

Datenmodellierung in der Medizin – inoffizieller Fragenkatalog

Zusammengestellt von Murrel (Murrel.vienna@gmx.at)

Teil I

1) Nennen Sie die in UML 2 existierenden Diagrammarten inklusive der Fragen, die diese Diagramme beantworten sollen. (Folien 33-36)

Die wichtigsten Diagramme in UML 2 sind die folgenden 13 (Merkmahlilfe Anfangsbuchstaben um keins zu vergessen [ja, seltsame Merkmahlilfe, aber so kann man es aussprechen und sich merken]: KAKKU KOPSVITZ [einfach vorstellen, das wäre ein schwedischer Maler oder so])

- **Kommunikationsdiagramm:** Wer kommuniziert mit wem?
- **Aktivitätsdiagramm:** Wie läuft ein bestimmter flussorientierter Prozess ab?
- **Kompositions-Strukturdiagramm:** Wie sieht es in der Klasse, in der Komponente aus?
- **Komponentendiagramm:** Wie werden Klassen zu wieder verwendbaren, verwaltbaren Komponenten?
- **Use-case Diagramm:** Was leistet mein System für meine Umwelt?
- **Klassendiagramm:** Aus welchen Klassen besteht mein System und wie stehen diese in Beziehung?
- **Objektdiagramm:** Welche innere Struktur hat mein System zu einem festgelegten Zeitpunkt der Laufzeit?
- **Paketdiagramm:** Wie kann ich mein Modell so schneiden, dass ich den Überblick bewahre?
- **Sequenzdiagramm:** Wer tauscht wann mit wem welche Info aus?
- **Verteilungsdiagramm:** Wie werden Komponenten wohin verteilt?
- **Interaktions-Übersichtsdiagramm:** Wann läuft welche Interaktion ab?
- **Timing-Diagramm:** Wann befindet sich wer in welchem Zustand?
- **Zustandsautomat:** Welche Zustände kann ein Objekt, Interface... bei welchen Ereignissen annehmen?

2) Woraus besteht ein Use-case diagramm? Beschreiben Sie kurz, wie man zu dessen Aufstellung vorgeht und wozu es dient. Gehen Sie auch auf den Unterschied zwischen includes und extends ein (Folien 42 und 46-47)

Diese Anwendungsfallanalyse soll zeigen, was ein geplantes System leisten soll, ohne Details und Realisierung zu beschreiben. Es ist aus Nutzersicht auf hohem Abstraktionsniveau und als fester Bestandteil von Vorgehensmodellen eine Haupttechnik in der Systemanalyse.

Das Diagramm besteht aus Akteuren, Use-Cases, Beziehungen zwischen diesen Use-cases (includes oder extends), Assoziationen zwischen Akteuren und Use-cases und Generalisierungen zwischen Akteuren.

Man geht bei der Erstellung eines Use-cases in 4 Schritten vor:

- Identifizieren der Akteure
 - Identifizieren der Anwendungsfälle
 - Visualisierung mit Anwendungsfalldiagramm
 - Beschreibung der Anwendungsfälle (durch Textschablone oder Verhaltensdiagramme)
- Zeigt ein „includes“ von einem Use-case auf einen anderen, so wird er in diesen immer eingeschlossen. Bei der „extends“ Beziehung ist die Erweiterung des Use-cases durch den anderen optional.

3) Gehen Sie auf die Bestandteile einer Use-Case Beschreibung ein (Folien 55 und 56)

Die Schablone enthält folgende Dinge:

- Name des Anwendungsfalls
- Ziel
- Kategorie (primär/sekundär/optional -> Wichtigkeit)
- Vorbedingung
- Nachbedingung (Erfolg/Fehlschlag)
- Akteure
- Auslösendes Ereignis
- Beschreibung (zB 1. erste Aktion 2. zweite Aktion)
- Erweiterung (zB 1a. Erweiterung zu 1)
- Alternativen (zB 2a. Alternative zu 2. 2b. Weitere Alternative zu 2)

Teil II

4) Wofür verwendet man das Klassendiagramm und woraus besteht es? (Folie 3)

Das Klassendiagramm modelliert die statischen Elemente eines Systems, ihre Eigenschaften und ihre Beziehungen zueinander. Es besteht aus Klassen, Attributen, Operationen, Assoziationen, Generalisationen und Schnittstellen.

5) Wofür verwendet man das Objektdiagramm und woraus besteht es? (Folien 7-10)

Das Objektdiagramm zeigt eine Momentaufnahme der Objekte eines Systems. Es besteht aus Objekten, Links und Rollen.

6) Definieren Sie:

a) Objekt (extern, intern) (Folien 14-15)

Allgemein ist ein Objekt (Exemplar, Instanz) ein Gegenstand des Interesses, also ein Ding (PC), eine Organisation (Ambulanz), eine Person (Patient) oder ein Begriff (Krankheit).

Externe Objekte existieren in der realen Welt und sind aktiv (Patient Huber schickt Krankenschein).

Interne Objekte sind nur für die Softwaremodellierung interessant und oft passiv (über Patient Huber werden Daten gespeichert).

b) Zustand (Folie 16)

Auch state genannt, ein Zustand beinhaltet die durch die Klasse bestimmten Attribute, die aktuellen Werte und die jeweiligen Verbindungen zu anderen Objekten. Eine Änderung oder Abfrage des Zustands ist nur mittels der Operationen möglich.

c) Verhalten (Folie 16)

Menge der Operationen (durch Klasse bestimmt)

d) Identität (Folie 18)

Jedes Objekt besitzt eine Identität, die es von anderen Objekten unterscheidet und die sich nicht ändern kann. Ein Objekt kann auch andere Objekte kennen.

e) Klasse (Folie 20)

Eine Klasse (Typ) definiert für eine Menge von Objekten deren Struktur (Attribute), Verhalten (Operationen) und Beziehungen. Sie kann neue Objekte erzeugen, wobei jedes Objekt genau einer Klasse angehört. Ihre Beziehungen sind Assoziationen und Vererbungsstrukturen.

f) Attribut (Folien 24-25)

Attribute stellen die strukturellen Eigenschaften einer Klasse dar. Alle Objekte einer Klasse haben dieselben Attribute und jedes Attribut hat einen bestimmten Typ. Sie müssen nicht atomar wie im ER-Modell sein (zB Name). Man unterscheidet Instanzenattribut (für jedes Objekt erzeugt, definieren Zustand des Objekts) und Klassenattribut (ein Wert für alle Objekte, unterstrichen).

Attribute können

- public(+, alle Systemkomponenten haben Zugriff),
- private (-, nur die eigene Klasse kann zugreifen),
- protected (#, nur eigene Klassen und Spezialisierungen) oder
- package (~, nur die Klassen, die sich im selben Paket befinden) sein.

g) Operation (Folien 29-30)

Eine Operation (auch Nachricht, Message, Botschaft, Funktion, Prozedur, Service) ist eine ausführbare Tätigkeit. Alle Objekte einer Klasse verwenden dieselben Operationen und die Menge der Operationen wird als Verhalten der Klasse (Schnittstelle) bezeichnet.

h) Assoziation / reflexible Assoziation / Assoziationsklasse(Folien 35-44)

Assoziationen modellieren im Klassendiagramm die Relationen zwischen Klassen, binäre Assoziationen verbinden dabei zwei Objekte. Dabei wird immer die Kardinalität angegeben und optionalerweise Assoziationsname (mit Leserichtungs-pfeil) und Rollennamen.

Man nennt die Assoziation reflexibel, wenn sie zwischen Objekten derselben Klasse besteht. Eine Assoziationsklasse wird modelliert, wenn die Assoziation zusätzlich Eigenschaften einer Klasse, also Attribute und Operationen, besitzt.

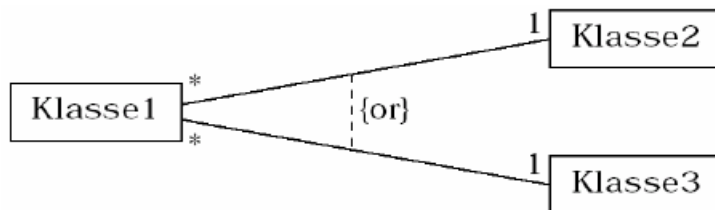
i) Qualifizierte Klasse (Folie 46)

Eine Qualifikationsangabe ist ein spezielles Attribut der Assoziation, dessen Wert ein oder mehrere Objekte auf der anderen Seite selektiert.

7) Nennen Sie die 3 Arten von Restriktionen (+Zeichnung)

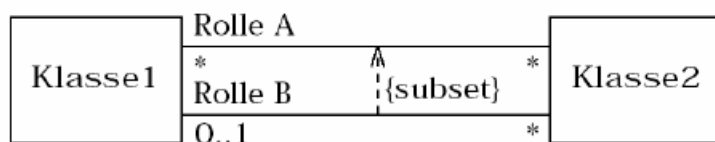
or-Restriktion

(Lagerorte eines Medikamentes)



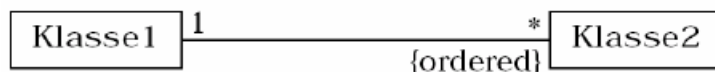
subset-Restriktion

(Rollen Teilnehmer und Sieger)



geordnete Assoz.

(wartende Patienten)



8) Erklären Sie den Unterschied zwischen Aggregation und Komposition (Folien 47-51)

Bei der Aggregation (weißes Kästchen) handelt es sich um eine Zusammensetzung eines Objektes aus einer Menge von Einzelteilen. Das Ganze übernimmt allerdings auch Aufgaben für seine Teile.

Bei der Komposition (schwarzes Kästchen) handelt es sich um eine strenge Form der Aggregation, die Teile sind hierbei vom Ganzen existenzabhängig. Jedes Objekt der Teilklasse kann immer nur Komponente eines Objekts der Aggregatklasse sein.

9) Erklären Sie, wofür die Vererbung verwendet wird und wie sie funktioniert.

Erklären Sie insbesondere auch die Begriffe Diskriminator und Abstrakte

Klasse. Welche Probleme kann die Mehrfachvererbung ergeben? (Folien 52-58)

Eine Vererbung (generalization) beschreibt die Beziehung zwischen einer allgemeinen Klasse (Basisklasse) und einer spezialisierten Klasse. Vererbt werden alle Attribute, Operationen und Assoziationen, die spezialisierte Klasse ist vollständig konsistent mit der Basisklasse, enthält aber zusätzliche Informationen. Unterklassen können das Verhalten verfeinern und überschreiben (gleiche Operationsnamen).

Eine Vererbung kann zusätzlich durch einen Diskriminator beschrieben werden, welcher angibt, nach welchem Unterscheidungsmerkmal die Unterklassen gebildet werden.

Abstrakte Klassen werden durch einen kursiven Namen dargestellt. Sie hat eine wichtige Rolle in der Vererbungsstruktur, indem sie Gemeinsamkeiten definiert. Von ihr selbst können nämlich keine Objekte erzeugt werden.

Bei der Mehrfachvererbung kann jede Klasse mehrere direkte Oberklassen besitzen, wodurch Namenskonflikte entstehen können.

10) Was ist ein Muster? Nennen Sie die 5 in der LVA behandelten Muster und ihre groben Eigenschaften. (Folien 62-73)

Muster erlauben die standardisierte Lösung bestimmter Probleme, da bei der Modellierung häufig ähnliche Probleme vorkommen. Sie werden vor allem für den Entwurf, aber auch in der Analyse eingesetzt. Die 5 in der LVA behandelten Muster waren:

- Liste (Komposition aus gleichartigen Teilen mit mindestens einem Teilobjekt)
- Stückliste (Teilobjekte, die in einer Komposition mehreren Aggregat-Objekten zugeordnet werden können)
- Koordinator (einfache Assoziationen mit Assoziationsklassen in n-ärer Assoziation)
- Wechselnde Rollen (Objekt kann mittels Vererbung mehrere Rollen einnehmen, wobei Objektverbindungen nur erweitert werden)
- Gruppenhistorie (Einzelobjekt gehört im Laufe der Zeit zu mehreren Gruppen-Objekten, Historie modelliert durch assoziative Klasse)

Teil III

11) Was ist ein Stereotyp (+ Beispiele), was eine Schnittstelle? (Folien 6-10)

Ein Stereotyp spezifiziert den Zweck oder die Art der Verwendung eines Notationselements (Bsp: <<interface>>, <<enumeration>>, <<entity>>...).

Eine Schnittstelle modelliert eine Art Vertrag zwischen Klassen. Sie wird mit <<interface>> notiert, ist abstrakt und kann Attribute und Operationen umfassen.

12) Wozu dient ein Paketdiagramm, was sind hierbei die Unterschiede zwischen den Beziehungen import, access und merge? (Folien 11-15)

Ein Paketdiagramm fasst UML-Elemente logisch zusammen, indem es die Elemente durch horizontale Strukturierung in Paketen strukturiert. Mit einer vertikalen Strukturierung werden hierbei unterschiedliche Abstraktionsebenen definiert.

Ein Paket-import importiert hierbei alle Namen eines Paketes als öffentlich, Paket-access als privat. Der Paket-merge verschmilzt Pakete (im Prinzip verkürzte Generalisierung).

13) Was ist Verhaltensspezifikation und aus welchen Teilen besteht sie? (Folien 18-20)

Die Verhaltensspezifikation ist ein spezieller Classifier, welcher spezialisierbar, generalisierbar, hierarchisch zerlegbar, instanzierbar und als Klasse darstellbar ist. Hierbei sind Vor- und Nachbedingungen und Ein- und Ausgabeparameter möglich. Dargestellt wird sie durch verschiedene Sichtweisen -> Verhaltensdiagramme.

14) Erklären Sie das Classifier-Konzept (Folien 21-22)

Das Classifierkonzept wird nur im Metamodell verwendet und ist damit keine grafische Komponente. Es basiert auf Elementen, welche ähnliche Eigenschaften wie Klassen besitzen und verallgemeinert das Konzept Klasse daher noch weiter (zB Datentyp, Akteur, Verhalten), wodurch sich eine Classifier-Hierarchie ergibt.

15) Woraus besteht ein Kommunikationsmodell? (Folien 23-25)

Das Kommunikationsmodell beschäftigt sich mit der Frage: „Wie werden Veränderungen und Vorgänge angestoßen?“ Es besteht aus Nachrichten (Operationsaufruf oder Signal) und Ereignissen (zB Zeitereignis, Änderungsereignis...).

16) Was ist ein Verhaltensdiagramm? (Folie 26)

Ein Verhaltensdiagramm dient der Darstellung der Verhaltensspezifikation aus verschiedenen Blickwinkeln. Verhaltensdiagramme sind die folgenden Diagrammtypen:

- Zustandsautomat
- Use-Case Diagramm
- Aktivitätsdiagramm
- Sequenzdiagramm (=Interaktionsdiagramm)
- Kommunikationsdiagramm (=Interaktionsdiagramm)
- Interaktionsübersichtsdiagramm (=Interaktionsdiagramm)
- Zeitverlaufsdiagramm (=Interaktionsdiagramm)

17) Wozu dient das Aktivitätsdiagramm und was sind seine Bestandteile? (Folien 29-46)

Aktivitätsdiagramme modellieren das Verhalten von Systemen unter Verwendung eines Kontroll- und Datenflussmodells. Es wird zur Zerlegung einer Aufgabe in Einzelschritte verwendet und um Reihenfolge, Nebenläufigkeiten und alternative Entscheidungswege aufzuzeigen. Es besteht aus Aktivitäten (beschreibt Ausführung von Funktionalität, = gesamte Diagrammeinheit), Aktivitätsbereichen (Bereiche mit bestimmten Gemeinsamkeiten, mehrdimensional möglich), Aktionen (Einzelbestandteile, Sonderformen: Signalempfang, -Sendung, Zeitereignis). Neben den (gerichteten) Kanten existieren ebenfalls Objektknoten, welche eine Art Objektspeicher darstellen und den Objektfluss, welcher mit PINS den Transport der Objekte repräsentiert.

18) Welche Arten von Kontrollknoten existieren im Aktivitätsdiagramm? (Folien 47-50)

Man unterscheidet Start- und Endknoten, Entscheidung (-> splittet die Möglichkeiten) und Verbindung (-> führt sie zusammen) und Gabelung und Vereinigung (Parallelisierung und Synchronisation des Kontrollflusses).

19) Erklären Sie das Tokenkonzept. Woraus besteht ein Schleifenknoten, woraus ein Bedingungsknoten? (Folien 51-54)

Im Tokenkonzept markiert der Token den Punkt, wo sich der Ablauf gerade befindet. Hierbei wandern beliebig viele Tokens durch die Aktivität, wobei eine Aktion durch Tokens ausgelöst wird und danach über eine Kante an die nächste übergeben wird. Diese Übergänge sind zeitlos.

Ein Schleifenknoten besteht aus Initialisierung (for), Test (while) und Schleifenkörper (do). Ein Bedingungsknoten aus ein oder mehreren if-Bedingungen (exklusiv zueinander), einem then und einem else-Bereich.

Teil IV

20) Was ist ein Szenario? (Folien 7-8)

Ein Szenario ist eine Sequenz von Verarbeitungsschritten. Es wird unter bestimmten Bedingungen ausgeführt, durch Ereignisse ausgelöst und kann gemeinsam mit anderen Szenarien einen Use-case dokumentieren.

21) Welche Arten von Interaktionsdiagrammen kennen Sie? Was sind deren Grundelemente? (Folien 10-12)

Interaktionsdiagramme sind:

- Sequenzdiagramm
- Kommunikationsdiagramm
- Interaktionsübersichtsdiagramm
- Timing-diagramm

Ihre Grundelemente sind Nachrichten und Lebenslinien.

22) Wofür und wie verwendet man das Sequenzdiagramm? In welche Dimensionen ist es unterteilt? (Folien 18-25)

Sequenzdiagramme definieren den Nachrichtenfluss zwischen Objekten. Sie zeigen den zeitlichen Ablauf der Nachrichten, beinhalten aber keine Informationen über Beziehungen der Objekte. Sie werden zur Analyse von Szenarien verwendet, wobei man sich auf den Nachrichtenaustausch und nicht auf die Darstellung aller Ablaufpfade konzentriert. Sie bestehen aus einem Rahmen mit Kürzel und Namen der Interaktion, den Kommunikationspartnern als Lebenslinien und den Nachrichten als Pfeile. Die beiden Dimensionen des Diagramms sind die Kommunikationspartner (horizontal) und die Zeit (vertikal).

23) Welche Arten von Nachrichten existieren in Sequenzdiagrammen? (Folien 27-30)

Nachrichten können synchron (Sender wartet auf Antwort), asynchron (Sender setzt Aktion fort) oder Antworten sein.

Die Notation ist Name[(Argument)] [:Rückgabewert]

24) Wozu dienen gefundene und verlorene Nachrichten? (Folien 35-36)

Gefundene und verlorene Nachrichten dienen zur Aufteilung umfangreicher Szenarien auf mehrere Diagramme. Sie zerlegen somit die Lebenslinien und beschreiben das Zusammenspiel der inneren Strukturteile eigenständig.

25) Wofür und wie verwendet man das Kommunikationsdiagramm, welche Notationselemente existieren darin? Wie werden Nachrichten angeschrieben? (Folien 40-46)

Ein Kommunikationsdiagramm beschreibt die Interaktion zwischen Objekten mit dem Fokus auf die Struktur ihrer Kommunikationsbeziehungen. Es zeigt Interaktionen von Teilen einer komplexen Struktur und beschreibt so das Zusammenwirken dieser Teile, ohne genaue zeitliche Übergänge zu unterstreichen.

Das Diagramm besteht aus Interaktionsrahmen, Lebenslinien und Nachrichten.

Die Reihenfolge der Nachrichten wird durch ihre Gliederung, welche sequenziell oder geschachtelt ist, bestimmt (1, 1.1, 1.2, 1.2.1 ...). Parallele Ausführung wird durch Buchstaben (1.2a, 1.2b), bedingte mit guard (1.2[Patient atmet nicht]) und iterative mit * (2*[für jeden Patienten] Blutdruck messen) bezeichnet.

26) Wofür und wie verwendet man das Interaktionsübersichtsdiagramm, woraus besteht es? (Folien 50-53)

Ein Interaktionsübersichtsdiagramm stellt einen Überblick über die Interaktionen und deren Reihenfolge dar. Es ist eine Spezialisierung des Aktivitätsdiagramms, statt Aktionen werden nur Interaktionsdiagramme verwendet. Damit modelliert es eine Gesamtübersicht über die Zusammenhänge und Reihenfolge einzelner Interaktionen auf hohem Abstraktionsniveau.

Es besteht aus Interaktion und Interaktionsreferenz, Interaktionsrahmen, Kontrollfluss und Kontrollknoten.

Teil V

27) Was ist ein Zustandsdiagramm, welche Arten davon existieren? (Folien 5-7)

Zustandsdiagramme modellieren wie Aktivitätsdiagramme das dynamische Verhalten eines Systems, wobei der Fokus aber auf den Reaktionen statt den Aktionen liegt. Damit sind sie zur Darstellung des Verhaltens von Benutzeroberflächen gut geeignet.

Man unterscheidet:

- Verhaltens-Zustandsautomaten (Zur Beschreibung von Use-cases und Klassen in der Analyse)
- Protokoll-Zustandsautomaten (Um zulässige Aufrufe zustandsabhängiger Operationen einer Klasse in der Analyse darzustellen)

28) Was ist ein Zustandsautomat und woraus besteht er? (Folien 8-12)

Der Zustandsautomat basiert auf dem Konzept der deterministischen endlichen Automaten, er besteht aus Zuständen und Zustandsübergängen (Transitionen) initiiert durch interne oder externe Ereignisse und charakterisiert so das Verhalten von Classifiern. Alle Objekte einer Klasse haben denselben Zustandsautomaten, er besitzt immer einen Anfangszustand und kann einen Endzustand besitzen. Er wird mit „sm“ (state machine) gekennzeichnet, ist in Regionen aufgeteilt und besitzt folgende Komponenten:

- Transitionen
- Ereignisse (lösen Transitionen aus)
- Zustände (werden beim Betreten aktiv, beim verlassen inaktiv)

29) Was für Arten von Zuständen existieren? Was ist der Unterschied zwischen Ende und Terminator? Zwischen Entscheidung und Kreuzung? (Folien 17-23)

Man unterscheidet einfache und zusammengesetzte (werden durch eigenen Zustandsautomaten dargestellt -> Hierarchiebildung möglich) Zustände.

Der Unterschied zwischen Ende und Terminator ist, dass der Endzustand nur eine Region oder Ebene beendet, während der Terminator den Zustandsautomaten voll und ganz beendet.

Während die Entscheidung eine Entscheidung zwischen Alternativen ist, ist die Kreuzung eine Verknüpfung von Transitionen ohne Zustände.

30) Was ist ein Dialog? Wie teilt man Dialoge ein? (Folien 29-31)

Ein Dialog ist eine Interaktion zwischen einem Benutzer und einem Dialogsystem, um ein bestimmtes Ziel zu erreichen. Man unterteilt Dialoge in:

- technisch (modal: muss vor nächster Aufgabe beendet werden, nicht-modal: ermöglicht Handlungsflexibilität durch mögliche Unterbrechung)
- nach der Anzahl bearbeitbarer Dokumente (single document interface [SDI] oder multiple document interface [MDI])
- nach Benutzersicht (Primärdialog: Arbeitsschritte zur direkten Aufgabenerledigung, Sekundärdialog: für zusätzliche situationsbezogene Entscheidungen)
- nach Bedienungsart (funktionsorientiert: zuerst Funktion dann Objekt, objektorientiert: zuerst Objekt dann Funktion, direkte Manipulation: zB „pick“, „drag“ & „drop“)

31) Nennen Sie die 7 ergonomischen Grundsätze und erläutern sie jeden kurz. (Folien 33-36)

Die 7 ergonomischen Grundsätze sind (Merkhilfe Anfangsbuchstaben: IS FALSE):

- **Individualisierbarkeit:** Anpassung an Fähigkeiten und Vorlieben möglich
- **Selbstbeschreibungsfähigkeit:** Durch Rückmeldungen verständlich oder Erklärung auf Anfrage
- **Fehlertoleranz:** Bei Fehlern sollte das Ergebnis trotzdem mit wenig Korrekturaufwand erreicht werden
- **Aufgabenangemessenheit:** Benutzer kann Aufgabe effektiv und effizient erledigen
- **Lernförderlichkeit:** Unterstützung und Anleitung beim Erlernen
- **Steuerbarkeit:** Start, Richtung und Geschwindigkeit des Dialogs bestimmbar
- **Erwartungskonformität:** Dialog ist konsistent und den Konventionen des Benutzers entsprechend

32) Wozu dient das Erfassungsfenster, wozu das Listenfenster? (Folien 38-39)

Das Erfassungsfenster dient zum Erfassen und ändern einzelner Objekte der Klasse, es bildet jedes Attribut auf ein Interaktionselement ab und jede Operation auf eine Schaltfläche oder eine Menüoption. Schaltflächen hier wären: OK, Übernehmen, Abbrechen, Liste

Das Listenfenster zeigt alle Objekte der Klasse und beschreibt sie durch einen Teil der Attribute. Schaltflächen hier wären: Neu, Ändern, Löschen, Schließen

33) Was für Arten von Vererbungsstrukturen existieren und wodurch unterscheiden sie sich? (Folien 43-45)

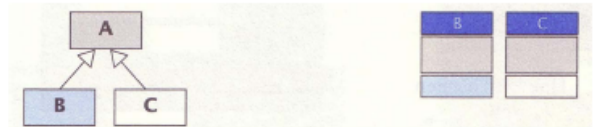
1. Konkrete Oberklasse

- Fenster für Oberklasse
- Fenster der Unterklassen enthalten Elemente der Oberklasse



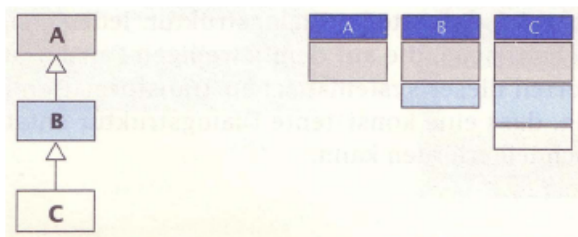
2. Abstrakte Oberklasse

- Kein eigenständiges Fenster für Oberklasse
- Unterklassen analog 1.



3. Mehrstufige Vererbung

- Analog 1. und 2.



34) Was sind die wichtigen Mittel der Gestaltung? Was ist dabei zu beachten? Was ist bei der Farbgebung zu beachten? (Folien 46-50)

Wichtig sind:

- **Gruppierung:** semantisch zusammengehörende Objekte zusammen, denn der Benutzer orientiert sich zunächst an den Gruppen – optimal sind 4-5 Gruppen mit 4-5 Objekten und spaltenweiser Anordnung
- **Hervorhebung:** Oft vorteilhaft, die Aufmerksamkeit des Benutzers mittels Größe, Farbe, Umrahmung, Isolierung zu lenken – bei maximal 5 Farben und maximal 10-20% hervorgehobenen Dingen

Bei der Farbgebung ist zu beachten, dass man farbig und nicht bunt gestaltet und man Farbkonventionen beachtet (Bsp: Rot = Signalfarbe). Farbtonunterschiede sind im Gelb- und Blaubereich leichter als im Rotbereich zu erkennen.

35) Erläutern Sie kurz die wichtigsten ästhetischen Aspekte (Folien 53-55)

Die wichtigsten ästhetischen Aspekte sind:

- Sequenz: Das Auge soll sequenziell durch das Fenster geführt werden
- Einfachheit: Jedes Fenster soll so einfach wie möglich sein
- Proportionen: Flächen erscheinen angenehmer, wenn sie breiter sind, als sie hoch sind
- Balance: Wenn das Fenster durch eine vertikale Linie getrennt wird, sollte die Informationsdichte auf beiden Seiten gleich sein
- Symmetrie: Verstärkung zur Balance, Horizontale gegenüberliegende Elemente sollten in Größe und/oder Art gleichartig sein

36) Was sind die Anforderungen an klinische Formulare? (Folien 59-60)

Klinische Formulare sollen die Dokumentation unterstützen, die Auswertung der Daten ermöglichen, leicht modifizierbar sein und Aufgaben verschiedener Usergruppen durch vordefiniertes Wissen möglichst beschleunigen.

Diese Anforderungen sind zum Teil schwer vereinbar. Problematisch ist die Wahl zwischen Freitext und strukturierten Daten, da ersteres mehr Freiheit, zweiteres aber spätere Analysen und Entscheidungsprozesse ermöglicht.

Teil VI

37) In welche Gruppen kann man Anforderungsbeitragende einteilen? (Folien 15-16)

Anforderungsbeitragende, auch Stakeholder genannt, sind folgende Personen:

- Anwender
- Auftraggeber
- Geldgeber
- Gesetzgeber
- Systementwickler
- Systembetreuer
- Management

38) Was ist bei der Verwendung von Werkzeugen zu beachten? (Folien 19-20)

Man braucht methodische Regeln für jedes Konzept, die den Analytiker unterstützen, aber nicht einengen. Das Ziel sind quantitativ gleichwertige Modelle. Zur Unterstützung der Kommunikation im Team ist es wichtig, schnell zu einer ersten Version des Modells zu kommen, die dann überarbeitet und verfeinert wird.

39) Nennen Sie die drei großen Modelle bei der Entwicklung und die Arbeitsschritte, die sie umfassen (Folien 21-27)

Das OO-Analysemodell ist die Analyse im Großen, bei der Use-cases aufgestellt werden und Pakete (Teilsysteme) gebildet.

Darauf folgt das statische Modell, bei dem nacheinander die Klassen, Assoziationen, Attribute und Vererbungsstrukturen mithilfe eines Klassendiagramms identifiziert werden, um dann die Assoziationen mittels eines Objektdiagramms zu vervollständigen und schließlich die Attribute zu spezifizieren.

Schließlich werden im dynamischen Modell Szenarien und Zustandsdiagramme erstellt und Operationen eingetragen und beschrieben.

40) Definieren Sie: Entwurf, OOD-Modell (Folie 40)

Entwurf: Aufbauend auf Ergebnis der Analyse die Erstellung der Architektur, der Komponenten und deren Schnittstellen, Funktions- und Leistungsumfang.

OOD-Modell: Technische Lösung des zu realisierenden Systems in objekt-orientierter Notation und Abbild der späteren Programme.

41) Wie müssen die folgenden Konzepte im Entwurf erweitert werden: (Folien 43-53)

a) Klasse

Die Klassen haben dieselbe Notation wie in der Analysephase und dabei Erweiterungen wie „parametrisierte Klassen“

b) Attribut und Operation

Die Namen müssen an die Konventionen der Programmiersprache angepasst, die Sichtbarkeit verfeinert und die Signatur der Operanden vollständig angegeben werden.

c) Assoziation

Der Zugriff (Navigation) wird durch Pfeilspitzen festgelegt (uni/bidirektional).

d) Polymorphismus

Wird erst in der Implementierung richtig genutzt (overloading != Monomorphismus).

e) Vererbung

Es wird Mehrfachvererbung benötigt, daraus entsteht die Gefahr der Spaghetti-Vererbung und Namenskonflikte können entstehen. Man spezifiziert daher Restriktionen (overlapping – disjoint; complete – incomplete)

f) Zustandsmuster

Jede Klasse mit einem Objekt-Lebenszyklus erhält eine abstrakte Klasse, jeder Zustand wird in einer eigenen Unterklasse repräsentiert.

42) Was ist bei der Umsetzung vom Klassendiagramm zu Tabellen zu beachten? (Folien 55-56)

Jede Klasse wird zu einer Tabelle, die unabhängig zu fachlichen Schlüsseln einen OID (object identifier) als primary key bekommt. Vererbungshierarchien können eine oder mehreren Tabellen sein. Assoziationen mit m:n Beziehungen werden eigene Tabellen.

43) Erläutern Sie die Vor/Nachteile von generischen und spezifischen Modellen. (Folie 62)

Generische Modelle:

Vorteile:

- mächtiger
- breite Anwendbarkeit
- weniger Konzepte

Nachteile:

- schwerer verständlich
- größerer Interpretationsaufwand

Spezifische Modelle:

Vorteile:

- einfacher verständlich
- Anwendung klar vorgegeben
- besser zu verifizieren

Nachteile:

- Größere Komplexität
- Weniger universell einsetzbar

44) Was sind Archetypen? (Folien 66-67)

Archetypen dienen der Modellierung wiederkehrender medizinischer Konzepte. Sie sind vergleichbar mit Lego-Steinen aus Lego-plänen.