatik,
2003 Mannheim