

bezeichnen.

Was ist Modellierung?

Modellierung
WS 17/18

Einführung

Modellierung

Modellierung ist der Prozess, bei dem ein Modell eines Systems erstellt wird.

Warum sollte man modellieren?

~> Um ein System zu entwerfen, besser zu verstehen, zu visualisieren, zu simulieren, ...

Um etwas konkreter zu werden, betrachten wir Beispiele in verschiedenen Disziplinen (Physik, Biologie, Klimaforschung).

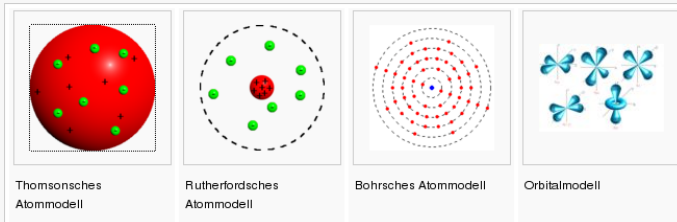
Modellierung in der Physik

Modellierung
WS 17/18

Einführung

Atommodelle

Atome bestehen aus Protonen, Neutronen und Elektronen. Wie diese Teilchen zusammenwirken, wird in Atommodellen beschrieben, die im Laufe der Zeit immer wieder verändert wurden.



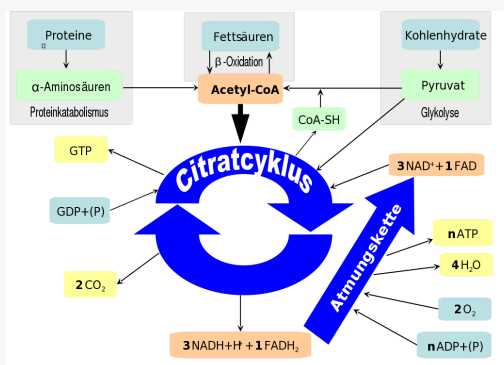
Modellierung in der Biologie

Modellierung
WS 17/18

Einführung

Zitronensäurezyklus

Der Zitronensäurezyklus oder Citratzyklus modelliert den Abbau organischer Stoffe im Körper.

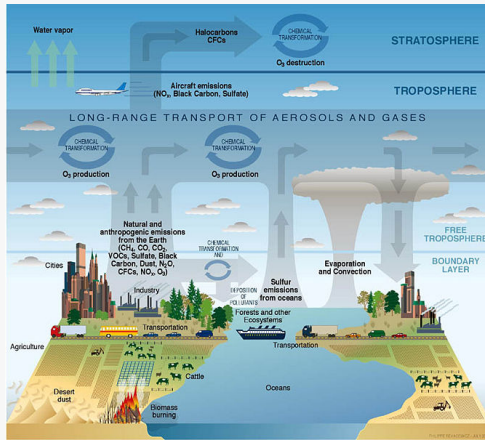


Modellierung in der Klimaforschung

Modellierung
WS 17/18

Einführung

Modell des Transports von Gasen in der Atmosphäre



Arten von Modellen

Modellierung
WS 17/18

Einführung

visuell vs. textuell

Nicht alle Modelle sind visuell bzw. grafisch. Auch mit textuellen Beschreibungen und Formeln kann man modellieren (siehe beispielsweise mathematische Modelle).

Dennoch werden häufig grafische Darstellungen benutzt, auch aus didaktischen Gründen und um sich besser über die Modelle verständigen zu können.

Arten von Modellen

Modellierung
WS 17/18

Einführung

qualitativ vs. quantitativ

- **qualitative Modelle:** Welche Objekte gibt es? Was passiert? Warum passiert es? In welcher Reihenfolge geschehen die Ereignisse? Was sind die kausalen Zusammenhänge? Welche Phänomene treten auf?
- **quantitative Modelle:** Wieviele Objekte gibt es? Wie lange dauert ein Vorgang? Wie wahrscheinlich ist ein bestimmtes Ereignis?

black box vs. white box

- black box: Nur das von außen beobachtbare Verhalten wird beschrieben.
- white box: Es wird auch beschrieben, wie das von außen beobachtbare Verhalten im „Inneren“ des Systems erzeugt wird.

[illegible]

statisch vs. dynamisch

- Ein statisches Modell beschreibt einen Zustand des Systems zu einem bestimmten Zeitpunkt.
- Ein dynamisches Modell beschreibt hingegen auch, wie das System sich entwickelt (ein oder mehrere mögliche Abläufe oder sogar das gesamte Systemverhalten).

[illegible]

formal vs. semi-formal vs. nicht-formal

Je nach Exaktheit der Modelle erhält man:

- formale Modelle, die vollkommen exakt in ihren Aussagen sind (vor allem mathematische Modelle)
- semi-formale Modelle, die teilweise exakt sind, jedoch nicht alles vollständig spezifizieren
- nicht-formale Modelle, die als grobe Richtlinie dienen können, jedoch eher vage Aussagen machen

[illegible]

Probleme mit nicht-formalen Modellen

Modellierung
WS 17/18

Einführung

Natürliche Sprache ist nicht immer eindeutig.

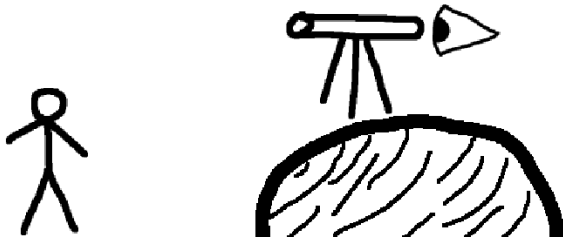
Beispiel:

Ich sah den Mann auf dem Berg mit dem Fernrohr.

Probleme mit nicht-formalen Modellen

Modellierung
WS 17/18

Einführung

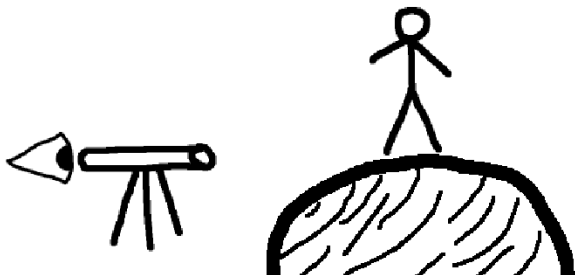


((Ich sah den Mann) auf dem Berg) mit dem Fernrohr)

Probleme mit nicht-formalen Modellen

Modellierung
WS 17/18

Einführung



((Ich sah (den Mann auf dem Berg)) mit dem Fernrohr)



((Ich sah den Mann) (auf dem Berg mit dem Fernrohr))



(Ich sah ((den Mann auf dem Berg) mit dem Fernrohr))



(Ich sah (den Mann (auf dem Berg mit dem Fernrohr)))

Probleme mit nicht-formalen Modellen

Modellierung
WS 17/18

Einführung



((((Ich sah den Mann) auf dem Berg) mit dem Fernrohr)



((Ich sah (den Mann auf dem Berg)) mit dem Fernrohr)



((Ich sah den Mann) (auf dem Berg mit dem Fernrohr))

5 mögliche Interpretationen!



(Ich sah ((den Mann auf dem Berg) mit dem Fernrohr))



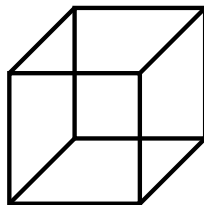
(Ich sah (den Mann (auf dem Berg mit dem Fernrohr)))

Probleme mit nicht-formalen Modellen

Modellierung
WS 17/18

Einführung

Auch grafische Darstellungen können uneindeutig sein:



Modellierung in der Informatik

Modellierung
WS 17/18

Einführung

In dieser Lehrveranstaltung geht es um Modellierungsmethoden in der Informatik.

Diese werden zum Entwurf verschiedener Systeme eingesetzt:

- (Objekt-orientierte) Programme
- (Große) Software-Systeme
- Benutzeroberflächen
- Datenbanken
- Virtual Reality, Computer-Spiele
- ...

Wozu benötigt man in der Informatik Modelle?

Je komplexer ein Informatik-System sein wird, desto wichtiger ist es, einen Plan zu haben, bevor man beginnt, es zu konstruieren.

Dies führt idealerweise zu:

- Vermeidung von Fehlern
- besserer Qualität
- niedrigeren Kosten
- besserer Dokumentation und Wiederverwendbarkeit

[illegible]

Wozu ist Modellierung gut?

Analogie: Bau eines Hauses

Beim Bau einer Hundehütte kann man zumeist ohne große Planung vorgehen. Die Hütte kann von einer einzelnen Person erstellt werden und ein Hund hat zumeist keine großen Anforderungen.

[illegible]

Wozu ist Modellierung gut?

Analogie: Bau eines Hauses

Beim Bau eines Einfamilienhauses ist Planung viel wichtiger. Die Familie ist anspruchsvoller als ein Hund, Bauvorschriften müssen eingehalten werden und vermutlich werden nicht alle Arbeiten von derselben Person durchgeführt.

[illegible]

Wozu ist Modellierung gut?

Modellierung
WS 17/18

Einführung

Analogie: Bau eines Hauses

Der Bau eines Hochhauses ist ohne vorherige Erstellung eines detaillierten Plans bzw. Modells nicht möglich. Das Risiko, Fehler zu machen, ist sehr groß und Fehler können extrem kostspielig werden.



Wozu ist Modellierung gut?

Modellierung
WS 17/18

Einführung

Schwierigkeiten beim Entwurf komplexer Systeme

- Menschen können sich komplexe Systeme normalerweise nicht in vollem Umfang vorstellen.
- Bei mehreren Beteiligten gibt es oft unterschiedliche Meinungen darüber, wie das System aussehen muss.
 ↳ Modelle dienen zur Kommunikation!

Modellierung ist in der Informatik weniger verbreitet als in den Ingenieurwissenschaften, aber ebenso relevant.

Besonderheiten von Software beeinflussen Rolle von Modellen:

- Immaterialität, daher gar nicht so leicht, etwa festzustellen, „wie viel“ eines Modells schon umgesetzt wurde.
- Einzigartigkeit jedes Softwareprojekts, daher Modelle nicht in der Rolle von Vorlagen zum Zweck mehrfacher Realisierung.

Probleme bei der Entwicklung großer Systeme

Modellierung
WS 17/18

Einführung

Feststellung (nach Glinz)

Die Entwicklung von Klein-Software unterscheidet sich fundamental von der Entwicklung größerer Software.

Klein-Groß-Gegensätze in der Software-Entwicklung:

klein	groß
Programme bis ungefähr <u>300 Zeilen</u>	<u>Längere</u> Programme
Für den <u>Eigengebrauch</u>	Für den Gebrauch durch <u>Dritte</u>
<u>Vage Zielsetzung</u> genügt, das Produkt ist seine eigene Spezifikation	<u>Genaue Zielbestimmung</u> , das heißt explizite Anforderungsanalyse, erforderlich

Probleme bei der Entwicklung großer Systeme

Modellierung
WS 17/18

Einführung

klein	groß
Ein Schritt vom Problem zur Lösung genügt: Lösung wird direkt programmiert	Mehrere Schritte vom Problem zur Lösung erforderlich: Anforderungsanalyse, Spezifikation, Konzept, Entwurf und Programmieren der Teile, Zusammensetzen, Inbetriebnahme
<u>Validierung</u> (Überprüfung/Testen) und nötige Korrekturen finden am Endprodukt statt	<u>Auf jeden Entwicklungsschritt muss ein Prüfschritt folgen,</u> sonst kann Endergebnis unbrauchbar werden

[illegible]

Probleme bei der Entwicklung großer Systeme

Modellierung
WS 17/18

Einführung

klein	groß
Eine Person entwickelt: keine Kooperation und Kommunikation erforderlich	Mehrere Personen entwickeln gemeinsam: Koordination und Kommunikation notwendig
<u>Komplexität</u> des Problems in der Regel <u>klein</u> , Strukturieren und Behalten der Übersicht nicht schwierig	<u>Komplexität</u> des Problems <u>größer bis sehr groß</u> , explizite Maßnahmen zur Strukturierung und Modularisierung erforderlich

[illegible]

Probleme bei der Entwicklung großer Systeme

Modellierung
WS 17/18

Einführung

klein	groß
Software besteht aus <u>wenigen Komponenten</u>	Software besteht aus <u>vielen Komponenten</u> , die spezielle Maßnahmen zur Komponentenverwaltung erfordern
In der Regel wird <u>keine Dokumentation</u> erstellt	<u>Dokumentation</u> dringend erforderlich, damit Software wirtschaftlich betrieben und gepflegt werden kann

[illegible]

Probleme bei der Entwicklung großer Systeme

Modellierung
WS 17/18

Einführung

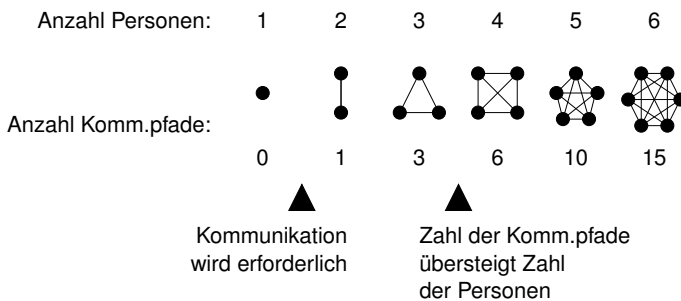
klein	groß
Keine Planung und Projektorganisation erforderlich	Planung und Projektorganisation zwingend erforderlich für eine zielgerichtete, wirtschaftliche Entwicklung

Probleme bei der Entwicklung großer Systeme

Modellierung
WS 17/18

Einführung

Starker Anstieg der Anzahl der Kommunikationspfade bei Anstieg der Entwicklerzahl:

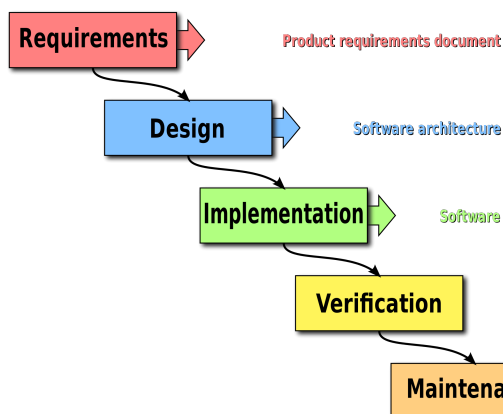


Wer modelliert, und wann?

Modellierung
WS 17/18

Einführung

Ein klassischer Softwareentwicklungs-Prozess, „Wasserfall“:

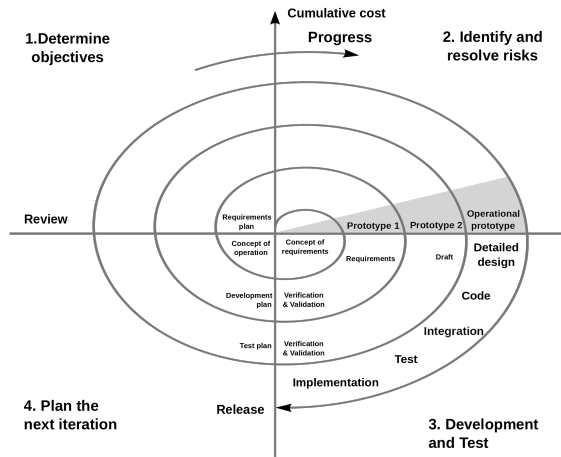


Wer modelliert, und wann?

Modellierung
WS 17/18

Alternativer iterativer Ansatz, „Spiral“:

Einführung



Wer modelliert, und wann?

Modellierung
WS 17/18

„Sonderfälle“:

- Agile Software Development
- Open Source Development
- Model Driven Engineering
- ...

Weitere Aspekte

Modellierung
WS 17/18

Weitere wichtige Gesichtspunkte sind:

- Analyse:
 - Ist ein Modell korrekt? Ist es in sich konsistent?
 - Stimmt das Modell mit der späteren Implementierung überein?
(Hier werden Verfahren zum Testen und zur Verifikation benötigt.)
- Werkzeuge, Software-Tools:
... werden benötigt zum Zeichnen, zum Darstellen (und Wechseln zwischen verschiedenen Darstellungen), zum Archivieren, zur Code-Generierung, zur Analyse, ...

Inhalt der Lehrveranstaltung

Modellierung
WS 17/18

Einführung

Inhalt

- Mathematische Grundlagen
- Graphen für statische und dynamische Systembeschreibungen
- Petrinetze
- UML (Unified Modeling Language)

Aus Gründen der Übersichtlichkeit und Didaktik werden wir uns hauptsächlich mit kleinen Modellen befassen.

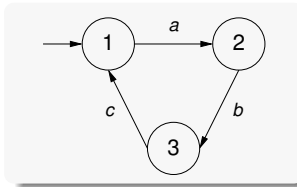
Die Verwendung von Modellen zur Verifikation von Eigenschaften wird nicht im Zentrum stehen.

Graphen

Modellierung
WS 17/18

Einführung

- Graphen sind netzartige Strukturen, bestehend aus Knoten und Kanten.
- Sie können eingesetzt werden für
 - Statische Modellierung: Komponenten und Beziehungen zwischen den Komponenten
 - Dynamische Modellierung: Zustände und Übergänge zwischen den Zuständen

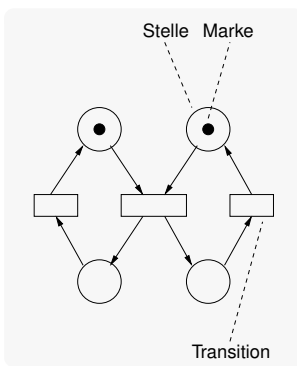


Petrinetze

Modellierung
WS 17/18

Einführung

- Modellierung nebenläufiger und verteilter Systeme, zur Beschreibung der gemeinsamen Nutzung von Ressourcen.
- Schwerpunkt liegt auf der Modellierung des dynamischen Verhaltens.
- Etablierter Ansatz, der vielfältig eingesetzt wird.
- Besitzt formale Semantik.
- Entwickelt von Carl Adam Petri (1962).

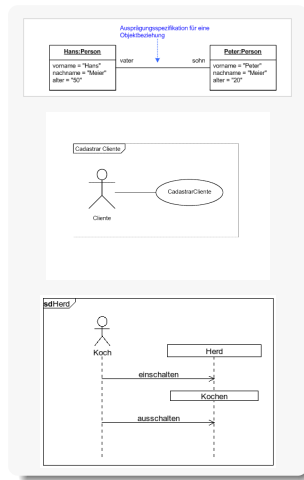


UML: Unified Modeling Language

Modellierung
WS 17/18

Einführung

- Standard-Modellierungssprache für Software Engineering.
- Basiert auf objekt-orientierten Konzepten.
- Sehr umfangreich, enthält viele verschiedene Typen von Modellen.
- Entwickelt von Grady Booch, James Rumbaugh, Ivar Jacobson (1997).



Inhalt der Lehrveranstaltung

Modellierung
WS 17/18

Einführung

Die vorgestellten Modellierungsmethoden sind nicht die einzigen Modellierungsmethoden in der Informatik.

Hier: Fokus auf visuelle Modellierung mit Hilfe von Diagrammen

Mögliche Alternative: algebraische Modellierungsmethoden, die sich stärker an der Mathematik orientieren

Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Einführung

Wir modellieren folgendes System, um eine mögliche Lösung zu finden.

Wolf-Ziege-Kohlkopf-Problem

Ein Farmer will einen Fluss überqueren. Er hat einen Wolf, eine Ziege und einen Kohlkopf bei sich. Wenn sie allein gelassen werden, so frisst die Ziege den Kohlkopf und der Wolf die Ziege. Zur Überquerung des Flusses steht ein Boot mit zwei Plätzen zur Verfügung. Nur der Farmer kann rudern und er kann das Boot entweder allein benutzen oder ein Tier oder den Kohlkopf mitnehmen.

Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Statisches Modell I: Beteiligte Akteure/Objekte

Einführung

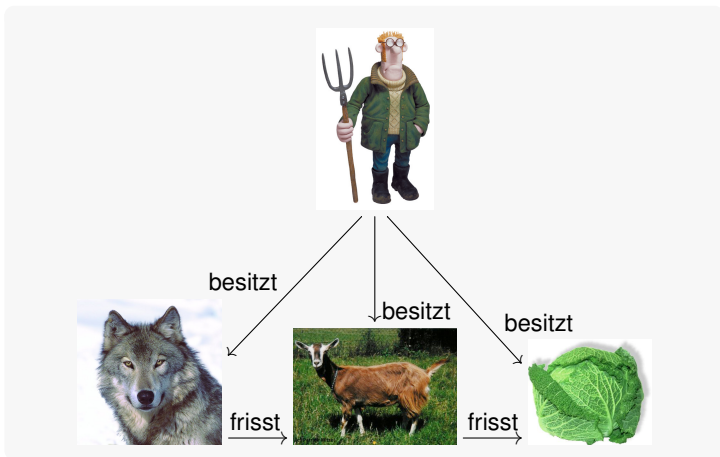


Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Statisches Modell II: Fress- und Eigentumsbeziehungen zwischen den Akteuren

Einführung

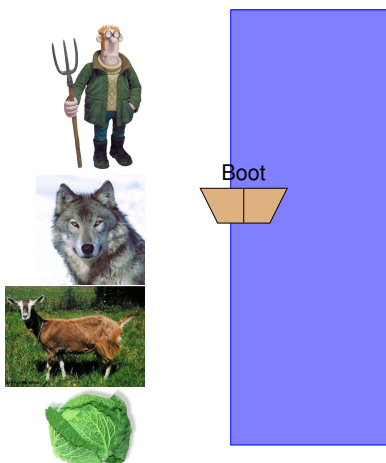


Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Ausgangssituation: vor Überquerung des Flusses

Einführung

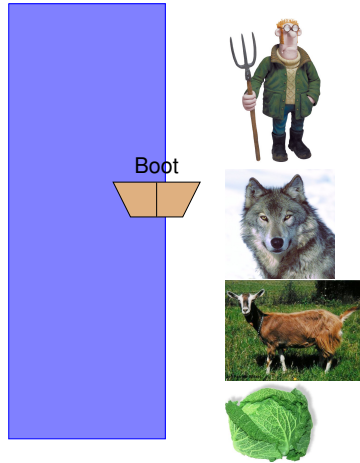


Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Zielsituation: nach Überquerung des Flusses

Einführung

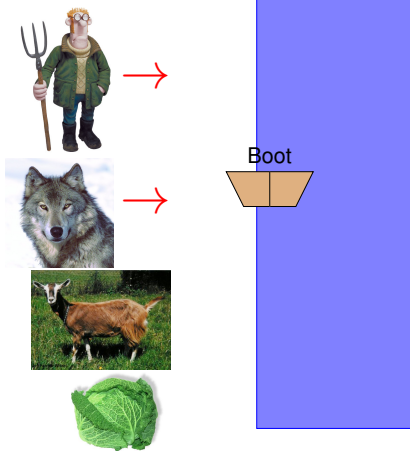
[illegible]

Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Dynamisches Modell: Beispielablauf, erster Schritt;
Farmer und Wolf setzen gemeinsam über

Einführung

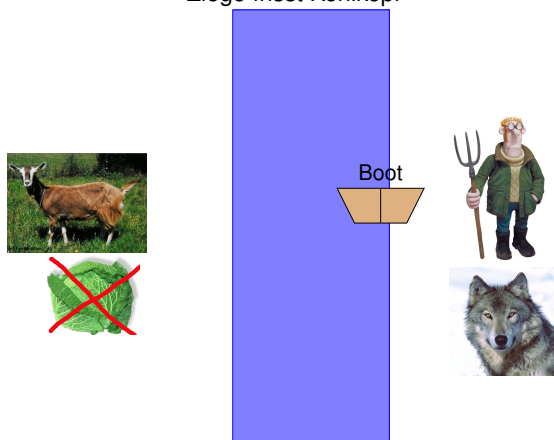
[illegible]

Beispiel: Wolf, Ziege, Kohlkopf

Modellierung
WS 17/18

Dynamisches Modell: Beispielablauf, zweiter Schritt;
Ziege frisst Kohlkopf

Einführung

[illegible]

Syntax und Semantik

Modellierung
WS 17/18

Einführung

Man unterscheidet bei der Modellierung zwischen:

- Syntax: Symbole und Formen/Diagramme, die für die Darstellung des Modells genutzt werden dürfen
Im Beispiel: Bild der Ziege, blaue Fläche, etc.
- Semantik: Bedeutung, die sich jeweils dahinter verbirgt
Im Beispiel: Die blaue Fläche symbolisiert den Fluss.
Die Pfeile bedeuten: „Fluss wird überquert“. Etc.

Zu einer Syntax gibt es nicht immer eine dazugehörige Semantik.
(Im Beispiel ist die Semantik sehr vage.)

Wünschenswert ist jedoch im Allgemeinen, dass die Bedeutung aller Symbole und Formen möglichst präzise festgelegt wird.

~> Einigung auf eine gemeinsame Sprache/Notation,
oder auf gemeinsame visuelle Beschreibungen;
zur Vermeidung von Missverständnissen

Ein weiteres Beispiel: Einschreiben an der Universität

Modellierung
WS 17/18

Einführung

Szenario: Eine Reorganisation der Universität, und insbesondere des Studiensekretariats, das für Einschreibungen zuständig ist, steht an.

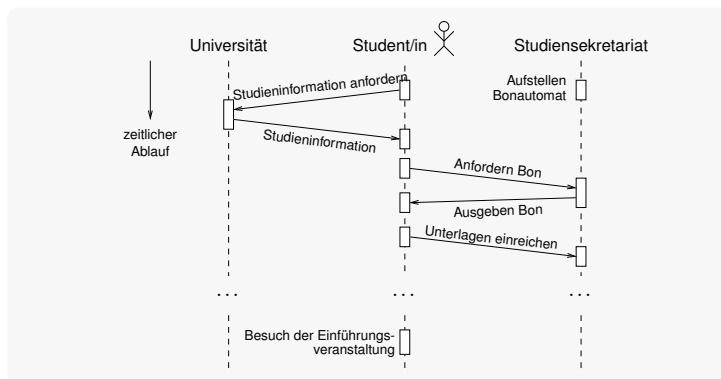
Hierzu soll der Ablauf des Einschreibens neuer Studierender modelliert werden ...

Ein weiteres Beispiel: Einschreiben an der Universität

Modellierung
WS 17/18

Einführung

Darstellung des zeitlichen Ablaufs durch Symbolisieren der beteiligten Partner als Linien und der Kommunikation durch Pfeile (Ausschnitt)



Universität

Solche Diagramme sind Bestandteil von UML.
Sie werden Sequenzdiagramme (engl. sequence diagrams, auch message sequence charts) genannt.

Einige mathematische Hilfsmittel

Mengen

Menge

Menge M von Elementen, oft beschrieben als Aufzählung

$$M = \{0, 2, 4, 6, 8, \dots\}$$

oder als Menge von Elementen mit einer bestimmten Eigenschaft

$$M = \{n \mid n \in \mathbb{N}_0 \text{ und } n \text{ gerade}\} = \{n \in \mathbb{N}_0 \mid n \text{ gerade}\}.$$

Allgemeines Format:

$$M = \{x \mid E(x)\}$$

M ist Menge aller Elemente, die die Eigenschaft E erfüllen.

$$M = \{x \in X \mid E(x)\}$$

M ist Menge aller entsprechenden Elemente aus Grundmenge X .

Bemerkungen:

- Die Elemente einer Menge sind ungeordnet, das heißt, ihre Ordnung spielt keine Rolle. Beispielsweise gilt:

$$\{1,2,3\} = \{1,3,2\} = \{2,1,3\} = \{2,3,1\} = \{3,1,2\} = \{3,2,1\}$$

- Ein Element kann nicht mehrfach in einer Menge auftreten.
Es ist entweder in der Menge, oder es ist nicht in der Menge.
Beispielsweise gilt:

$$\{1, 2, 3, 4, 4\} = \{1, 2, 3, 4\} \neq \{1, 2, 3\}$$

Mengen

Element einer Menge

Wir schreiben $a \in M$, falls ein Element a in der Menge M enthalten ist.

Anzahl der Elemente einer Menge

Für eine endliche Menge M gibt $|M|$ die Anzahl ihrer Elemente an.

Teilmengenbeziehung

Wir schreiben $A \subseteq B$, falls jedes Element von A auch in B enthalten ist.

Leere Menge

Mit $\emptyset$ oder $\{\}$ bezeichnen wir die leere Menge. Sie enthält keine Elemente und ist (echte) Teilmenge jeder anderen Menge.

Mengen

Mengenvereinigung

Die Vereinigung zweier Mengen A und B ist diejenige Menge, welche die Elemente enthält, die in A oder B (oder in beiden) vorkommen. Man schreibt dafür $A \cup B$.

$$A \cup B = \{x \mid x \in A \text{ oder } x \in B\}$$

Mengenschnitt

Der Schnitt zweier Mengen A und B ist diejenige Menge, welche die Element enthält, die sowohl in A als auch in B vorkommen. Man schreibt dafür $A \cap B$.

$$A \cap B = \{x \mid x \in A \text{ und } x \in B\}$$

Mengen

Modellierung
WS 17/18

Einführung

Mengendifferenz

Die Differenz zweier Mengen A und B ist diejenige Menge, welche die Elemente enthält, die in A vorkommen und in B nicht vorkommen. Man schreibt dafür $A \setminus B$.

$$A \setminus B = \{x \mid x \in A \text{ und } x \notin B\} = \{x \in A \mid x \notin B\}$$

Beispiele:

- $\{0, 1, 2, 3, 4, 5\} \setminus \{0\} = \{1, 2, 3, 4, 5\}$
- $\{a, b, c\} \setminus \{c, d\} = \{a, b\}$
- $\{1, 2, 3\} \setminus \emptyset = \{1, 2, 3\}$
- $\{1, 2, 3\} \setminus \mathbb{N}_0 = \emptyset$
- $(\{1, 2, 3\} \setminus \{1, 2\}) \setminus \{1\} \neq \{1, 2, 3\} \setminus (\{1, 2\} \setminus \{1\})$

Mengen

Modellierung
WS 17/18

Einführung

Kreuzprodukt (Kartesisches Produkt)

Das Kreuzprodukt zweier Mengen A und B ist diejenige Menge, welche alle Paare (a, b) enthält, wobei die erste Komponente des Paares aus A , die zweite aus B kommt. Man schreibt dafür $A \times B$.

$$A \times B = \{(a, b) \mid a \in A \text{ und } b \in B\}$$

Beispiele:

- $\{1, 2\} \times \{3, 4, 5\} = \{(1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)\}$
- $\{1, 2\} \times \emptyset = \emptyset$

Anmerkungen:

- Für endliche Mengen A und B gilt $|A \times B| = |A| \cdot |B|$.
- Wenn $A \subseteq B$, dann $A \times C \subseteq B \times C$ und $C \times A \subseteq C \times B$.

Mengen

Modellierung
WS 17/18

Einführung

Weitere Bemerkungen:

- Wir betrachten nicht nur Paare, sondern auch Tupel aus mehr als zwei Komponenten (Tripel, Quadrupel, Quintupel, ...). Ein Tupel $(a_1, \dots, a_n)$ bestehend aus n Komponenten heißt auch n -Tupel.
- In einem Tupel sind die Komponenten geordnet! Es gilt z.B.:

$$(1, 2, 3) \neq (1, 3, 2) \in \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$$

- Ein Element kann mehrfach in einem Tupel auftreten. Tupel unterschiedlicher Länge sind immer verschieden. Beispielsweise:

$$(1, 2, 3, 4) \neq (1, 2, 3, 4, 4)$$

Mengen

Modellierung
WS 17/18

Einführung

Potenzmenge

Die Potenzmenge einer Menge M ist diejenige Menge, welche alle Teilmengen von M enthält. Man schreibt dafür $\mathcal{P}(M)$.

$$\mathcal{P}(M) = \{A \mid A \subseteq M\}$$

Beispiele:

- $\mathcal{P}(\{1, 2, 3\}) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$
- $\mathcal{P}(\emptyset) = \{\emptyset\}$
- $\mathcal{P}(\mathcal{P}(\emptyset)) = \{\emptyset, \{\emptyset\}\}$

Anmerkung: Für endliche Mengen M gilt $|\mathcal{P}(M)| = 2^{|M|}$.

Mengen

Modellierung
WS 17/18

Einführung

Beispiel: Zustandsmodellierung

Angenommen, wir betrachten einen einfachen Snackautomaten für Riegel und Chips. Von jedem dieser beiden Snacks hat er maximal 30 Stück auf Vorrat. Der Automat hat eine gelbe und eine rote Warnleuchte („kein Wechselgeld mehr“ bzw. „keine Scheine mehr akzeptiert“), die unabhängig voneinander leuchten können. Die Menge der möglichen Zustände dieses Automaten können wir als

$$\mathcal{P}(\{\text{gelb, rot}\}) \times \{0, 1, \dots, 30\} \times \{0, 1, \dots, 30\}$$

beschreiben. Das Element $(\emptyset, 20, 10)$ dieser Menge zum Beispiel entspricht dem Zustand, in dem beide Warnleuchten ausgeschaltet sind und noch 20 Riegel und 10 Packungen Chips vorrätig. Wären bei diesem Vorrat die Warnleuchten beide eingeschaltet, so befände sich der Automat stattdessen im Zustand $(\{\text{gelb, rot}\}, 20, 10)$.

Funktionen

Modellierung
WS 17/18

Einführung

Funktion (Abbildung)

Funktionen bilden Elemente eines Definitionsbereiches auf Elemente eines Wertebereiches ab. Man schreibt: „ $f : A \rightarrow B$ “. Paare aus einem Element a des Definitionsbereiches A und dem (eindeutig gegebenen) Element $b = f(a)$ des Wertebereiches B , auf welches die Funktion es abbildet, notiert man in der Form „ $a \mapsto b$ “.

Die gleiche Notation verwendet man, um eine allgemeine Zuordnungsvorschrift anzugeben: „ $a \mapsto f(a)$ “.

Beispiel: Quadratfunktion auf der Menge der ganzen Zahlen

$$f : \mathbb{Z} \rightarrow \mathbb{N}_0, \quad f(z) = z^2, \quad \text{bzw. Angabe als: } z \mapsto z^2$$

Angabe konkreter Paare:

$$\dots, -3 \mapsto 9, -2 \mapsto 4, -1 \mapsto 1, 0 \mapsto 0, 1 \mapsto 1, 2 \mapsto 4, 3 \mapsto 9, \dots$$

Graphen

Modellierung
WS 17/18

Einführung

Graphen sind netzartige Strukturen, bestehend aus Knoten und Kanten. Sie bilden die Grundlagen vieler diagrammatischer Modellierungstechniken.

Gerichteter und kantenbeschrifteter Graph

Sei L eine Menge von Beschriftungen (oder Labels).

Ein gerichteter und kantenbeschrifteter Graph $G = (V, E)$ besteht aus

- einer Knotenmenge V und
- einer Kantenmenge $E \subseteq V \times L \times V$.

Bemerkung: V steht für vertices und E für edges.

Graphen

Modellierung
WS 17/18

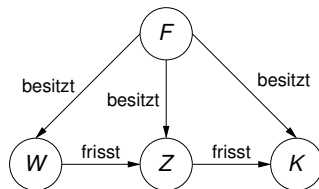
Einführung

Beispiel: (Farmer, Wolf, Ziege, Kohlkopf)

$$V = \{F, W, Z, K\} \quad L = \{\text{besitzt, frisst}\}$$

$$E = \{(F, \text{besitzt}, W), (F, \text{besitzt}, Z), (F, \text{besitzt}, K), (W, \text{frisst}, Z), (Z, \text{frisst}, K)\}$$

Bildhaft:



Graphen

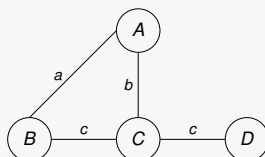
Modellierung
WS 17/18

Einführung

Weitere Arten von Graphen:

Ungerichteter Graph

Bei ungerichteten Graphen spielt die Richtung der Kanten keine Rolle.



Graphen

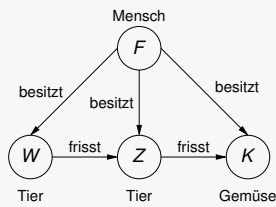
Modellierung
WS 17/18

Einführung

Weitere Arten von Graphen:

Knotenbeschrifteter Graph

Auch Knoten können Beschriftungen tragen, wobei zwei verschiedene Knoten durchaus gleich beschriftet sein dürfen.



Graphen

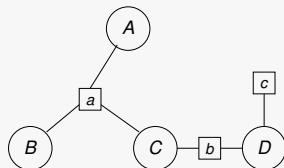
Modellierung
WS 17/18

Einführung

Weitere Arten von Graphen:

Hypergraph

Bei Hypergraphen kann eine Kante (symbolisiert durch ein Quadrat oder Rechteck) mit einer beliebigen Anzahl von Knoten verbunden sein. Eventuell sind dabei die Knoten in Bezug auf die (Hyper-)Kante geordnet (wie bei gerichteten Graphen).



Graphen

Modellierung
WS 17/18

Einführung

Graphen können in vielfältiger Weise zur Modellierung eingesetzt werden. Wir betrachten zwei typische Fälle.

Graphen zur statischen Modellierung

Knoten sind Komponenten oder Objekte, die untereinander über Kanten verbunden sind bzw. in Beziehung stehen.

Beispiel: Beziehungen zwischen Farmer, Wolf, Ziege, Kohlkopf

Zustandsübergangsdiagramme

Modellierung
WS 17/18

Einführung

Graphen zur dynamischen Modellierung

Knoten sind Zustände und Kanten sind Zustandsübergänge.

Klassischer Vertreter: Zustandsübergangsdiagramme
(auch Transitionssysteme genannt)

Zustandsübergangsdiagramm

Ein Zustandsübergangsdiagramm besteht aus einem gerichteten Graphen (Z, U) , wobei Z (= Zustände) die Knotenmenge des Graphen und U (= Übergänge) die Kantenmenge des Graphen ist, und außerdem aus einem Startzustand $z_0 \in Z$.

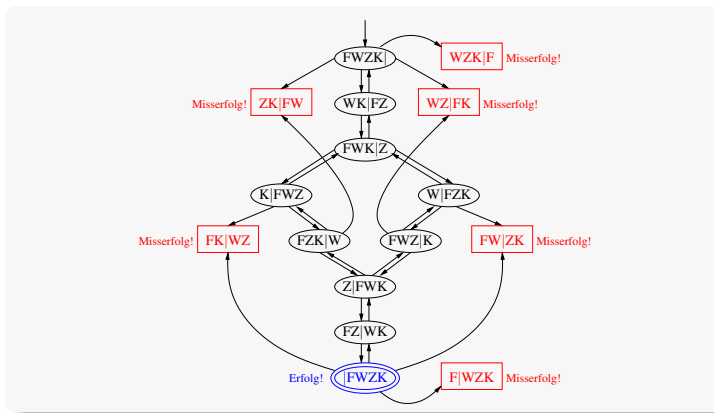
Zustandsübergangsdiagramme werden grafisch wie normale gerichtete Graphen dargestellt. Der Startzustand (auch Anfangszustand genannt) wird dabei meist durch eine eingehende Pfeilspitze gekennzeichnet.

Zustandsübergangsdiagramme

Modellierung
WS 17/18

Einführung

Beispiel: Zustandsübergangsdiagramm für das
Wolf-Ziege-Kohlkopf-Problem



Zustandsübergangsdiagramme

Modellierung
WS 17/18

Einführung

Bemerkungen:

- Der senkrechte Strich | steht hier für den Fluss. Links und rechts davon befinden sich die Akteure/Objekte (F = Farmer, W = Wolf, Z = Ziege, K = Kohlkopf). Deren Reihenfolge ist egal, und das Boot braucht man nicht anzugeben, denn es ist immer beim Farmer.
- Übergänge sind hier aus Gründen der Übersichtlichkeit nicht beschriftet. Sinnvolle Beschriftungen wären die ausgeführten Aktionen (z.B. „Farmer bringt Ziege über den Fluss“).
- Eckige (rote) Zustände symbolisieren hier Misserfolg (z.B. „Ziege frisst Kohlkopf“; solche Aktionen haben Priorität vor Überfahrten). Kanten, die aus solchen Zuständen herausführen, wurden hier weggelassen.
- Die doppelte (blaue) Ellipse symbolisiert hier Erfolg (erwünschter Zielzustand ist erreicht).

Weitere Bemerkungen:

- Es gibt mehrere (sogar unendlich viele und beliebig lange) Wege zum Zielzustand. Die zwei kürzesten enthalten jeweils sieben Übergänge.
- Zustandsübergangsdiagramme für selbst relativ einfache Systeme werden oft erstaunlich groß (sogenannte Zustandsexplosion).

Wichtig:

Zustandsübergangsdiagramme stellen im Allgemeinen alle Zustände und alle Übergänge eines Systems dar.

[illegible]