Prüfungsprotokoll praktische Informatik

Datum:	12.10.2005
Prüfling:	Mathias Lüstraeten
Prüfer:	Prof. Lichter und Prof. Kowalewski
Fächer:	Objektorientierte Softwarekonstruktion, Software-Qualitätssicherung, Einführung in eingebettete Systeme, formale Methoden für eingebettete Systeme und embedded software design
Note:	1.0

OOSK:
In einem Meta-Modell für die objektorientierte Programmierung, was wären da so die Modellierungselemente?
Man hat Klassen, Objekte, Methoden und Attribute. Da zwischen gibt es natürlich auch noch Beziehungen. Zwischen Klassen kann es Vererbungsbeziehungen geben. Zwischen Objekten kann es benutzt-Beziehungen geben, da ein Objekt Nachrichten an ein anderes Objekt schicken kann, die dann zu einem Methodenaufruf führen. Ein Objekt ist ein Exemplar einer Klasse.

Was heißt Vererbung?
Eine Oberklasse erbt alle Eigenschaften, also alle Methoden und Attribute von der Oberklasse. Sie kann die Oberklasse definieren, redefinieren und erweitern (wollte er nicht wirklich hören)

Vererbung ist ja aus dieser Sicht eigentlich nur ein programmiertechnischer Kniff. Was will man denn mit einer Vererbungsbeziehung sagen?
Klassen modellieren ja Begriffe…

Klassen sind die softwaretechnische Umsetzung für Begriffe..
Ja… Also man sollte Vererbung dann einsetzen, wenn aus fachlicher Sicht der Begriff, der durch die Unterklasse repräsentiert wird, eine Spezialisierung des Begriffs der Oberklasse ist.

Wie kann man denn Objekte erzeugen?
Je nach Programmiersprache kann es zum Beispiel eine Klassenoperation geben, die ein Objekt der jeweiligen Klasse erzeugt. Gut – also gibt es in dem Meta-Modell auch noch Klassenmethoden.

Sie sagten Klassen-Methoden. Wie ist denn dann die Beziehung zwischen Klasse und Objekt?
In dem Fall kann man die Klasse als Objekt ansehen – allerdings gibt es dann ja nur ein Objekt pro Klasse.

Nee, nee. Ich kann durchaus mehrere Instanzen haben…
Dann wird die Beziehung zwischen Objekt und Klasse im Metamodell etwas aufgeweicht…

Nein – kann man so nicht sagen. In dem Fall sind Klassen Instanzen von Klassen! Aber zurück zum Meta-Modell? Gibt es da nicht noch mehr?
Achso, ja es gibt da noch abstrakte Klassen.

Was sind abstrakte Klassen? Warum heißen die abstrakt?

Eine Klasse heißt abstrakt, wenn mindestens eine Methode abstrakt ist, dass heißt noch nicht ausimplementiert. Das macht man deshalb, um für die Unterklassen eine gemeinsame Schnittstelle zur Verfügung zu stellen. Man kann auch abstrakte Klassen nur mit Hilfe der Vererbung benutzen, in dem man alle abstrakten Methoden ausimplementiert, also definiert.

Warum packt man nicht die vollständige Implementierung in die Oberklasse?
Man will ja generisch bleiben. Wenn ich da so an eine Template-Methode denke, dann kann sie allgemein formuliert werden, und nur noch die Hooks müssen neu implementiert werden.

Ja na gut, aber warum heißen die abstrakt?
(Keine Ahnung)

Weil auf dem Niveau gar nicht die Möglichkeit besteht, die Methoden genau zu beschreiben, da die gewünschte Funktionalität noch gar nicht bekannt ist! Wie sieht das Klassifikationsschema von Meyer aus?
Man hat 3 große Bereiche: Model Inheritance, Variation Inheritance und Software Inheritance. Model Inheritance ist die Spezialisierung eines Begriffs. Außerdem gibt es da noch so was mit Einschränkungen der Oberklasse – wir hatten da das Beispiel Rechteck – Quadrat. Variation Inheritance beschäftigt sich mit der Redefinition von Eigenschaften der Oberklasse – also vor allen Dingen Redefinition von Methoden. Software Inheritance bezeichnet das Erben von Implementierung. Also Erben von Methoden oder Attributen der Oberklasse.

In dem Schema gibt es ja teilweise merkwürdige Arten der Vererbung. Wissen wie, was mit View Inheritance gemeint war?
Also vom Begriff her würde ich daraus schließen, dass man eine abstrakte Sicht auf ein gewisses Modell hat…

Also sie wissen nicht richtig was damit gemeint ist?
Nein.

Macht nichts. Es geht darum, dass man verschiedene Aspekte einer Klasse getrennt voneinander betrachtet. Man hat Teile der Aspekte und erbt sich diese dann zusammen.
OK. Gehen wir mal davon aus, Sie haben die perfekte Anforderungsspezifikation, also vollständig, widerspruchsfrei und so weiter. Wie können Sie daraus einen Entwurf machen?
Ich muss ein Begriffsnetz bilden, damit ich die zentralen Komponenten, also zum Beispiel Klassen des Systems identifizieren kann. Dann muss ich noch die Beziehungen zwischen den Komponenten festlegen – und wenn ich noch den Feinentwurf dazu nehme muss ich auch noch die Schnittstellen der Komponenten kennen.

Naja die sollte man eigentlich schon im Grobentwurf kennen. Reicht es denn, nur die Statik zu betrachten?
Wenn Sie so fragen, anscheinend nicht. Aber meiner Meinung nach spiegelt der Entwurf die statische Sicht des Systems wieder. Die dynamische Sicht kommt ja erst in den Methoden und so weiter zum Vorschein…

Man sollte sich schon beim Entwurf Gedanken über die gewünschte Interaktion machen, sonst kriegt man das System nicht richtig modelliert! Wenn Sie die nicht im Hinterkopf haben, wird das nicht hinhauen, die statische Sicht richtig hinzukriegen. Wie kann man denn einen besonders guten Entwurf hinbekommen?

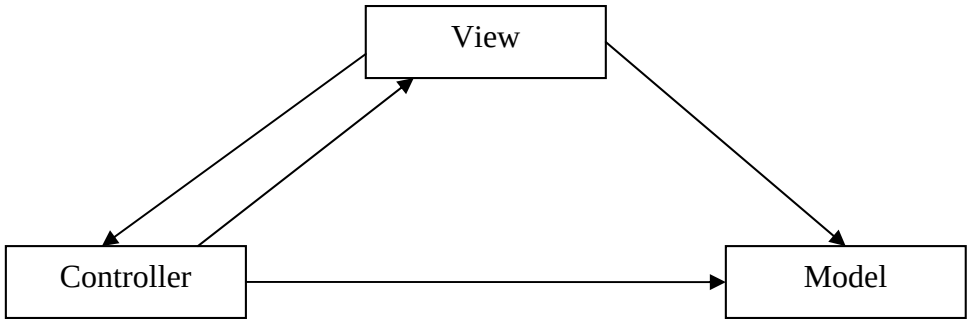
Man kann bewährte Verfahren zur Modellierung einsetzen, Entwurfsmuster benutzen und die Entwurfsprinzipien beachten. Es gibt da Open-Closed, Information Hiding, Abstraktion, Teile und Herrsche, hohe Kohäsion/geringe Kopplung… und vielleicht noch ausreichend/vollständig/einfach

Dieses ausreichend/vollständig/einfach ist ja nur was, mit dem man das nachher bewerten kann. Es gibt da auch noch Separation of Concerns. Was ist das?
Eine Trennung der Zuständigkeiten. Zum Beispiel bei einem Model und einer View. Wenn man da die Zuständigkeiten trennt, kann man auch mehrere Views zu einem Model hinbekommen. Aber vor allen Dingen verbessert man die Wiederverwendbarkeit. Man muss ja oft eine neue GUI entwickeln. Wenn man Model und View getrennt hat, dann kann man das Model ohne irgendwas zu verändern einfach wieder verwenden.

Mit Model und View, da sind sie ja schon beim Observer-Pattern…
Ja, beim Observer-Pattern hat man den Vorteil, dass das Model die Views vorab, also bei der Codierung nicht kennen muss. Zur Laufzeit natürlich schon. Die Views melden sich beim Model an und werden dann benachrichtigt wenn sich am Model was ändert…

OK OK, wollen wir da mal nicht zu sehr ins Detail gehen. Wenn sie schon Model und View haben…
Es gibt da das MVC-Modell!

Haben wir was zum Schreiben hier? Malen Sie mal auf!



Man hat hier eigentlich nur ein erweitertes Observer-Pattern. Die View ist Observer des Models. Zusätzlich hat man noch einen Controller, der auf GUI-Ereignisse reagiert, und dann entsprechende Methoden bei dem Model aufruft.

Wenn Sie jetzt eine Grenze ziehen müssten zwischen den Daten und der Darstellung, wo wäre die dann?
(Model durch eine Linie abgetrennt)

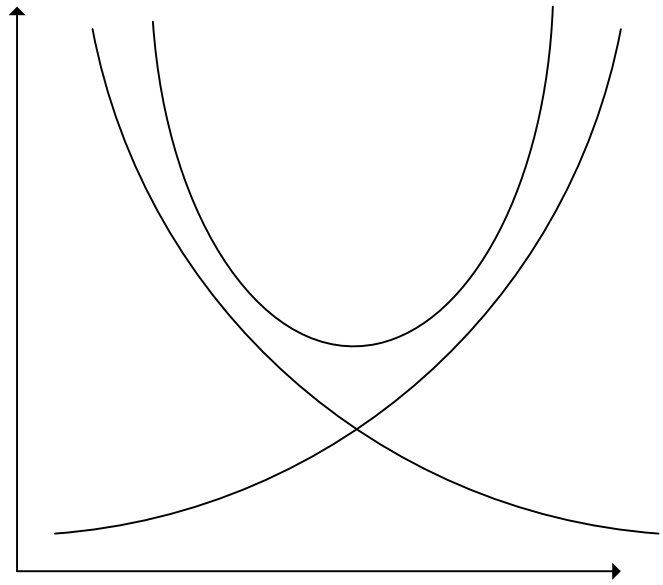
Genau! Wir gehen jetzt mal zum Test. Also jetzt kommt ein kleiner Shift…

SQS:
Was muss für gute Tests gelten?
Ein Test muss redundanzarm, fehlersensitiv, ökonomisch und repräsentativ sein. Außerdem muss man Tests systematisch auswählen, am besten anhand eines Fehlerauswahlkriteriums.

Ja schon richtig, aber wann ist ein Test gut?

Wenn er Fehler findet. Wenn ein Test keine Fehler findet, heißt das nicht, dass das Programm keine Fehler hat. Also hat und der Testfall keinen Nutzen gebracht. Wenn er aber einen Fehler findet, kann man diesen gezielt beheben und man hat einen Fehler weniger.

Wann hört man mit dem Testen auf?
Es gibt da mehrere Testende-Kriterien. Bei einem von denen malt man einen Graph der so aussieht:



Das eine sind die Fehlerfolgekosten und das andere die Kosten für die Prüfung und die Beseitigung des Fehlers. Dann versucht man, das Minimum von beiden anzustreben.

Wie sind jetzt genau die Achsenbeschriftungen?
Auf der y-Achse sind die Kosten, auf der x-Achse der Projektfortschritt? (Ich habe noch rumgeraten, mir fiel es aber nicht mehr ein: Auf der x-Achse ist der Qualitätsgrad)

Im Projektmanagement sind ja Kosten besonders wichtig. Wie müsste man deswegen die die gefundenen Fehler bewerten?
Die reine Anzahl der Fehler ist nicht entscheidend, da unterschiedliche Fehler unterschiedliche Fehlerfolgekosten verursachen können. Also wichtet man die gefundenen Fehler mit den Fehlerfolgekosten.

Außer Tests gibt es ja noch mehr analytische Möglichkeiten für die Qualitätssicherung…
Es gibt noch statische Prüfung, beispielsweise durch Reviews und Messen.

Beim Messen gibt es ja LCOM. Wissen Sie, was das ist?
Man zählt die Anzahl der Methodenpaare einer Klasse mit gemeinsamen Instanzvariablen und zieht davon die Anzahl der Methodenpaare ohne gemeinsame Instanzvariablen ab. Oder andersherum? Betraglich wäre das auf jeden Fall gleich.

In den Ingenieurwissenschaften wird ja unheimlich viel gemessen. In der Informatik jedoch kaum. Was ist das Problem mit den Metriken in der Softwaretechnik?
Der Unterschied ist, dass es bei der Softwareentwicklung um Entwicklung geht und nicht um Produktion. Deshalb ist es so schwierig, gute Metriken zu definieren, die für beliebige Software gilt. Außerdem gibt es das Problem, dass einfach zu ermittelnde Metriken kaum

Aussagekraft haben wie beispielsweise LOC und aussagekräftige Metriken schwer zu ermitteln sind.

Ja das stimmt schon. Wie können Sie denn überprüfen, ob jetzt die Kollision tatsächlich hoch ist, wenn Sie einen niedrigen Wert für LCOM gemessen haben?
Also prinzipiell würde ich sagen das geht nicht.

Oh, das geht schon…
Ich habe ja die Kohäsion definiert als Kopplung über die Instanzvariablen, also muss ja immer dasselbe rauskommen…

Aber man kann ja das leicht so umschreiben, dass LCOM größer ist aber die Kohäsion trotzdem die Gleiche ist. Nein, man kann das einem Experten vorlegen, der dann aufgrund seiner Expertise sagen kann, ob die Kohäsion groß ist oder nicht. Sie nannten ja gerade schon Reviews. Die gelten ja als besonders gut. Warum ist das so?
Wir finden Ursachen und nicht Fehler. Außerdem finden wir meist dann schon die Fehler viel eher im Prozess und das ist ja wegen dem Fehlerverstärkungsmodell besonders gut.

Warum testet dann jeder? Das ist doch eine Standard-QS-Maßnahme. Und nur so wenige Leute veranstalten Reviews?
Das Bewusstsein und das Know-How bezüglich Reviews ist noch nicht so ausgeprägt. Außerdem ist ein Test das Erste was man macht, wenn einem nichts Besseres einfällt – weil es einfach intuitiv ist. Außerdem haben Reviews einen hohen Management-Aufwand: Ich muss Leute freistellen, Räume organisieren und so weiter. Deswegen ist ein Review teuer als ein einfacher Unit-Test zum Beispiel.

Kowalewski (alles durcheinander):
Wir haben zwei Bereiche gehabt, kontinuierliche Regelung und diskrete Steuerung. Worum geht es dabei der diskreten Steuerung?
Wir haben eine Strecke, die kontinuierlich sein kann, aber durch diskrete Aktuatoren und Sensoren beeinflusst wird. Dadurch kriegt man eine diskrete Steuerung der Anlage.

Was heißt in dem Zusammenhang Steuerung?
Das System gezielt so beeinflussen, dass es so reagiert, wie wir es wollen.

Was macht mir bei der zeitkontinuierlichen Regelung?
Man betrachtet Systeme. Das sind Blöcke, die alle möglichen Eingabeverhalten auf ein Ausgabeverhalten abbilden. Grundaufgaben sind hierbei die Sollwertverfolgung und die Störungsbeseitigung. Achso, die Störgrößen fließen auch noch in das System ein.

Was ist ein eingebettetes System?
Ein eingebettetes System ist ein System, was in ein technisches System eingebettet ist. Man hat keine bis eine sehr geringe Schnittstelle zum Benutzer und zur Umgebung. Das System ist aber für die korrekte Funktion des technischen Systems zuständig. Man kann unterscheiden zwischen Produktionsautomatisierung und Produktautomatisierung…

Was ist ein dynamisches System?
Ein dynamisches System ist ein System mit Gedächtnis. Man kann hier nicht einfach mit Hilfe des aktuellen Eingangswertes und einer Funktionsgleichung einen Ausgangswert berechnen, sondern muss auch noch das Verhalten des Systems in der Vergangenheit mit betrachten. Das kann man zum Beispiel mit einem Zustand machen.

Wie kann man dynamische Systeme modellieren?
Mit y(t)=F{u(t)}, F ist dabei ein Operator. Das kann man dann meistens auflösen zu einer Differentialgleichung.

Wie kann dieses „vorherige Verhalten“ modellieren?
Mit der Zustandsdarstellung des dynamischen Systems:

$$\dot{x}(t) = \frac{dx(t)}{dt} = f(x(t), u(t))$$

$$y(t) = g(x(t), u(t))$$

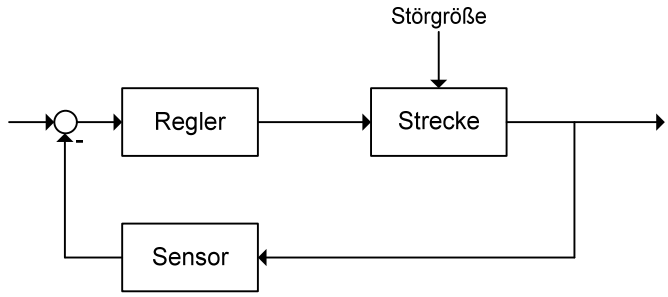
(ein bisschen die Formel erklärt)

Gibt es so eine Darstellung auch für dynamische Systeme?
Ja man hat dann einfach einen Mealy-Automaten:

$$x(k+1) = f(x(k), u(k))$$

$$y(k) = g(x(k), u(k))$$

Ja, man hat also quasi dieselbe Darstellung wie im kontinuierlichen Fall. In manchen Fällen reicht auch ein Moore-Automat. Wir hatten ein allgemeines Bild für eine solche kontinuierliche Regelung, zeichnen Sie diese bitte auf.



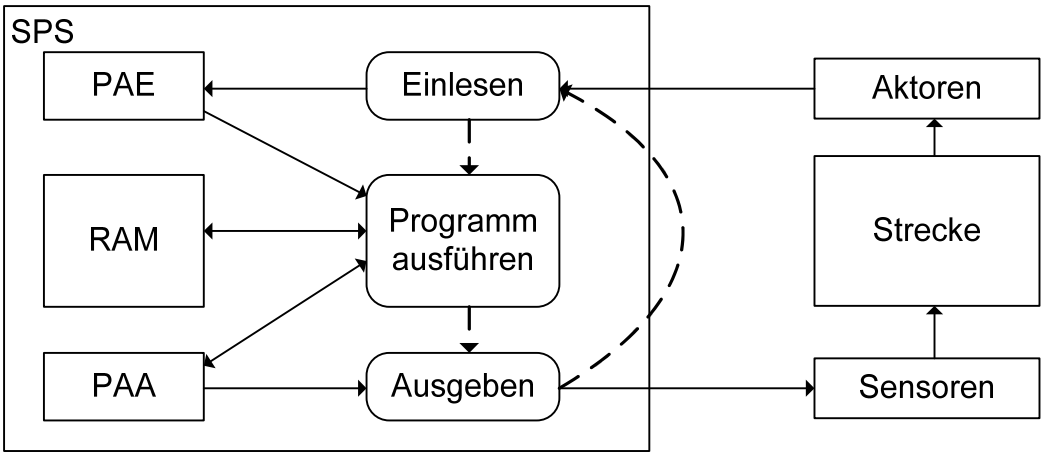
(Ich hatte ein bisschen Probleme, die einzelnen Ein- und Ausgänge richtig zu benennen)

Nehmen wir jetzt im Auto dieses ACC - apdaptive cruise control- Beispiel. Was wären da die Variablen die in den einzelnen Elementen ein- und ausfließen?
Also wir haben die Stellgröße, das wäre zum Beispiel der gewünschte Abstand, der Regler regelt dann die gewünschte Geschwindigkeit daraus, die dann zur Motorsteuerung geht…

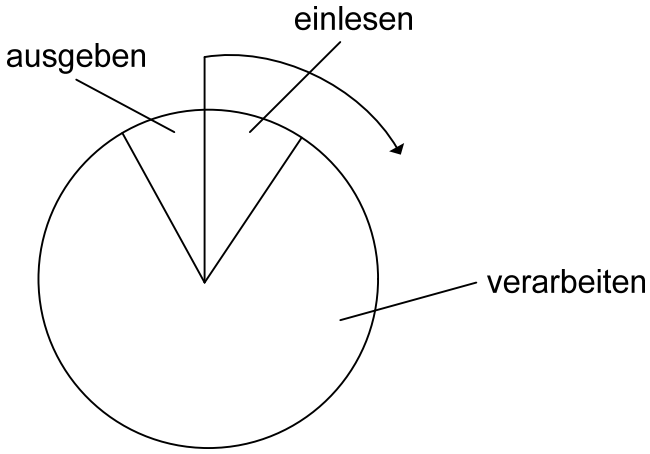
Ja gut, nehmen wir mal an, der Regler ist die Motorsteuerung…
OK, dann liefert der Regler die Spannung – oder was da gebraucht wird – um den Motor schneller oder langsamer laufen zu lassen. Die Strecke ist nicht nur der Motor, sondern alles weitere, wie Getriebe, Räder und so. Also alles das, was die Geschwindigkeit ausmacht. Aus der Strecke raus kommt dann die momentane Geschwindigkeit – oder der momentane Abstand. Den kann aber auch der Sensor ermitteln. Auf jeden Fall fließen die Daten wieder über den Sensor zurück und werden dann mit der Führungsgröße verglichen. Also die Differenz gebildet. Und je nach dem, ob wir jetzt zu nah dran oder zu weit weg vom Fahrzeug vor uns sind, müssen wir jetzt also die Geschwindigkeit erhöhen oder erniedrigen.

Mann kann alles in die Strecke reinpacken, was uns bei der Steuerung nicht interessiert. Wo ist jetzt der Unterschied zur der diskreten Steuerung?
Bei der diskreten Steuerung haben wir diskrete Werte von den Sensoren und beeinflussen die Aktoren auch mit diskreten Werten. Die Grundaufgabe ist aber gleich: Wir wollen das System so beeinflussen, dass es sich wie gewünscht verhält.

Was ist eine SPS oder wie funktioniert sie?
Wollte das



malen. Ich bin auch damit angefangen. Aber er wollte auf das:



hinaus, was ich ihm dann auch hingemalt und erläutert habe.

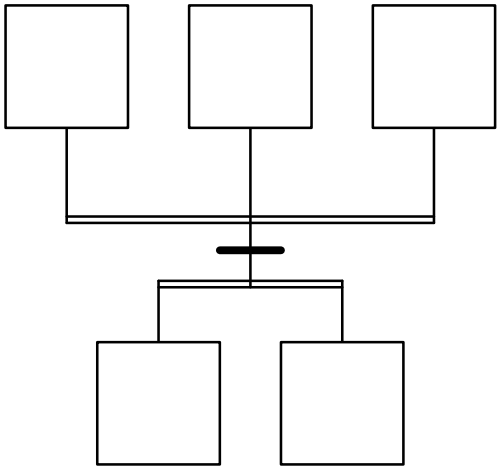
Was kann im permanent zyklischen Betrieb passieren?
Es kann sein, dass man kurzzeitige Ereignisse nicht mitbekommt, da sie nach dem Einlesen auftreten und vor der nächsten Einlese-Phase schon wieder weg sind. Außerdem kann es sein, dass man erst nach 2 Zyklen auf ein Ereignis reagiert. (Zeitdiagramm gemalt und das daran erklärt)

Wie könnte man das Ganze denn modellieren, wenn so was nicht passieren soll?
Zum Beispiel mit Interrupts, aber dann ist es nicht so einfach, dann noch die Echtzeitfähigkeit sicher zu stellen.

Wir hatten das so eine Sprache, um Abläufe zu steuern mit Hilfe einer SPS. Wie hieß die?
Sequential Function Charts.

Was sind die Modellierungselemente bei SFC?
Man hat so was Ähnliches wie Petri-Netze. Nur die Modellierung von parallelem Verhalten ist syntaktisch etwas anders. Man hat alternative Verzeigungen und parallele Verzweigungen, dabei hat man jeweils 2 Striche oben und unten.

Wenn wie hier eine parallele Verzweigung haben, wann kann dieser Übergang stattfinden?



(hier habe ich etwas länger gebraucht, weil ich etwas verunsichert war) Der Übergang kann stattfinden, wenn alle Eingangsplätze belegt sind, die Bedingung wahr ist und wenn alle Ausgängsplätze frei sind.

Wir hatten bei ESD noch mehr Möglichkeiten kennen gelernt, paralleles Verhalten zu modellieren. Wie funktioniert dies?
Wir hatten die Synchronkomposition von Zustandsautomaten.

Was ist die Semantik davon wenn man zwei parallele Automaten hat?
(Beispiel hingemalt).

OK, OK. Was ist denn jetzt die Semantik?
Dass beide Automaten unabhängig voneinander schalten, es sei denn sie sind explizit durch ein Synchronisationslabel synchronisiert.

Was ist daran grundlegend anders als bei der Modellierung mit Petri-Netzen?
Man modelliert mit Petri-Netzen kausale Zusammenhänge und keine temporalen Zusammenhänge. Das macht man deshalb, weil kausale Zusammenhänge objektiv sind und nicht von der Ausbreitungsgeschwindigkeit des Ereignisses abhängen. Wir hatten da ja das Beispiel mit der Ente…

Wir haben ja gerade schon über kontinuierliche Systeme gesprochen. Es gibt jetzt auch noch hybride Systeme. Wann tritt so was auf?
So etwas tritt dann auf, wenn kontinuierliches Verhalten durch ein diskretes Ereignis geschaltet wird. Also ein Umschalten der Differentialgleichung auf der rechten Seite.

Wir hatten ja den Hytech-Algorithmus. Was ist das Problem bei der Erreichbarkeitsanalyse?
Bei hybriden Automaten ist die Erreichbarkeit unentscheidbar, bei rektangulären Automaten semi-entscheidbar. Entscheidbar sind Timed Automata. Aber in der Praxis hat man noch mehr Probleme: Die Integer laufen über und man hat eine kombinatorische Zustandsexplosion, weshalb das Ganze ziemlich oft abstürzt oder ewig läuft.

Fazit
Im Nachhinein sind die Lichter-Fragen durchaus Fragen, die eher ins Standard-Repertoire gehen, allerdings kommt einem das so nicht in der Prüfung vor. Er will zwar kurze Antworten haben, die sollen aber vollständig und genau sein (Beispiel: Abstrakte Klassen). Also zuhause schon mal überlegen, was man zu den Standardfragen sagen würde schadet nicht, aber man

muss darauf gefasst sein, dass dann noch mal genauer nachgefragt wird. So wie ich das aus den Protokollen gelesen habe: „abstrakte Klasse = Schnittstelle“ reicht definitiv nicht! Der Zusammenhang ist extrem wichtig und auch eine eigene Auseinandersetzung damit ist gefragt (Beispiel LCOM). Außerdem ist wichtig, dass man besonders bei OOSK die Begriffe richtig benutzt. Nicht so einfach hab ich die „zähl-mal-alles-auf-was-dir-einfällt“-Fragen empfunden. Zu Prof. Kowalewski kann ich sagen, dass er sich relativ genau an die Stoffzusammenfassung am Ende der jeweiligen Vorlesungen hält. Man muss die deutschen Fachbegriffe aus der Regelungstechnik usw. auf jeden Fall nachschlagen. Prof. Kowalewski möchte eher lange Antworten und besteht auch auf genaue Antworten auf die Fragen. Zur Not gibt er solange Tipps, bis man drauf kommt.