

7 Informatische Modellbildung

Wir haben in Abschnitt 6 erarbeitet, daß man die Informatik als Wissenschaft von der Herstellung ausführbarer Modelle bzw. der Simulation künstlicher Welten betrachten kann. In diesem Abschnitt analysieren wir den Modellbildungsprozeß genauer. In den weiteren Abschnitten stellen wir formale Methoden zur Präzisierung vor.

Als **Modell** bezeichnet man umgangssprachlich jedes Abbild von etwas.

Genauer ist ein Modellbildungsprozeß durch eine Relation $R(S,P,T,M)$ beschrieben, wobei ein *Subjekt* S zum *Zwecke* P (purpose) zu einem *Original* T (prototype) das *Modell* M entwirft (Abb. 1). Dabei besteht zwischen T und M stets eine *Verkürzungsrelation* in dem Sinne, daß M nicht alle, sondern nur die aus Sicht von S bezgl. P relevanten Eigenschaften von T erfaßt.

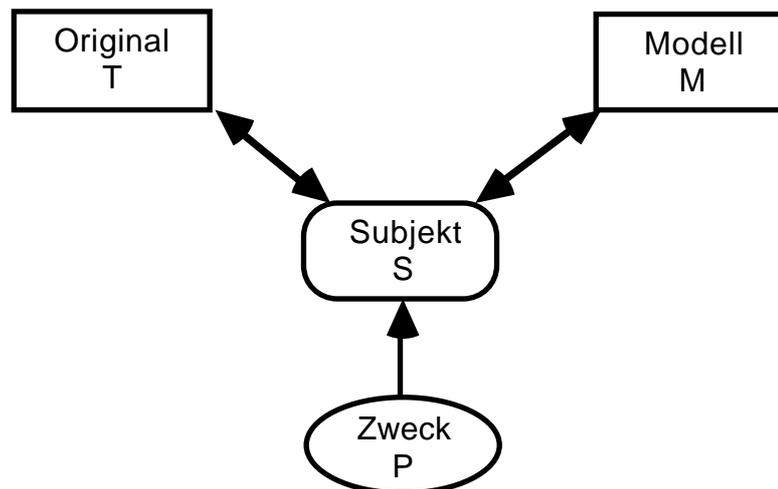


Abb. 1: Modellbildung

Die einzelnen Wissenschaften haben unterschiedliche Formen der Modellierung hervorgebracht. Die Unterschiede betreffen neben den Methoden, die S verwendet, um M aus T zu erzeugen, auch die Originale, die Zwecke der Modellbildung und die Modelle selbst. In der Logik sind Modelle mathematische Strukturen, die gegebenen Gesetzen genügen, in den Naturwissenschaften sind es formale Darstellungen, die Teile der Realität in mehr oder weniger stark vereinfachter Form beschreiben.

In der Informatik bilden Modelle die Basis für jede Form der Speicherung, Übertragung, Zerlegung und Verarbeitung. Sie verkörpern hier Wunsch- oder Zielvorstellungen (Berechnungsmodelle, Datenmodelle, Rechnermodelle), die häufig nicht oder nur in unvoll-

kommener Weise realisierbar sind. Genauer lassen sich die Elemente, durch die ein Modellierungsprozeß in der Informatik bestimmt ist, wie folgt charakterisieren:

Die Originale T.

Die Informatik modelliert meist Sachverhalte, die einer vom Menschen geschaffenen, künstlichen Welt entstammen (Bürovorgänge, Fahrzeugströme an Kreuzungen, Bibliothekssysteme). Damit fehlt ihnen eine "natürliche Einfachheit". Vielmehr können diese Originale beliebig kompliziert sein, wobei die Kompliziertheit im wesentlichen auf menschlicher Willkür beruht und daher kaum reduktionistischen Regeln unterliegt. Zugleich sind die Originale durch diese Willkür in hohem Maße diskret und ihr Verhalten hochgradig unstetig. Selbst kleinste Veränderungen der Eingabe eines bekannten Ein-/Ausgabepaares eines Programms lassen keine Rückschlüsse auf die Änderung der Ausgabe zu.

Die Zwecke P.

Die Informatik modelliert die reale durch eine künstliche Welt, die aber weitgehend realistisch bleibt und kaum idealisiert ist. Sie beschreibt ihre Originale nicht (nur auf Zwischenstufen des Modellbildungsprozesses), sondern bildet sie so nach, "wie sie sind" (z.B. Akten bleiben Akten, Karteikarten bleiben Karteikarten), besser: wie sie vom Menschen unmittelbar wahrgenommen werden. Die Modelle erlangen so eine eigene (virtuelle) Realität und könnten in jeder Hinsicht an die Stelle ihrer Originale treten.

Die Modelle M.

Das Endergebnis informatischer Tätigkeit ist meist ein maschinell ausführbares Modell, das durch einen Algorithmus bzw. ein Programm beschrieben wird. Dieses Modell besitzt typische Merkmale, von denen zwei besonders hervorzuheben sind:

- Elementarbausteine der Modelle.

Die Modelle der Informatik können aus gewissen, meist wenigen atomaren Grundbausteinen in endlich vielen Schritten konstruiert werden (vgl. die fundamentale Idee der Orthogonalisierung). Keine Informatikmodelle sind in diesem Sinne die reellen Zahlen oder stetige Funktionen (Dennoch können diese natürlich in der Informatik nützlich sein).

- Modellierung der Zeit.

In der Informatik ist die Zeit stets Teil des Modells, sie wird nicht wie in anderen Wissenschaften gequantelt oder in den Modellen "eingefroren", sondern sie "vergeht (als Eigenzeit des Modells) tatsächlich", so daß dynamische Vorgänge auch im Modell dynamisch repräsentiert sind und nicht durch statische approximiert werden müssen.

Beispiel: Der freie Fall.

Die Physik modelliert das Verhalten eines fallenden Steins statisch durch Formeln, die für jeden Zeitpunkt t Geschwindigkeit, Position und kinetische Energie des Steins angeben:

$$\begin{aligned}v(t) &= at, \\s(t) &= \frac{1}{2} at^2, \\E(t) &= \frac{1}{2} m(v(t))^2\end{aligned}$$

Die Formeln liefern für konkrete Parameter jeweils Momentaufnahmen (Zustände des Steins). Der Stein selbst wird auf eine Zahl, seine Masse, reduziert.

In der künstlichen Welt, die ein informatisches Modell beschreibt, existiert eine Eigenzeit, und der Stein fällt "tatsächlich" (virtuell). Der Stein selbst wird so modelliert, wie er in der Realität wahrgenommen wird, d.h. als Objekt, dem gewisse Eigenschaften und Operationsmöglichkeiten anhaften.

Bei Modellen unterscheidet man *ikonische* Modelle, die einen anschaulich bildhaften Bezug auf das Abgebildete (das Original) haben und *symbolische* Modelle, die mit Hilfe einer formalen Sprache wohldefinierter Syntax und Semantik beschrieben werden.

Ikonische Modelle dienen vor allem zur Veranschaulichung von Sachverhalten, erklären sie aber meist nicht, weil sie keine Gesetzmäßigkeiten oder kausalen Zusammenhänge erfassen.

Symbolische Modelle sind die allgemeinste Form von Modellen. Sie sind schwerer vorstellbar, liefern aber Erklärungen anstelle von Beschreibungen und ermöglichen daher auch Voraussagen über das zukünftige Verhalten des Originals.

Die Informatik verwendet bei der Softwareentwicklung ebenfalls ikonische (Bäume, Graphen, Struktogramme, Datenflußpläne) und symbolische Modelle (Programme, Grammatiken, formale Sprachen, logische Formeln), allerdings nur als Zwischenschritte auf dem Weg zu einem Endergebnis, das weder ikonisch noch symbolisch genannt werden kann: die künstliche Welt, die der durch ein Programm gesteuerte Prozeß generiert. Diese Welt veranschaulicht keine Sachverhalte, sie erklärt sie nicht, und sie erfaßt keine Gesetzmäßigkeiten (wohl aber das dahinterstehende Programm), sie *ist* der Sachverhalt und damit "identisch" zu ihrem Original.

Man kann hier von *enaktiven* (handlungsbezogenen) Modellen sprechen: Die Wirklichkeit wird durch Objekte modelliert, an denen man Handlungen vornehmen kann, und die selber aktiv werden und auf andere Objekte einwirken können.

Beispiel: Gegeben sei das deutsche Autobahnnetz (*Original* T). Wir suchen einen Algorithmus, der zu je zwei der folgenden Großstädte Hamburg, Berlin, Düsseldorf, Frankfurt, Nürnberg, Stuttgart, München, zwischen denen man häufig pendelt, den kürzesten Weg

und seine Länge in Kilometern ausgibt (Zweck P). Zum Beispiel wäre der kürzeste Weg von Düsseldorf nach München der Weg D-F-N-M mit 610 km. Abb. 1 zeigt ein ikonisches Modell dieser Aufgabenstellung (ein sog. *Graph*), Abb. 2 ein symbolisches. In Abb. 2 ist das Autobahnnetz ein Tripel $A=(X,Y,d)$. X ist die Menge der Städte, Y die Menge aller 1- und 2-elementigen Teilmengen von X, die Direktverbindungen, und $d: Y \rightarrow \mathbb{N}_0$ die Entfernungsfunktion.

Offenbar besteht zwischen Original und Modell eine (erhebliche) Verkürzungsrelation, denn im Original vorkommende Elemente, wie die geographische Lage der Städte, Staus, Baustellen, Geschwindigkeitsbegrenzungen, Fahrbahnzustand, Wetter, Städte unterwegs und vieles mehr sind im Modell nicht erfaßt. Das ist für den vorgegebenen Zweck "kürzeste Wege finden" aber auch nicht erforderlich.

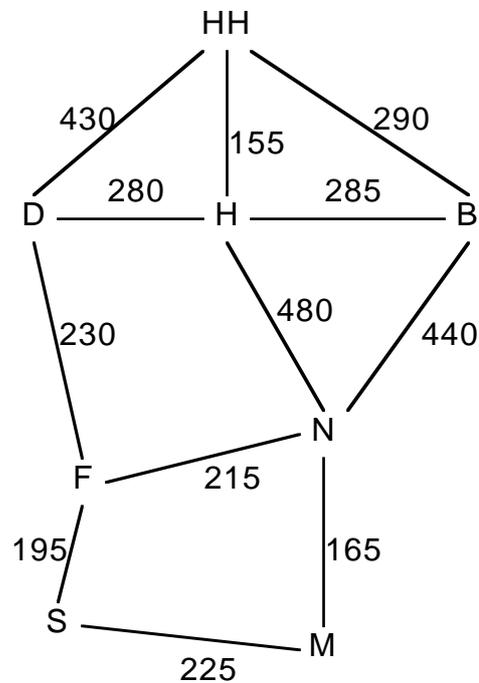


Abb. 2: Ikonisches Modell

$A=(X,Y,d)$, wobei
 $X=\{HH,D,H,B,F,N,S,M\}$,
 $Y\subseteq 2^X$ mit
 $Y=\{\{HH\},\{D\},\{H\},\{B\},\{F\},\{N\},\{S\},\{M\},\{HH,D\},\{HH,H\},\{HH,B\},\{H,D\},\{H,N\},\{H,B\},$
 $\{D,F\},\{B,N\},\{F,N\},\{F,S\},\{N,M\},\{S,M\}\}$ und
 $d: Y\rightarrow\mathbb{N}_0$ mit
 $d(\{HH,D\})=430, d(\{HH,H\})=155, d(\{HH,B\})=290, d(\{H,D\})=280,$
 $d(\{H,N\})=480, d(\{H,B\})=285, d(\{D,F\})=230, d(\{B,N\})=440, d(\{F,N\})=215,$
 $d(\{F,S\})=195, d(\{N,M\})=165, d(\{S,M\})=225,$
 $d(z)=0$ für alle übrigen $z\in Y$.

Abb. 3: Symbolisches Modell

Nach diesen Überlegungen zum Wesen der (informatischen) Modellbildung müssen wir nun die einzelnen Aspekte präzisieren. Offenbar müssen die Präzisierungen vor allem beim Zweck der Modellbildung und beim Modell selbst ansetzen, an den Originalen und an den Subjekten können wir ja nicht viel ändern.

Das Ziel informatischer Modellbildung ist zumeist die Lösung eines Problems. Den *Zweck* der Modellbildung formalisieren wir folglich am besten, indem wir exakt durch eine sog. *Spezifikation* festlegen, welches Problem unsere algorithmische Lösung erfassen soll (Kapitel 8).

Das *Modell* präzisieren wir einerseits durch die *Datenobjekte*, mit denen wir die Realität nachbilden; dazu geben wir Vorschriften zur Konstruktion von Datentypen und Datenstrukturen an (Kapitel 9). Andererseits formalisieren wir die Sprachmittel, mit denen die *Verarbeitung* der Datenobjekte beschrieben wird (Kapitel 10). Bei dieser Beschreibung unterscheiden wir – wie bereits in Abschnitt 4.5 erläutert – drei verschiedene Darstellungsformen: die *imperative*, die *funktionale* und die *prädikative* Form. Die imperative Form kennen wir bereits via PRO recht genau, die beiden anderen Formen lernen wir im weiteren Verlauf der Vorlesung kennen. Die funktionale Form behandeln wir dabei vertieft; zugleich werden wir in dieser Sprache auch die ersten Erfahrungen mit einer konkreten Programmiersprache, der Sprache ML, sammeln.