

GRUNDBEGRIFFE DER INFORMATIK

Kurzbeschreibung

Die Kurseinheit führt in zentrale Grundbegriffe der Informatik (Information, Algorithmus, Modell, Architektur) ein und vertieft diese anhand konkreter Beispiele.

Schlüsselwörter

Information, Signal, Wissen, Semiotik, Algorithmus, Euklidischer Algorithmus, Modell, Wirklichkeit, objektorientierte Modellierung, Rechnerentwicklung, von-Neumann-Architektur, Software-Architektur

Lernziele

1. Der Zusammenhang des Begriffs der Information zu den Begriffen Signal, Nachricht, Syntax, Semantik, Wissen und Codierung ist bekannt.
2. Die wesentlichen Eigenschaften eines Algorithmus werden verstanden.
3. Das Modell als Abbildung eines Ausschnitts der Wirklichkeit und die Bedeutung des Modellierens können nachvollzogen werden.
4. Die den Systemen zugrunde liegenden Architekturprinzipien können wiedergegeben werden.

Hauptquellen

- Gerhard Goos: Vorlesungen über Informatik, Band 1: Grundlagen und funktionales Programmieren, Springer Verlag 1995.

Inhaltsverzeichnis

1	INFORMATION.....	2
1.1	Signal.....	2
1.2	Syntax und Semantik.....	4
1.3	Codierung und Shannonsche Informationstheorie.....	6
2	ALGORITHMUS.....	11
2.1	Euklidischer Algorithmus.....	12
2.2	Effizienz von Algorithmen.....	14
3	MODELL.....	15
3.1	Ziele von Modellen.....	16
3.2	Modell-Beispiel.....	18
4	ARCHITEKTUR.....	23
4.1	Rechensysteme.....	24
4.2	Verteilte Systeme und Verteilte Anwendungen.....	27
	VERZEICHNISSE.....	31
	Abkürzungen und Glossar.....	31
	Index.....	32
	Informationen und Interaktionen.....	32
	Literatur.....	33

- INFORMATION
 - Signal, Information, Wissen, Syntax, Semantik, Codierung
- ALGORITHMUS
 - Definition, Ablaufelemente, Beispiel, Effizienz
- MODELL
 - Wirklichkeit und Modell, objektorientierte Modellierung
- ARCHITEKTUR
 - von-Neumann-Rechner, Software-Architektur

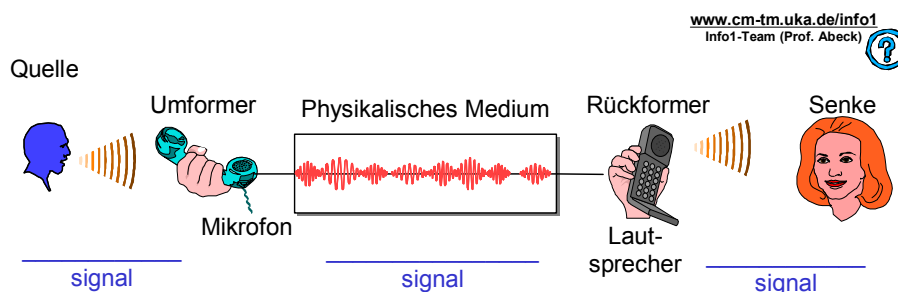
Information 1: GRUNDBEGRIFFE DER INFORMATIK

1 INFORMATION

Das Kunstwort Informatik ist durch den Begriff der Information (sowie durch den Begriff der Mathematik) geprägt. In diesem Kapitel werden die wichtigsten Begriffe und Sachverhalte, die mit der Information in Zusammenhang stehen, skizziert [Go95].

1.1 Signal

Der Ausgangspunkt der Überlegungen, die zur Information führen, ist das Signal [C&M-BUS].



- Signal als Ausgangspunkt für den Informationsbegriff
 - Signalparameter sind die Eigenschaften des Signals, die der zeitlichen Veränderung unterliegen
 - Signalübertragung ist die Weitergabe von Mitteilungen auf der Ebene der Signale
- Inschrift oder schriftliche Darstellung ist die dauerhafte Darstellung einer Mitteilung auf einem physikalischen Medium

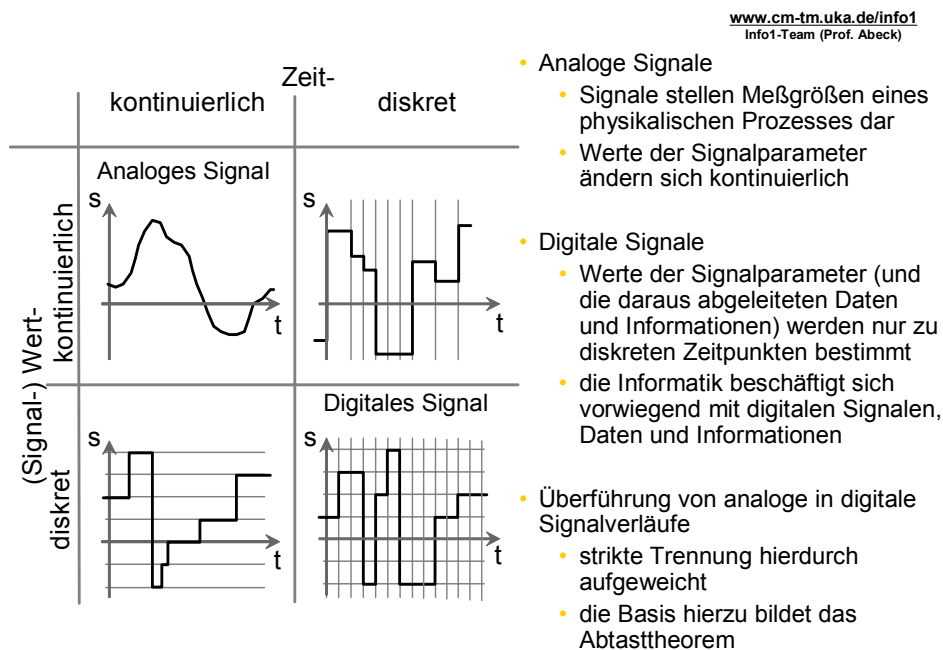
Interaktion 1: INFORMATION – Signal und Inschrift

Eine gesprochene Mitteilung wird in Form von Schallwellen ausgedrückt. Falls mit dem Empfänger, der die Mitteilung erhalten hat, über das Telefon kommuniziert wurde, wie in der Abbildung in Interaktion 1 dargestellt ist, hat eine zusätzliche Transformation in elektromagnetische Impulse oder Lichtwellen stattgefunden, die beim Empfänger wieder umgesetzt wurden und als Schallwellen aus dem Hörer kommen. In diesem Fall wird ein

Übertragungssystem bestehend aus einem Umformer und einem Rückformer genutzt, die eine Transformation vom Primärsignal in das Übertragungssignal und wieder zurück durchführen. Die beiden Signaltypen, die im Zusammenhang mit dem Umformer und dem Rückformer unterschieden werden, sind in Interaktion 1 zu ergänzen.

Häufig besteht die Anforderung, eine mittels Signalen gebildete Mitteilung dauerhaft (persistent) zu speichern, woraus die so genannte Inschrift resultiert. Die Medien zur Speicherung von Inschriften werden als Schriftmedien bezeichnet.

Eine nähere Analyse der Signale führt zu den in Information 2 [C&M-BUS] angegebenen Signalklassen.



Information 2: Signalklassen

Die analogen Signale resultieren aus den zeitlich sich kontinuierlich ändernden physikalischen Prozessen. Ein Beispiel eines physikalischen Prozesses ist die Beschleunigung eines Fahrzeugs, bei dem die Geschwindigkeit kontinuierlich mit der Zeit zunimmt. Digitale Signale entstehen dadurch, dass die Werte der Signalparameter nur zu diskreten Zeitpunkten bestimmt werden.

Die Signalklassen werden durch den Signalverlauf bestimmt, d.h. durch die Änderung des Signalwerts s über der Zeit t . Dabei gilt sowohl für den Signalwert als auch für den Zeitverlauf die Unterscheidung von kontinuierlich (stetig) und diskret (sprunghafter Verlauf beschreibt, bei dem nur bestimmte Werte gültig sind). Diese Unterscheidung ergibt die vier in Information 2 angegebenen Signalklassen.

Es stellt sich die Frage, ob sich analoge und digitale Signale geeignet ineinander überführen lassen. Die Vermutung ist zunächst, dass durch die Wahl der diskreten Zeitpunkte beim Übergang „von analog zu digital“ Information verloren geht. Falls diese diskreten Zeitpunkte genügend kurz nacheinander gewählt werden, entsteht kein Informationsverlust. Diese Aussage wird durch das Abtasttheorem präzisiert.

Im Bereich der Telekommunikation wurden Verfahren entwickelt, um ein Signal einer bestimmten Klasse in ein Signal einer anderen Klasse zu wandeln. Besondere Bedeutung hat

hierbei die Wandlung vom kontinuierlichen zum diskreten Fall, die so genannte Digitalisierung. Dieser Vorgang wandelt zeit- und wertkontinuierliche Signale in digitale Signale um. Das Verfahren ermöglicht beispielsweise die Speicherung von Sprache und Musik auf einer CD.

1.2 Syntax und Semantik

Signale zur Darstellung von Mitteilungen, Inschriften zur Speicherung und die Verarbeitung von Signalen bzw. Inschriften werden allgemein als (technische) Gegenstände bezeichnet [Go95].


www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck)

- Nachricht
 - Mitteilung, bei der vom verwendeten Medium (Signale und Inschriften) und den Einzelheiten der Signale und Signalparameter abstrahiert wird
 - die Informatik konzentriert sich auf gewisse Eigenschaften von Signalen, Inschriften und deren Verarbeitung
- Information
 - ist die einer Nachricht zugeordnete Bedeutung
 - für die Zuordnung von Bedeutung wird ein Bezugssystem benötigt
 - wird gewonnen durch die Interpretation von Nachrichten auf der Grundlage eines Bezugssystems

Information 3: Nachricht und Information

Diese Gegenstände sind für die Informatik bedeutsam, allerdings reichen gewisse gemeinsame Eigenschaften der Gegenstände aus, um die Bedeutung zu erfassen. Die Einschränkung auf bestimmte Gegenstände heißt Abstraktion.

Der Übergang von einer Nachricht zu einer Information ist eine wesentliche Frage, die sich die Informatik stellt. Hierdurch wird den die Nachricht bildenden Symbolen eine Bedeutung zugewiesen. Es wird der Übergang von der Syntax zur Semantik hergestellt.

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck) 

- Nachricht - α -> Information
 - α ist die Interpretationsvorschrift
 - Interpretationsvorschriften sind selbst Informationen, die durch Nachrichten repräsentiert werden
 - Der Satz "Person P versteht die Sprache S" ist mittels der Interpretationsvorschrift zu präzisieren (Hinweis: S ist als eine Menge von Nachrichten aufzufassen)
-
- Das Paar (Nachricht, zugeordnete Information) wird als Datum bezeichnet
 - Für dieselbe Nachricht können unterschiedliche Interpretationsvorschriften bestehen
 - Beispiel 1: Position einer Person _____
 - Beispiel 2: x^2 _____

Interaktion 2: Interpretationsvorschrift α

Eine Interpretationsvorschrift α erhält als Eingabe die Nachricht und liefert als Ausgabe die Information. Interpretationsvorschriften sind ihrerseits ebenfalls Informationen, die durch Nachrichten repräsentiert werden.

In Interaktion 2 soll der Zusammenhang zwischen der Interpretationsvorschrift und dem Satz Person P versteht Sprache S beschrieben werden.

Das durch die Interpretationsvorschrift erzeugte Paar wird auch als Datum bezeichnet.

Anhand der in Interaktion 2 angegebenen Beispiele soll aufgezeigt werden, dass die Interpretation einer Nachricht, also die Zuordnung einer Bedeutung, nicht immer eindeutig ist. Eine besondere Art von Information ist das Wissen.

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck)

- Wissen ist die Information bzw. Kenntnis, wie Nachrichten zu interpretieren sind
- Zwei Arten von Wissen
 1. Faktenwissen
 - unmittelbare Kenntnis der durch ein Datum gegebenen Information
 2. Synthetisches oder prozedurales Wissen
 - Kenntnis von Interpretationsvorschriften zur Erzeugung solcher Information
- Weitere Eigenschaften von Wissen
 - statisch - dynamisch
 - scharf - unscharf - ungenau
 - detailliert - grob

Information 4: Wissen

Information 4 gibt eine Begriffserklärung von Wissen und beschreibt die zwei Arten von Wissen. Wissen kann sich unmittelbar aus den Daten ergeben (Faktenwissen) oder es wird entsprechende Kenntnis zur Erzeugung von Wissen (prozedurales Wissen) benötigt. Wissen selbst muss in Form von geeigneten Daten dargestellt werden. Das bedeutet, dass die Wissensrepräsentation geeignete Datenstrukturen zur Darstellung von Wissen erfordert.

Gemäß folgender Kriterien lässt sich Wissen klassifizieren:

- **Kriterium Zeit**

Der Unterschied zwischen statischem und dynamischem Wissen besteht in der Bedeutung, die die Zeit spielt: Dynamisches Wissen beschreibt im Gegensatz zu statischem Wissen zeitlich veränderliche Gegenstände und Sachverhalte zu den jeweiligen Zeitpunkten.
- **Kriterien Genauigkeit und Granularität**

Zwischen diesen Kriterien besteht folgender Zusammenhang: Häufig kann ein Sachverhalt, wie z.B. die Länge eines Holzstücks, aufgrund der Messgenauigkeit nicht genau, also exakt, angegeben werden. Man muss sich auf eine gewisse Granularität der Angabe einschränken – in diesem Fall die Maßgenauigkeit der Messeinrichtung. Man spricht dann von einem unscharfen Wissen.

- Syntax oder syntaktische Struktur von Daten
 - Beziehungen, die sich unmittelbar aus der Anordnung der Zeichen ergeben
- Semantik
 - Weitergehende Beziehungen, die sich aus der Interpretationsvorschrift ergeben
- Pragmatik oder pragmatische Bedeutung
 - Wissen, das sich ergibt, indem man Daten zu Gegenständen oder Sachverhalten außerhalb der vorgegebenen Datenmenge in Beziehung setzt
- Semiotik ist die Lehre von den Zeichen und setzt sich aus den drei obigen Teilen zusammen

Information 5: Semiotik

Ziel der Interpretation ist es, Wissen über Gegenstände innerhalb, aber insbesondere auch außerhalb der Informatik zu gewinnen und zu formulieren. Eine Interpretationsvorschrift gilt als konsistent, wenn sich keine Einsichten aus den Daten herauslesen lassen, die dem darzustellenden Wissen widersprechen.

Falls die Daten sich nicht zur Darstellung eines bestimmten Wissens eignen, handelt es sich um eine so genannte inhärente Interpretationsvorschrift. Man sagt dann auch, die Daten "sind kein Modell" für das Wissen. Die Frage, ob Daten ein Modell für irgendwelches Wissen bilden, ist ausschließlich auf syntaktischer Ebene zu beantworten, da nur die Syntax der Daten beobachtbar und messbar ist.

Die Syntax ist der erste Teil der als Semiotik bezeichneten Lehre der Zeichen, deren weiteren zwei Teile – die Semantik und die Pragmatik – in Information 5 beschrieben sind.

1.3 Codierung und Shannonsche Informationstheorie

Die Frage, wie sich Informationen in Form von Zeichen als Nachrichten darstellen lassen, wird durch die in Interaktion 3 behandelte Codierung beantwortet [Go95].

- Codierung ist die Darstellung einer Information in Form von Zeichen
 - der für die Codierung verwendete Zeichenvorrat heißt Code
- Beispiel: Codierung der Information zwei
 1. _____ Code: _____
 2. _____ Code: _____
 3. _____ Code: _____
- Redundante Codierung ermöglicht das Erkennen von fehlender oder verfälschter Information aufgrund der bestehenden codierten Information
- Binärcode
 - Zeichenvorrat, der aus zwei Werten, den Binärzeichen besteht
 - Beispiel: $B = \{0, 1\}$
 - beliebige Zahlwerte hiermit codierbar

Interaktion 3: Codierung von Informationen

Eine Codierung basiert immer auf einem Zeichenvorrat, dem Code. In dem in Interaktion 3 nachgefragten Beispiel wird der Code aus Folgen lateinischer Buchstaben oder aus einer Dezimalziffer oder aus Strichen gebildet.

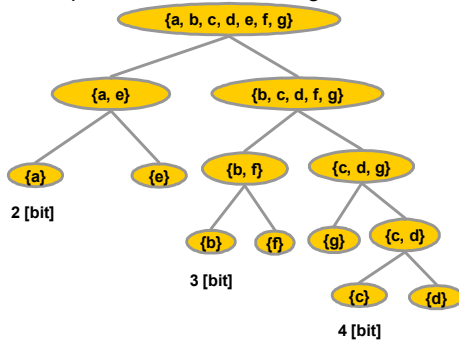
Ein Beispiel für einen redundanten Code ist die Verkehrsampel: Bei Ausfall eines der Lichter lässt sich das Signal aufgrund der zwei noch funktionierenden Lichter rekonstruieren. In der Telekommunikation erreicht man einen redundanten Code dadurch, dass zu der Codierung noch geeignete Prüfinformation hinzugefügt wird.

Eine Antwort, wie eine "optimale" Codierung von Zeichen eines vorgegebenen Zeichenvorrats aussieht, liefert die Shannonschen Informationstheorie [BG73].



- Kern der Theorie: Bei Eintreffen eines von einer Shannonschen Quelle (Nachrichtenquelle) gesendeten Zeichen ist zu entscheiden, welches Zeichen eines im voraus bekannten Zeichenvorrats vorliegt
 - das Ziel ist, für häufig auftretende Zeichen die Anzahl der dafür aufzuwendenden Entscheidungen möglichst klein zu halten
 - das Ziel ist durch geeignete Zerlegung in "gleichwahrscheinliche" Teilmengen zu erreichen

Beispiel einer Entscheidungskaskade



- p_i sei die Wahrscheinlichkeit, dass das i -te Zeichen auftritt (a ist 1. Zeichen, b ist 2. Zeichen,..., g ist 7. Zeichen)

Wie groß sind die Wahrscheinlichkeiten für die im Beispiel auftretenden Zeichen unter der Annahme, dass eine exakte Gleichheit der Teilmengen gegeben ist?

$p_1 =$ _____ $p_2 =$ _____
 $p_3 =$ _____ $p_4 =$ _____
 $p_5 =$ _____ $p_6 =$ _____
 $p_7 =$ _____

Interaktion 4: Shannonsche Informationstheorie

Von C. Shannon wurde im Jahr 1948 eine Theorie entwickelt, durch die der Entscheidungsgehalt eines Zeichens mathematisch gefasst werden kann.

Der Theorie, dessen Kern in Interaktion 4 beschrieben ist, geht dabei von einer elementaren Entscheidung zwischen zwei Zeichen aus. Jede solche Alternativentscheidung wird in der Einheit [bit] gemessen. Der Zusammenhang, der zwischen dieser Maßeinheit [bit] (mit kleinem b) und dem gerade eingeführten Bit (mit großem B) wird weiter unten geklärt.

Anhand der für ein einfaches Beispiel aufgestellten Entscheidungskaskade wird das Vorgehen, das der Shannonschen Informationstheorie zugrunde liegt, deutlich: Ist ein Zeichen zu bestimmen, so steigt man die Entscheidungskaskade schrittweise durch die Alternativentscheidung ab, ob das Zeichen in der einen oder anderen Teilmenge liegt. Der Entscheidungsgehalt des Zeichens ist durch die Anzahl der erforderlichen Alternativentscheidungen bestimmt.

Das Ziel der Zerlegung, die der Entscheidungskaskade zugrunde liegt, besteht darin, dass die Summen der Wahrscheinlichkeiten der in den Teilmengen enthaltenen Zeichen für die zwei Teilmengen möglichst genau gleich sind. Im Beispiel wird von der (in realen Zeichenvorräten nicht gültigen) Annahme ausgegangen, dass eine exakte Gleichheit erzielt werden kann. Aus der Entscheidungskaskade lassen sich die Wahrscheinlichkeiten der Zeichen unter dieser Annahme einfach ermitteln (siehe Interaktion 4).



- Zusammenhang zwischen Wahrscheinlichkeit des Auftretens des i-ten Zeichens p_i und der für dieses Zeichen benötigten Alternativentscheidungen-Anzahl k_i

$$p_i = \left(\frac{1}{2}\right)^{k_i}$$

- Daraus ergibt sich der Entscheidungsgehalt eines Zeichens

$$k_i = \text{_____ [bit]}$$

- Entscheidungsgehalt H eines Zeichenvorrats
 - $H = \sum p_i \cdot \text{ld}(1/p_i)$ [bit]
 - ist der mittlere Entscheidungsgehalt pro beliebig herausgegriffenem Zeichen
- Entscheidungsgehalt eines Zeichenvorrats mit n Zeichen
 - für $n = 2$ Zeichen gilt $H = \text{_____}$
 - für $n > 2$ Zeichen gilt $H \leq \text{_____}$

Interaktion 5: Entscheidungsgehalt

Unter der Annahme der exakten Gleichheit besteht offensichtlich der in Interaktion 5 dargestellte Zusammenhang zwischen der Wahrscheinlichkeit des Auftretens eines Zeichens und der Anzahl an Alternativentscheidungen, die zu dessen Auswahl erforderlich sind. Diese Gleichung lässt sich einfach so umformen, dass eine Formel für die Anzahl der Alternativentscheidungen gewonnen werden kann.

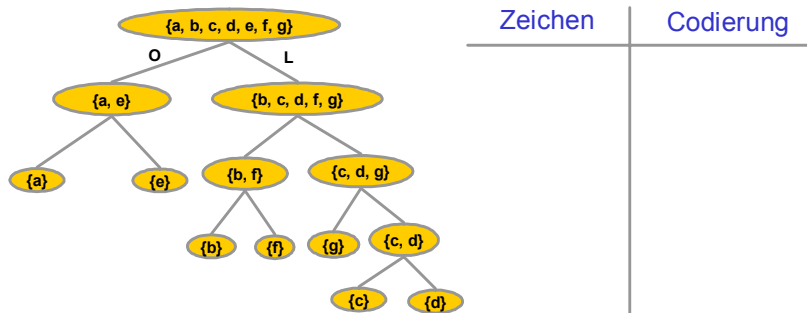
Diese Anzahl wird der Entscheidungsgehalt eines Zeichens genannt. Die Messeinheit ist die eingeführte Einheit [bit]. Der Entscheidungsgehalt eines Zeichens bildet die Grundlage zur Ermittlung des Entscheidungsgehalts des Zeichenvorrats, in dem dieses Zeichen enthalten ist, wie den Ausführungen in Interaktion 5 entnommen werden kann.

Die Größe H und deren Bestimmung bilden die Basis der Shannonschen Informationstheorie. H wird als mittlerer Entscheidungsgehalt pro Zeichen oder als Information pro Zeichen bezeichnet.



- Das Ergebnis einer einzelnen Alternativentscheidung kann durch O bzw. L wiedergegeben werden
 - jedem Zeichen wird dadurch ein Binärwort zugeordnet, durch das das Zeichen codiert wird

Aus der Entscheidungskaskade resultierende Zeichencodierung



Interaktion 6: Codierung von Zeichen

Im Zusammenhang mit der Codierung erhält die Größe H eine weitere Bedeutung, nämlich die mittlere Wortlänge des Codes, der in der Einheit Bit gemessen wird (siehe Interaktion 6). Hiermit ist auch der Zusammenhang zwischen [bit] und Bit hergestellt. Durch die Codierung werden die Zeichen des Zeichenvorrats in eine Reihenfolge gebracht, was zum Begriff des Alphabets führt.

- Ein Zeichenvorrat, in dem eine Reihenfolge (lineare Ordnung) für die Zeichen definiert ist, heißt ein Alphabet
- Beispiele für Alphabete
 - Alphabet der Dezimalziffern {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
 - Alphabet der lateinischen Buchstaben
- Die wichtigsten für die Zeichenverarbeitung benötigten Alphabete sind Bestandteil des ASCII-Zeichenvorrats
 - 7-Bit-Codierung
 - wird erweitert durch den Unicode (16-Bit-Codierung)
 - auf den ASCII-Code und Unicode wird im Zusammenhang mit der Behandlung der Sprache Java ausführlich eingegangen

Information 6: Alphabet und Codes

Mithilfe eines Alphabets können Zeichenreihen aufsteigend oder absteigend lexikographisch angeordnet werden; diese Anordnung wird auf die Anordnung der einzelnen Zeichen zurückgeführt.

Allerdings werden eventuell noch gewisse Zusatzinformationen benötigt. So muss beispielsweise beim lateinischen Alphabet der Umgang mit den Umlauten geklärt werden. Die Anordnung nach DIN 5007 sagt aus, dass die Umlaute ä, ö, ü wie a, o, u zu behandeln sind.

Diese Regelung gilt beispielsweise nicht bei der Ordnung von Namen im Telefonbuch, in dem Umlaute wie Ligaturen ae, oe, ue behandelt werden.

Ein bekannter, in der Informatik weit verbreiteter Code ist der ASCII-Code. ASCII steht für *American Standard Code for Information Interchange* und ist eine Ausprägung des ISO-7-Bit-Codes.

Auf den ASCII-Code und dessen Erweiterung zum so genannten Unicode wird in einer späteren Kurseinheit PROGRAMMIERGRUNDLAGEN [C&M-PG] im Zusammenhang mit der Programmiersprache Java und dem darin bereitgestellten Zeichentyp (char) näher eingegangen.

2 ALGORITHMUS

Ein weiterer zentraler Begriff der Informatik ist der Algorithmus. Es besteht ein enger Zusammenhang zwischen Algorithmen und der im letzten Kapitel behandelten Information [Go95].

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck)

- Ziel der Informationsverarbeitung ist die Verknüpfung von Informationen unterschiedlicher Herkunft zur Gewinnung neuer Informationen
- Zwei verschiedene Formen der Verknüpfung lassen sich unterscheiden
 1. transformierende (selektierende) Informationsverarbeitung
 - die verknüpften Informationen gehen in das Ergebnis auf und lassen sich nicht oder nur noch teilweise zurückgewinnen
 2. strukturierende (relationale) Informationsverarbeitung
 - zwischen den Eingangsdaten werden Beziehungen hergestellt, die Bestandteil des Ergebnisses sind
- Aus technischer Sicht werden der Informationsverarbeitung Eingangsdaten zugeführt
 - Ergebnis der Transformation bzw. Strukturierung liefern die Ausgangsdaten
 - Informationsverarbeitung entspricht einer Funktionsberechnung

Information 7: ALGORITHMUS - Informationsverarbeitung

In Information 7 werden zwei Formen der Informationsverarbeitung unterschieden. Beispiele für eine transformierende Informationsverarbeitung ist die Durchführung einer Addition oder die Bestimmung der n-ten Primzahl. Eine strukturierende Informationsverarbeitung liegt beispielsweise vor, wenn Wörter (bestehende Information) zu einem Satz (neue Information) zusammengefügt werden.

Die Verarbeitung von Informationen setzt entsprechende Eingangsdaten bzw. die sie repräsentierenden Signale voraus. Die durch Transformation oder Strukturieren entstehende Information schlägt sich in entsprechenden Ausgangsdaten nieder.

Die Informationsverarbeitung kann als eine Funktionsberechnung aufgefasst werden, die die Eingangsdaten in Ausgangsdaten überführt. Eine Funktionsberechnung kann auf drei verschiedenen Arten erfolgen, was in Information 8 zum Begriff des Algorithmus führt.

- statisch (deklarativ)
 - Benutzung von Gleichungen
 - kann zu einem konstruktiven Verfahren weiterentwickelt werden
- tabellarisch
 - Aufstellen einer Tabelle, in der für alle Werte a des Definitionsbereichs A der Wert $f(a)$ im Wertebereich B angegeben ist
 - nicht anwendbar für unendliche bzw. zu große Definitionsbereiche
- algorithmisch (operativ, prozedural, synthetisch)
 - Beschreibung zur Berechnung des Ergebnisses für beliebige Argumente a des Definitionsbereichs A
 - die in der Beschreibung verwendeten elementaren Operationen müssen hinreichend präzise spezifiziert sein

Information 8: Arten der Berechnung einer Funktion $f:A \rightarrow B$

statisch oder deklarativ: Die Zusammenhänge zwischen dem Definitionsbereich A und dem Wertebereich B werden statisch durch Gleichungen angegeben. Im ersten Moment erscheint es, dass man hierdurch kein konstruktives Verfahren erhält, sondern nur die Möglichkeit nachzuprüfen, ob ein Resultat richtig ist. Im Zusammenhang mit der funktionalen Programmierung wird verdeutlicht, dass aus einer solchen deklarativen Beschreibung häufig doch eine konstruktive Berechnungsmethode hergeleitet werden kann.

tabellarisch: hierbei handelt es sich um die direkteste Form der Angabe einer Funktion, die aber nur in seltenen und besonders einfachen Fällen zur Anwendung kommen kann. Insbesondere muss erfüllt sein, dass der Definitionsbereich A nicht allzu groß ist. Der Berechnungsvorgang besteht darin, in der Tabelle das gewünschte Element aus dem Definitionsbereich zu suchen und den entsprechenden Tabelleneintrag des Wertebereichs auszugeben.

algorithmisch: Folge von endlich vielen elementaren Operationen, durch die das Vorgehen zur Ermittlung des gewünschten Funktionsergebnisses hinreichend präzise beschrieben ist. Diese Anforderungen spiegeln sich im Begriff des Algorithmus sehr gut wider. Algorithmen haben in der Informatik eine große Bedeutung, da sie die Grundlage jedes Informatiksystems bilden.

2.1 Euklidischer Algorithmus

Ein berühmtes Beispiel eines solchen mechanisch ausführbaren Rechenverfahrens ist der so genannte Euklidische Algorithmus. Seinen Namen hat dieser Algorithmus von Euklid, einem griechischen Mathematiker (um 300 v. Chr.). Der Euklidische Algorithmus ist der älteste und damit erste Algorithmus überhaupt. Ein Vergleich mit der Jahreszahl von Al Khwarizmi (um 800 n. Chr.), nach dem der Begriff des Algorithmus benannt ist, macht deutlich, dass der Begriff Algorithmus zu dieser Zeit noch gar nicht bekannt war.



- Aufgabe: Bestimmung des größten gemeinsamen Teilers (ggT) aus zwei gegebenen natürlichen Zahlen p und q
- Die zwei Schritte des Euklidischen Algorithmus
 1. Dividiere p durch q . Dabei erhält man einen Rest r , der zwischen 0 und $q-1$ liegt.
 2. Wenn $r=0$ ist, dann ist q der gesuchte größte gemeinsame Teiler.
Wenn $r < > 0$, dann benenne man das bisherige q in p sowie das bisherige r in q um und wiederhole Schritt 1 und Schritt 2 so lange, bis $r=0$ geworden ist.
- Beispiel: Der ggT(24, 18) ist gemäß dem angegebenen Verfahren zu berechnen

Interaktion 7: Euklidischer Algorithmus

Wie in Interaktion 7 beschrieben wird, liefert der Euklidische Algorithmus zu zwei natürlichen Zahlen den größten gemeinsamen Teiler.

Der Euklidische Algorithmus berechnet den ggT in zwei, ggf. mehrfach ausgeführten Schritten. Zunächst wird p durch q dividiert, wobei es sich hier um eine ganzzahlige Division handelt. Das bedeutet, dass die Division keine Nachkomma-Stellen berechnet, sondern einen Rest erzeugt, falls die Division nicht aufgeht. Schritt 2 beginnt mit der Endbedingung des Algorithmus: Man ist fertig, wenn die ganzzahlige Division von Schritt 1 aufgegangen ist, d.h. keinen Rest ergeben hat. Andernfalls startet man das Verfahren von Neuem (allerdings mit den im letzten Durchlauf ermittelten Werten).

Das Beispiel macht deutlich, dass der Ablauf des Euklidischen Algorithmus durch die in den Berechnungsgrößen (hier r , p , q) enthaltenen Werte abhängt.



- Die Ablaufsteuerung legt die Reihenfolge und Ausführungsbedingungen von Tätigkeiten (elementaren Operationen) in einem Algorithmus fest
- Ablaufsteuerungselemente
 - sequentielle Ausführung
 - parallele bzw. kollaterale Ausführung
 - bedingte Ausführung
 - Ausführung in einer Schleife
 - Ausführung eines Unterprogramms
- Hiermit können beliebige Abläufe in algorithmischer Form beschrieben werden
- Welche Ablaufsteuerungselemente wurden im Euklidischen Algorithmus genutzt?

Interaktion 8: Ablaufsteuerung

Es werden die in Interaktion 8 genannten Elemente der Ablaufsteuerung unterschieden. Zu jedem der aufgeführten Ablaufsteuerungselemente werden in den nachfolgenden Kurseinheiten entsprechende Anweisungen der Programmiersprache Java eingeführt, die zur Formulierung von Algorithmen dienen, die auf einem Rechner ausgeführt werden können. Man spricht dann von Programmen.

2.2 Effizienz von Algorithmen

Bei einem Algorithmus stellt sich grundsätzlich die in Information 9 angesprochene Frage nach seiner Effizienz [BB93].

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck)

- Mit der Effizienz ist unmittelbar die Frage verbunden, welcher von mehreren vorgegebenen Algorithmen zur Lösung eines Problems vorzuziehen ist
- Zwei unterschiedlich Vorgehensweisen zur Beantwortung der Frage
 - (1) Empirisches (a posteriori) Vorgehen
Programmieren der Algorithmen und Erprobung mit verschiedenen Fällen (Eingaben) auf einem Rechner
 - (2) Theoretisches (a priori) Vorgehen
Bestimmung des Ressourcenverbrauchs eines Algorithmus (Ausführungszeit, Speicherplatz, ...) als eine Funktion der Größe der betrachteten Fälle

Information 9: Effizienz von Algorithmen

Der Vorteil einer theoretischen Näherung des Ressourcenverbrauchs gegenüber dem empirischen Vorgehen besteht darin, dass die Effizienz eines Algorithmus unabhängig von

- dem eingesetzten Rechner
 - der verwendeten Programmiersprache oder
 - dem Geschick des Programmierers
- bestimmt werden kann.

Die Größe eines Falles, die bei der Bestimmung der theoretischen Effizienz zu bestimmen ist, zielt ab auf den Umfang der Darstellung dieses Falls auf einem Rechner. Formell handelt es sich dabei um eine Anzahl von Bits, in einer konkreten Analyse interessiert die Anzahl der Komponenten eines Falls – auch wenn die Komponenten zur Darstellung üblicherweise mehr als ein Bit benötigen.

- Die Ordnung von $f(n)$, also $O(f(n))$, ist die Menge aller Funktionen $t(n)$, die nach oben durch ein positives reelles Vielfaches von $f(n)$ begrenzt werden, vorausgesetzt n ist genügend groß (größer als eine Schwelle n_0)
- Asymptotische Schreibweise der Ordnung
 - $O(f(n)) = \{t: \mathbb{N} \rightarrow \mathbb{R}^+ \mid (\exists c \in \mathbb{R}^+) (\exists n_0 \in \mathbb{N}) (\forall n \geq n_0) [t(n) \leq c \cdot f(n)]\}$
 - \mathbb{N} natürliche Zahlen, \mathbb{R}^+ nicht-negative reelle Zahlen
 - $f: \mathbb{N} \rightarrow \mathbb{R}^+$ beliebige Funktion

Information 10: Ordnung und asymptotische Schreibweise

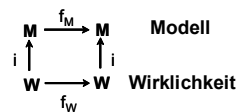
Durch die in Information 10 eingeführte asymptotische Schreibweise wird erreicht, die multiplikative Konstante angemessen in die theoretischen Analysen von Algorithmen einzubeziehen.

3 MODELL

Von besonderer Bedeutung in der Informatik ist der Vorgang des Modellierens, d.h. die Abbildung und Darstellung der Wirklichkeit in einem Modell [Go95].

- Wirklichkeit
 - Dinge, Personen und deren Beziehungen
 - Abläufe in der Zeit
- Modell
 - Begriffe von Dingen und Personen und Beziehungen zwischen diesen Begriffen
 - Abläufe in gedachter Zeit
- Ein Modell ist wahr, wenn die Begriffe die Wirklichkeit richtig wiedergeben

i : Gegenstände der Wirklichkeit ?
 Begriffe der Modellwelt
 f_M : beliebige Beziehung zwischen
 Begriffen der Modellwelt
 f_W : reale Beziehung




Information 11: MODELL – Beziehung zwischen Wirklichkeit und Modell

Das Ziel muss darin bestehen, richtige oder wahre Modelle aufzubauen. Das Modell sollte also die Sachverhalte der Wirklichkeit richtig wiedergeben. Diese Forderung lässt sich durch die Einführung von Abbildungen formal ausdrücken.

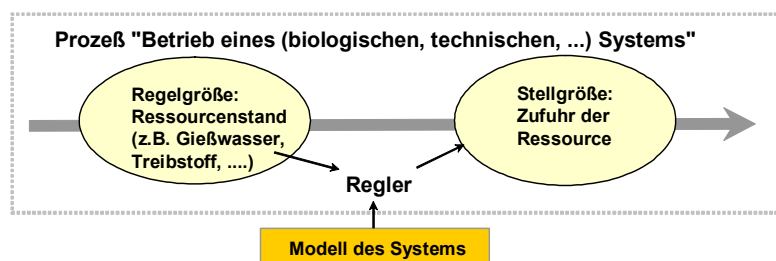
Die Aussage "das Modell ist wahr" bedeutet, dass Beziehungen, die zwischen Gegenständen (bzw. deren Zuständen) innerhalb des Modells bestehen, auch für die entsprechenden Gegenstände innerhalb der Wirklichkeit bestehen müssen. Man sagt dann auch, dass das Beziehungs-Diagramm zwischen Modell und Wirklichkeit kommutiert.

3.1 Ziele von Modellen

Ein Beispiel eines Modells ist ein Plan, der von einem Statiker zu einer in Zukunft zu erbauenden Brücke angefertigt wird. Das Modell soll verhindern, dass die gemäß dem Plan gebaute Brücke einstürzt. Mit dem Modell wird somit eine Vorhersage auf zukünftiges Verhalten ermöglicht.

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck) 

- Gewinn von Einsichten in Vergangenes und Bestehendes sowie Ermöglichung von Vorhersagen über zukünftiges Verhalten
- Die Informationsverarbeitung greift auf der Grundlage der aus den Modellen abgeleiteten Aussagen steuernd oder regelnd in die Wirklichkeit ein



- Welches Ziel wird im Beispiel mit dem Modell verfolgt?
- Wovon hängt die Qualität des Ergebnisses ab?

Interaktion 9: Mit Modellen verfolgte Ziele

Modelle können auch dazu genutzt werden, auf die reale Welt regelnd einzuwirken. Ein Beispiel ist die Regelung einer Ressource, die von einem technischen System für dessen Betrieb benötigt wird, wie die Abbildung in Interaktion 9 zeigt. Konkret könnte das technische System ein Kraftfahrzeug sein und die Ressource der Treibstoff. Ein anderes Beispiel ist das Gießen eines Gartens. In diesem Fall ist das System der Garten (also ein biologisches System), die Ressource ist das Gießwasser und der Regler ist der Gärtner.

Der bislang behandelte Begriff der Regelung unterscheidet sich vom Begriff der Steuerung. Regelung beinhaltet immer eine Rückkopplung zwischen Regler und System. Fehlt ein solcher Rückkopplungsmechanismus, so handelt es sich um eine Steuerung.

- Abstraktion eines Modells bedeutet Weglassen von Details
 - Ziel: Vereinfachung der Informationsverarbeitung
 - Frage: entspricht das abstrahierte Modell noch der Wirklichkeit?
- Validierung eines Modells bedeutet die Überprüfung des Wahrheitsgehalts dieses Modells
 - kann nur durch Experimente und nicht durch logische Schlüsse in der Modellwelt erfolgen
- Spezifikation ist die Beschreibung einer Wirklichkeit
 - ist dann gegeben, wenn die Wirklichkeit einer Gedankenwelt entspringt
- Verifikation ist die Validierung eines Modells gegen eine Spezifikation
 - basiert ausschließlich auf logischen Schlüssen
 - in der Praxis stellt sich das Problem der (zu) hohen Komplexität der Prüfaufgabe

Information 12: Arbeiten mit Modellen

Aufgrund der Komplexität der Wirklichkeit ist man in der Informationsverarbeitung häufig gezwungen, das Modell zu abstrahieren bzw. zu vereinfachen (siehe Information 12). Hierdurch kann es zu einem Realitätsverlust kommen, der Ursache für Fehler und somit ungünstige steuernde Eingriffe sein kann.

Daher ist es wichtig, das Modell auf seinen Wahrheitsgehalt zu überprüfen. Eine solche Überprüfung wird Validierung genannt. Da die Überprüfung in der Wirklichkeit stattfindet, sind hierfür Experimente durchzuführen.

Auf der Grundlage von logischen Schlüssen kann nur dann der Wahrheitsgehalt eines Modells überprüft werden, wenn die Wirklichkeit unserer Gedankenwelt entspringt und als Spezifikation vorliegt. Diese Form der Überprüfung wird als Verifikation bezeichnet.

Modelle werden immer vom Menschen erdacht, er trägt die Verantwortung für sämtliche Interpretationsschritte. Informatiker sollten daher immer verantwortungsvoll mit Modellen umgehen (siehe Information 13).

- Konstruierte und eingesetzte Modelle können zur Veränderung der Wirklichkeit führen
 - Beispiel Telekommunikationssysteme: Das Modell legt die (technische) Kommunikation zwischen den Teilnehmern fest
- Der Konstrukteur sollte sich die Konsequenzen, die durch den regelnden und steuernden Eingriff in die Wirklichkeit entstehen, bewußt machen
- Kein Modell oder Verfahren der Informatik ist von vornherein gut oder schlecht
 - es kommt auf deren Einsatzzweck und Einsatzform an
 - Beispiel: Datenbanken und Datenschutzaspekte

Information 13: Verantwortungsvoller Umgang mit Modellen

Der Informatiker hat sich speziell damit auseinander zu setzen, welche Auswirkungen der Einsatz von Rechnern und Telekommunikation auf die Menschheit hat. Hierin liegen Chancen (z.B. Einsparung von Energie durch Tele-Arbeit), aber auch Gefahren (z.B. Vereinsamung, gläserner Mensch).

3.2 Modell-Beispiel

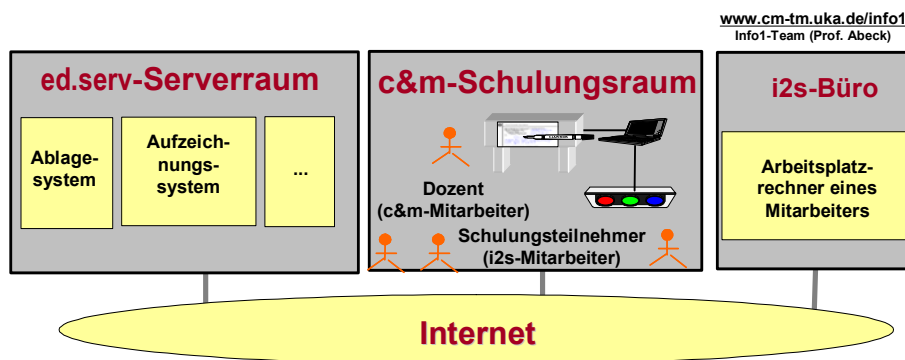
Anhand des in der Kurseinheit INFORMATIK I IM ÜBERBLICK [C&M-IÜ] eingeführten Beispielszenarios des Internet-basierten Wissenstransfers wird in diesem Abschnitt aufgezeigt, wie ein Teil der Wirklichkeit konkret in ein Informatik-gerechtes Modell überführt werden kann. Dabei wird hier der Ansatz der objektorientierten Modellierung zugrunde gelegt, der eine große Bedeutung in der Informatik hat. Als Modellierungssprache kommt die allgemein akzeptierte und standardisierte *Unified Modeling Language* (UML, [Oe03]) zum Einsatz.

Auf die Konzepte der Objektorientierung wird später noch detailliert im Rahmen der Kurseinheit OBJEKTORIENTIERTE PROGRAMMIERUNG [C&M-OP] eingegangen. In diesem Abschnitt wird das Ziel verfolgt, die bislang eher abstrakt beschriebene Aufgabe der Modellierung mittels dieser Konzepte zu konkretisieren.

- Die Ausführungen zum Modell werden im Folgenden an dem Beispiel-Szenario des Internet-basierten Wissenstransfers konkretisiert
- Es wird ein objektorientierter Modellierungsansatz verfolgt
 - als Modellierungssprache wird die UML (Unified Modeling Language) genutzt
- Ein Objekt beinhaltet die Information, die bzgl. des Gegenstandes der Wirklichkeit innerhalb des Modells benötigt wird
- Alternative Formen der Verarbeitung
 1. innerhalb eines Objekts
 2. durch Kommunikation von Nachrichten zwischen zwei Objekten

Information 14: Modell-Beispiel

Das Objekt ist ein Informationsträger innerhalb des Modells, wie in Information 14 ausgeführt ist. Die Verarbeitung der Information erfolgt entweder innerhalb des Objekts oder zwischen zwei Objekten, indem die Objekte untereinander Nachrichten austauschen [BH03].



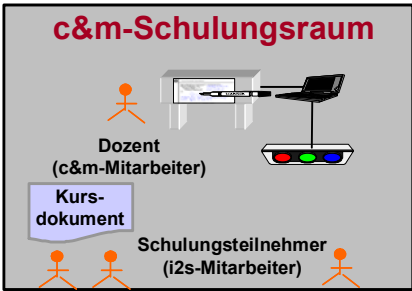
- Das Schulungsunternehmen c&m nutzt die vom IT-Dienstleister ed.serv angebotenen Dienste des Internet-basierten Wissenstransfers, um i2s einen Schulungsdienst anzubieten
- Obige Darstellung beschreibt die Wirklichkeit auf einem informellen Niveau
 - schlägt die Brücke zum Anwender (der "Kunde" des Informatikers)
 - Einstieg in ein im Folgenden weiter zu detaillierendes Modell

Information 15: Beispielszenario

Information 15 zeigt das in [C&M-IÜ] eingeführte Beispielszenario. Die Abbildung kann als eine wirklichkeitsnahe Skizze aufgefasst werden. Die Skizze dient dazu, das mittels Informatiksystemen zu lösende Problem informell zu beschreiben. Die Darstellung ist bewusst frei von fachspezifischen Modellelementen gehalten, damit die Problemstellung und deren Lösung mittels eines Informatiksystems den künftigen Anwender dieses Informatiksystems vermittelt werden kann.

Das Identifizieren von Objekten, die im Beispielszenario auftreten, ist ein wichtiger Schritt der Modellverfeinerung, der nachfolgend an einem Ausschnitt aus dem Beispielszenario durchgeführt wird.

Ausschnitt aus der Realität



c&m-Schulungsraum

Dozent
(c&m-Mitarbeiter)

Kurs-
dokument

Schulungsteilnehmer
(i2s-Mitarbeiter)

Beispiele für Objekte im Modell

Objekt-Art	Objekt-Name
_____	_____
_____	_____
_____	_____
_____	_____

- Kandidaten für Objekte
 - Personen, Gebäude, Räume, Geräte, Dokumente, ...
- Es werden nur ganz bestimmte Informationen zu den realen Gegenständen in dem Objekt gehalten
 - entscheidend ist das mit dem Modell verfolgte Ziel

Interaktion 10: Auftretende Objekte im Beispielszenario

Im vorgegebenen Szenario lassen sich problemlos Objekte identifizieren, durch die reale innerhalb einer c&m-Schulung auftretende "Gegenstände" modelliert werden. Es sollen in Interaktion 10 einige solcher Objekte exemplarisch durch Angabe der Art (Personen, Gebäude, Geräte, Dokumente, ...) und beliebig gewählter Namen genannt werden.

Durch das Hinzufügen von Attributen bzw. Operationen werden die Objekte zu Informationsträgern bzw. informationsverarbeitenden Einheiten.

- Ein Objekt ist durch seinen Zustand und seine Funktionalität bestimmt
 - zur Beschreibung des Zustands dienen Attribute und darin gehaltene Werte
 - die Funktionalität wird durch Operationen, die auch als Methoden bezeichnet werden, beschrieben

- Beispiel: Vorliegendes Kursdokument
 - Attribute
 - Titel: "GRUNDBEGRIFFE DER INFORMATIK"
 - Autor: "Abeck"
 - Seitenanzahl: 34
 - Methoden
 - Wie lautet der Titel? Welcher Autor? Wie viele Seiten?
 - Ändere Seitenanzahl

Information 16: Attribute und Operationen

Die Beschreibung eines Objekts betrifft die zwei in Information 16 genannten Aspekte: Durch die Einführung von Attributen wird das Objekt zu einem Informationsträger. Hierdurch lassen sich alle Aspekte des Objekts beschreiben, die aus der Sicht der Informatiklösung über dieses Objekt bekannt sein sollten. Durch die Angabe von Operationen wird das Objekt zu einer aktiven Einheit, die eine Verarbeitung der in den Attributen oder auch außerhalb des Objekts zugreifbaren Information durchführen kann.

Am Beispiel eines Kursdokuments, das als ein Objekt des dem Beispielszenario zugrunde liegenden Modells identifiziert werden kann, lässt sich die Festlegung von Attributen und Operationen aufzeigen. Es sei betont, dass diese Festlegung ausschließlich von den Anforderungen der zu entwickelnden Informatiklösung abhängt.

Typische Operationen sind das Lesen und Schreiben von Attributwerten des Objekts, wie die in Information 16 genannten Operationen des Beispiel-Objekts verdeutlichen.

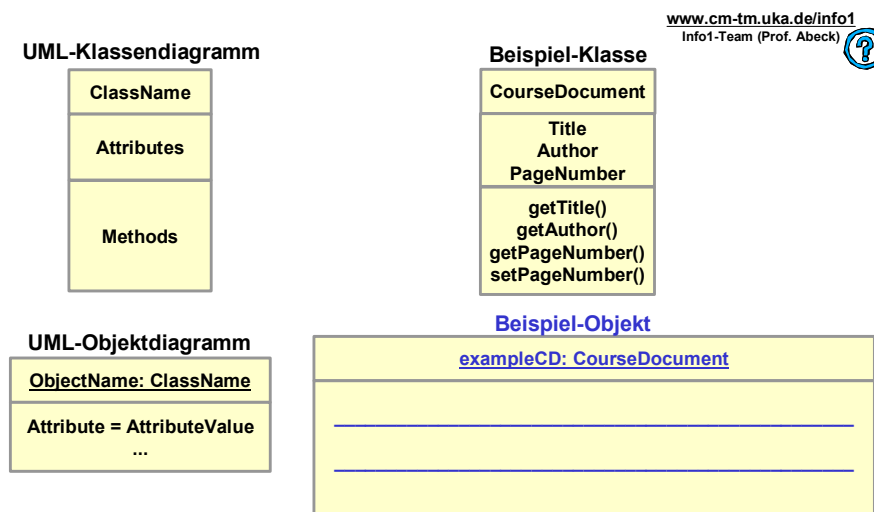
Die Zusammenfassung bzw. die "Klassifikation" von Objekten mit gleichen Merkmalen (d.h. Attribute und Methoden) führt zu dem wichtigen Begriff der Klasse.

- Eine Klasse beschreibt eine Menge von Objekten mit gleichen Merkmalen
- Merkmale eines Objekts sind die Attribute und Methoden dieser Klasse
- Ein Objekt dieser Menge wird auch als Instanz der Klasse bezeichnet
- Eine Klasse beschreibt die Merkmale ihrer Objekte
 - Die Klassenbeschreibung wirkt wie eine Schablone zur Erzeugung (Instanziierung) von ihr zugeordneten Objekten (Instanzen)

Information 17: Klasse

Anhand des Kursdokuments lässt sich der Zusammenhang zwischen einer Klasse und einem Objekt einfach klarmachen: Ein Kursdokument ist eine Klasse, durch die die Merkmale (Attribute und Operationen) der zu dieser Klasse gehörenden Objekte, wie z.B. das Kursdokument mit dem Titel GRUNDBEGRIFFE DER INFORMATIK, beschrieben werden.

Zur Beschreibung von Klassen und Objekten sowie den darin enthaltenen Attributen und Methoden bietet die UML grafische Beschreibungselemente an, durch die eine anschauliche Darstellung von Klassen und Objekten ermöglicht wird.




- Konvention, die für diese Veranstaltung eingeführt wird
 - In entwicklernahen (UML-) Klassen-/Objektdiagrammen sowie in den die Diagramme umsetzenden (Java-) Programmen werden durchgängig englische Bezeichnungen verwendet

Interaktion 11: Klassendiagramm und Objektdiagramm

In der UML werden Klassen und Objekte durch ein Rechteck beschrieben, in dem die Elemente der Klassenbeschreibung (Name, Attribute und Operationen) bzw. Objekte (Name und Attributwerte) in einzelnen Abschnitten aufgeführt werden. Interaktion 11 zeigt die als Diagramme bezeichneten grafischen Beschreibungen und liefert für das Klassen- und Objektdiagramm jeweils ein Beispiel, das das zuvor beschriebene Kursdokument-Objekt

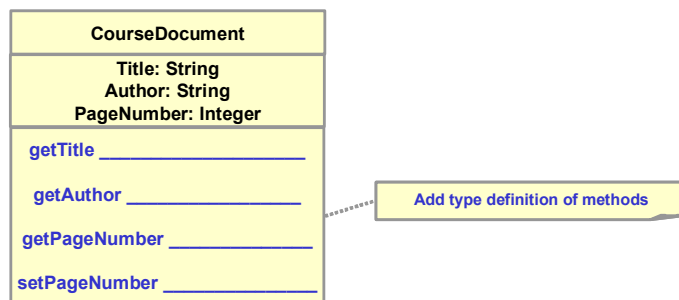
aufgreift. Das Objektdiagramm zum Beispiel-Objekt exampleCD ist gemäß den in Information 16 angegebenen Attributwerten zu vervollständigen.

Klassen- und Objektdiagramme lassen sich unmittelbar in einer objektorientierten Programmiersprache, wie z.B. Java umsetzen. Da in den Java-Programmen, die in nachfolgenden Kurseinheiten entwickelt werden, durchgängig die englische Sprache zur Bezeichnung von Klassen, Objekten, Attributen, Methoden, usw. verwendet werden, ist es sinnvoll, auch in den entwicklernahen UML-Diagrammtypen ausschließlich englischsprachige Begriffe zu verwenden.

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck) 

- In den eingeführten Diagrammen sind die Attribute ausschließlich durch ihre Namen beschrieben
- Das Beispiel zeigt, dass die Werte von einem bestimmten Typ sind
 - Titel und Autor tragen als Werte eine Zeichenkette (String)
 - Seitenanzahl ist eine (positive) ganze Zahl zuzuweisen (Integer)
- Die Klassenbeschreibung kann bei Bedarf um diese Typangaben ergänzt werden

- Beispiel:



Interaktion 12: Ergänzung der Typangabe

Anhand der einfachen Beispiel-Klasse `CourseDocument` wird bereits offensichtlich, dass Attributen Werte unterschiedlicher Wertemengen zugeordnet sein können. Dieser Sachverhalt wird durch eine Typangabe ausgedrückt, die zu einem Attribut im Klassendiagramm angegeben werden kann. Die Typangabe zu den Methoden ist in Interaktion 12 zu ergänzen.

Beispiele für Typen sind `String` (Menge der Zeichenketten) oder `Integer` (Menge der ganzen Zahlen). Auf Typen wird in den nachfolgenden Kurseinheiten detailliert eingegangen.

4 ARCHITEKTUR

In [Oe01] wird die Architektur definiert als die Spezifikation der grundlegenden Struktur eines Systems.

- Es sind verschiedene ineinander greifende Architekturkonzepte zu unterscheiden
 - Rechensystem
 - Architektur eines einzelnen Rechners
 - vorherrschend ist die von-Neumann-Rechnerarchitektur
 - Verteiltes System
 - Architektur eines vernetzten Rechensystems
 - Client-Server-Prinzip
 - Verteilte Anwendung
 - Softwaresystem, das auf den (Client-Server-) Systemen abläuft
 - Mehrschichtenarchitektur

Information 18: ARCHITEKTUR - Überblick

Zunächst wird als System das Rechensystem zugrunde gelegt, dessen Architektur durch den so genannten von-Neumann-Rechner vorgegeben ist.

Rechensysteme werden heute üblicherweise vernetzt (meist über das Internet). Hieraus resultieren Verteilte Systeme, die gemäß einem Architekturprinzip aufgebaut sind, das als Client-Server-Prinzip bezeichnet wird.

Die auf den (Client-Server-) Systemen laufende Anwendungssoftware ist in gewissen Schichten organisiert, wobei jede der Schichten ein bestimmtes Aufgabengebiet in der Anwendung übernimmt. Diese Architektur wird als Mehrschichtenarchitektur (oder auch *N-Layer-* bzw. *N-Tier-Architektur*) bezeichnet.

4.1 Rechensysteme

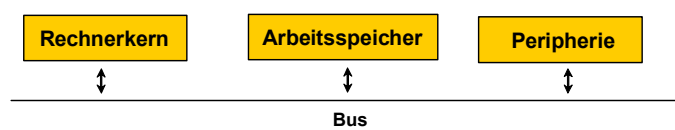
Das Hauptanwendungsgebiet der Informatiksysteme bestand anfangs in der Durchführung von arithmetischen Berechnungen. Konrad Zuse hat daher auch von Rechenautomaten gesprochen. Im Bereich der eher wissenschaftlichen Informatik ist man bei diesen Begriffen geblieben; man spricht heute auch üblicherweise von Rechner oder Rechensystem [Go95].

- Der Rechner ist das technische Hilfsmittel zur Realisierung (Implementierung) von Algorithmen
 - ursprünglich vor allem arithmetische Berechnungen
- Wichtige Elemente
 - Speicher für Eingabedaten und (Zwischen-) Ergebnisse
 - Programmsteuerung für die Entscheidung über die nächsten Rechenschritte
- Geschichte
 - Babbage entwickelt den ersten programmgesteuerten, mechanisch arbeitenden Rechner ('Analytische Maschine')
 - erster elektromechanisch arbeitender Rechner von Zuse ('Z3')
 - Arbeitsweise der meisten elektronisch arbeitenden Rechner geht auf Arbeiten von von-Neumann zurück

Information 19: Rechnerentwicklung

Die zwei für die Durchführung von Berechnungen notwendigen Elemente liegen auf der Hand. Man kann diese direkt aus der Überlegung ableiten, was benötigt wird, um z.B. eine per Hand durchgeführte Addition auf einer Maschine auszuführen:

- Die auf das Papier geschriebenen Zahlen müssen in der Maschine gespeichert und abgerufen werden können. Dabei müssen Eingabe-, Zwischenergebnis- und Ausgabedaten gespeichert werden können.
- Außerdem wird eine Programmsteuerung benötigt, die die gerade kennen gelernten Konstrukte, wie Bedingungen, Schleifen oder Unterprogramme, auszuführen gestattet.



- Rechnerkern: führt das Programm aus
- Arbeitsspeicher: enthält Programme und Daten
- Peripherie: Ein-/Ausgabegeräte, (Peripherie-) Speicher
- Bus: verknüpft seriell oder parallel die oben genannten Komponenten
- Rechnerkern führt fortlaufend Befehle aus
 - z.B. Addition zweier Zahlen, boolesche Verknüpfung, Sprungbefehl
 - liest Operanden in den Speicher und schreibt Ergebnisse in den Speicher
→ Problem des "von-Neumannschen Flaschenhalses"

Information 20: Aufbau eines von-Neumann-Rechners

Die heute eingesetzten Rechensysteme entsprechen ihrem Aufbau nach einem von-Neumann-Rechner. Die Abbildung in Information 19 zeigt den Aufbau eines von-Neumann-Rechners, der auch als Princeton-Architektur bezeichnet wird. Falls mehr als ein Bus verwendet wird, spricht man von einer Harvard-Architektur.

- Herzstück der von-Neumann-Architektur ist der Rechnerkern, der auch als Zentralprozessor bezeichnet wird.
- Eine wichtige Eigenschaft des Arbeitsspeichers besteht darin, diesen nicht nur zur Speicherung von Daten, sondern auch des auszuführenden Programms zu nutzen. Das war von vornherein nicht gegeben, da die Programme zunächst auf getrennten Medien, wie z.B. Lochstreifen, gehalten wurden.
- Im Zusammenhang mit der Peripherie werden ebenfalls Prozessoren eingesetzt. Da diese hauptsächlich Kommunikationsaufgaben zu erledigen haben, werden sie als Kommunikationsprozessoren bezeichnet (im Gegensatz zu den obigen Zentralprozessoren).
- Auf dem Bus werden Befehle, Adressen, Daten, Steuersignale übertragen.

Bei der Ausführung eines Programms spielen diese vier Einheiten zusammen und bilden ein reaktives System. Es wiederholt sich dabei folgender so genannter Befehlszyklus: Der Rechnerkern holt den gerade anstehenden vom Programm vorgegebenen Befehl und führt ihn aus. Dazu sind entsprechende Speicherzugriffe erforderlich (sowohl zum Holen des Befehls, als auch zum Lesen und Schreiben der Daten).

Das Problem des so genannten von-Neumannschen Flaschenhalses resultiert aus physikalischen Gegebenheiten des (relativ schnellen) Rechnerkerns und des (relativ langsamen) Speichers.

www.cm-tm.uka.de/info1
Info1-Team (Prof. Abeck)

- Organisation des Speichers
 - Folge von Speicherzellen der Größe von 8 Bits (1 Byte)
 - Durchnummerieren der Speicherzellen ab 0, wobei die Nummern Adressen heißen
 - nicht nur Daten, sondern auch die Programmbefehle liegen im Speicher
- Komplexere Peripheriegeräte (z.B. Drucker, Netzwerkkarte) sind mit einem eigenen (Spezial-) Prozessor ausgestattet, der den Zentralprozessor entlastet
- Der Zugriff auf den Bus erfolgt prioritätengesteuert
- Die meisten heute eingesetzten Prozessoren sind Signalprozessoren

Information 21: Speicher und Peripherie

Der Speicher besteht aus Speicherzellen, die heute immer die Größe von 8 Bit, d.h. 1 Byte besitzen.

Daten und Befehle können länger als 1 Byte sein. In diesem Fall dient die niedrigste Adresse der beteiligten Bytes als Adresse des Datums oder des Befehls.

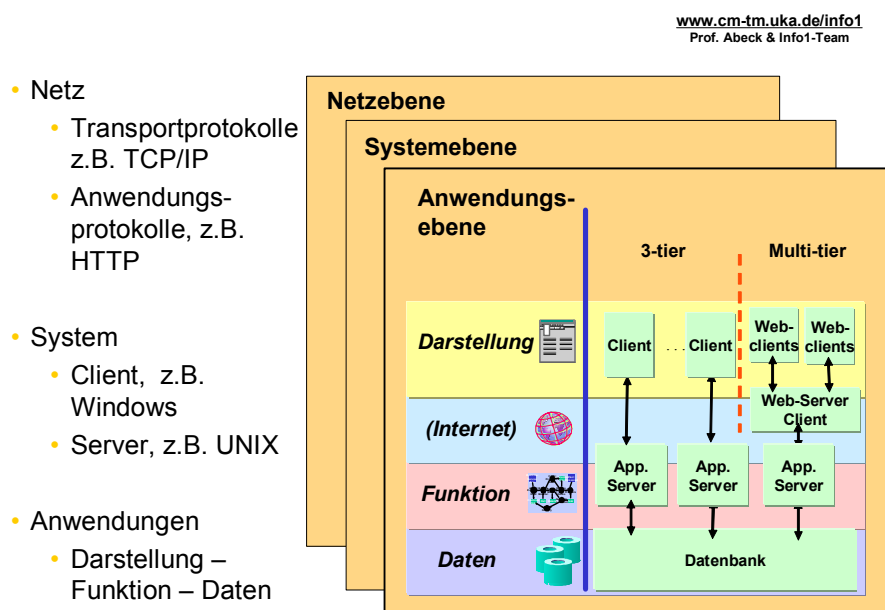
Viele Peripheriegeräte können direkt auf den Speicher zugreifen, ohne dabei den Zentralprozessor belasten zu müssen. Man spricht hierbei von einem direkten Speicherzugriff (*Direct Memory Access* DMA).

Die Prioritäten verhelfen dem Bus, bei konkurrierenden Zugriffen zu entscheiden, welcher Zugriffswunsch zuerst erfüllt wird.

Signalprozessoren verrichten Spezialaufgaben der technischen Signalverarbeitung. Neben den Prozessoren (Hardware) benötigt man die entsprechenden Programme (Software), um diese Aufgaben zu lösen.

4.2 Verteilte Systeme und Verteilte Anwendungen

Die bislang betrachtete Architektur war auf ein einzelnes Rechensystem beschränkt. Die heute eingesetzten Informatiksysteme setzen sich üblicherweise aus mehreren über ein Kommunikationsnetz verbundene Systeme zusammen. Auf dem so entstehenden Verteilten System läuft eine Anwendungssoftware, die ebenfalls ganz bestimmten Architekturkonzepten unterliegt.



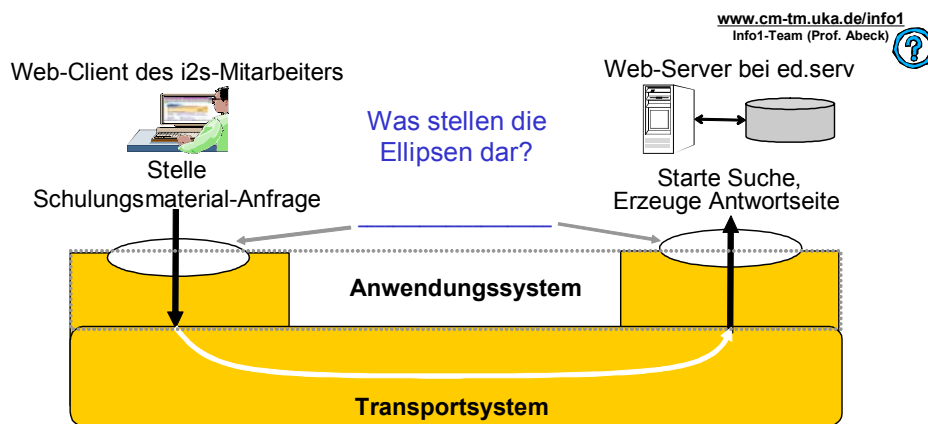
In Information 22 [C&M-AUR] sind die drei Aspekte eines Verteilten Systems – Netz, (Rechen-) System, Anwendung – im Überblick dargestellt.

Heutige und zukünftige Kommunikationsnetze werden maßgeblich durch die Internet-Technologie bestimmt, deren wichtigste Bestandteile das IP-Protokoll und die Web-Technologien [PD00] darstellen. Um die Internet-Standards herum scharen sich eine Vielzahl von konkreten Netztechnologien und -produkten, die einen bestimmten Einsatzbereich (z.B. Hochleistungs- oder Mobilkommunikation) adressieren. Hieraus resultiert eine heterogene und komplexe Netzinfrastruktur mit unterschiedlichen Eigenschaften. Von den über diese Netzinfrastruktur angeschlossenen Rechensystemen und den darauf laufenden (verteilten) Anwendungen wird ein Kommunikationsdienst mit bestimmten, möglichst garantierten Qualitätseigenschaften gefordert (so genannter *Quality of Service*, QoS).

Die Rechensysteme [CD01] arbeiten auf einer Client-Server-Basis, wobei im Client-Bereich die PCs mit dem Windows-Betriebssystem vorherrschen und im Server-Bereich zudem verstärkt Varianten des Unix-Betriebssystems (z.B. Linux, SUN Solaris, HP UX, IBM AIX) und Großrechner-Betriebssysteme zum Einsatz kommen. Da einige der Großrechner-Betriebssysteme eher eine "Altlast" darstellen, werden sie auch als *Legacy*-Systeme bezeichnet.

Die Verteilten Anwendungen [In02] laufen auf dieser heterogenen (d.h. eine Vielzahl von Betriebssystemen einbeziehenden) vernetzten Systemlandschaft. In Verteilten Anwendungen lassen sich drei elementaren Aspekte der Daten, der Funktionen und der Darstellung unterscheiden.

Die Struktur eines Kommunikationsnetzes wird anhand des Beispielszenarios in Interaktion 13 skizziert.

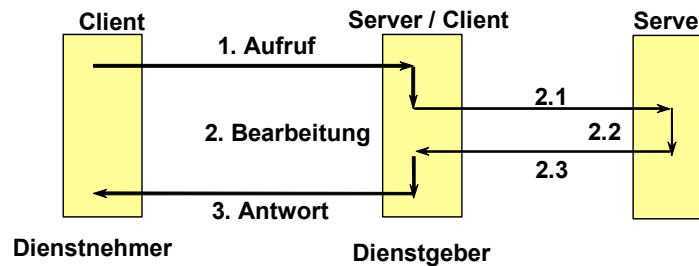


- Transportsystem stellt eine sichere Ende-zu-Ende-Verbindung her
 - wird aus Vermittlungsknoten (im Internet so genannte IP-Router) gebildet
- Anwendungssystem stellt anwendungsnahe Kommunikationsdienste bereit
 - im Beispiel ist der Dienst der Zugriff auf Web-Seiten

Interaktion 13: Nutzung des Kommunikationsnetzes im Beispielszenario

Wie die Abbildung zeigt, setzt sich das Kommunikationssystem aus zwei Teilsystemen zusammen. Während das Transportsystem den sicheren Transport der Daten zwischen den beiden kommunizierenden Rechensystemen (den Endsystemen) übernimmt, stellt das auf dem Transportsystem aufsetzende Anwendungssystem höherwertige Dienste bereit, wie beispielsweise der Zugriff auf Web-Seiten, der in Interaktion 13 gezeigt ist.

Im Beispiel zur Nutzung des Kommunikationsnetzes waren die zwei kommunizierenden Systeme ein Web-Client und ein Web-Server. Information 23 stellt das allgemeine Prinzip von Client-Server dar.



- Unterbeauftragung: Server beauftragt im Rahmen der Bearbeitung eines Auftrags einen weiteren Server
- Ein Beispiel eines wichtigen Client-Server-Basisprotokolls ist der Remote Procedure Call (RPC)
 - realisiert einen entfernten Prozeduraufruf

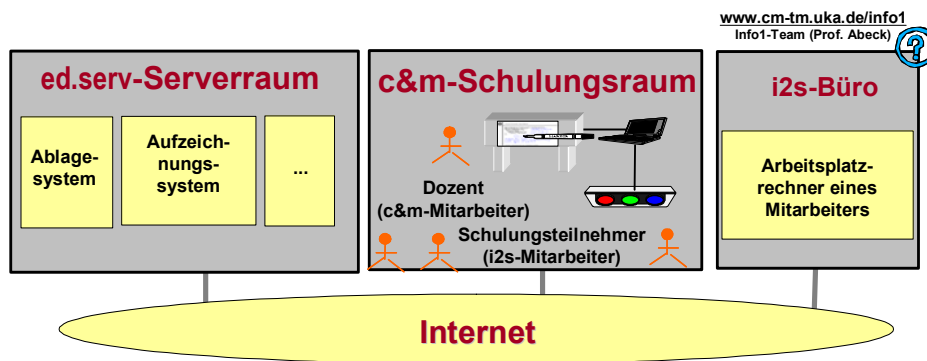
Information 23: Client-Server-Prinzip

Das in Information 23 gezeigte Client-Server-Prinzip dient zur allgemeinen Strukturierung der Kooperationsbeziehung auf der System-Ebene. Das Prinzip geht davon aus, dass der Kooperation von zwei Systemen eine Dienstbeziehung zugrunde liegt. Dabei übernimmt das eine System die Rolle des Diensthnehmers – genannt Client – und das andere System die Rolle des Dienstbringers – genannt Server.

Ein System kann in verschiedenen Dienstbeziehungen stehen und in einer Beziehung die Rolle des Clients und in einer anderen die des Servers übernehmen. Ein Spezialfall eines solchen Rollentausches ist die Unterbeauftragung, in dem ein Server zur Bearbeitung eines Dienstes ein anderes System als Server in Anspruch nimmt und hierzu die Rolle des Clients einnimmt.

Es besteht ein enger Zusammenhang zwischen dem Client-Server-Prinzip und den Protokollen auf der Netzebene. So liegt dem zuvor in Interaktion 13 skizzierten Zugriff auf einen Web-Server das Client-Server-Prinzip zugrunde. Das Beispiel kann problemlos dahingehend erweitert werden, dass der vom Web-Client angefragte Web-Server zur Erbringung der Anfrage andere Server-Systeme in Anspruch nehmen muss.

Aus der Sicht der verteilten Anwendung muss ein Protokoll vorhanden sein, das das Client-Server-Prinzip mit der Übertragung des Aufrufs und der Antwort flexibel unterstützt. Ein solches Protokoll ist der *Remote Procedure Call (RPC)*, der in unterschiedlichen Ausprägungen in verschiedenen standardisierten Verteilten Systemarchitekturen auftritt.



- Welche Systeme übernehmen die Rolle eines Servers bzw. eines Client?
 Server-Systeme: _____
 Client-Systeme: _____
- Zwischen welchen dieser Systeme könnte ein RPC ablaufen und was wäre ein Beispiel eines entfernten Aufrufs?

Interaktion 14: Client-Systeme und Server-Systeme im Beispielszenario

In der abschließenden Interaktion 14 ist zu ermitteln, an welchen Stellen im Beispielszenario Client-Systeme und Server-Systeme auftreten und es sind Beispiele für das Zusammenspiel dieser Systeme auf der Grundlage des *Remote Procedure Calls* anzugeben.

Mit den Begriffen der Information, des Algorithmus, des Modells und der Architektur sowie dem in der Kurseinheit INFORMATIK I IM ÜBERBLICK [C&M-IÜ] dargestellten Systembegriff wurde in den Kern der Informatik eingeführt. Es bestehen zahlreiche Beziehungen zwischen diesen Begriffen, die man sich klarmachen sollte.

Diese vier eingeführten Grundbegriffe stehen im Zentrum der Informatik. Alle folgenden Begriffe, die in den weiteren INFORMATIK-I-Kurseinheiten eingeführt werden, sind um dieses Zentrum herum angeordnet.

VERZEICHNISSE

Abkürzungen und Glossar

Abkürzung oder Begriff	Langbezeichnung und/oder Begriffserklärung
Algorithmus	Ein mechanisch ausführbares (Rechen-) Verfahren.
Architektur	Spezifikation der grundlegenden Struktur eines Systems.
<i>Client-Server</i>	Allgemeines Architekturprinzip zur Strukturierung von Kooperationsbeziehungen auf der System-Ebene.
Codierung	Festlegung der Darstellung von Informationen in Form von Zeichen auf der Grundlage eines Codes.
c&m	cooperation & more Im Szenario des Internet-basierten Wissenstransfers auftretendes Schulungsunternehmen.
DMA	<i>Direct Memory Access</i> Direkter Zugriff eines Peripheriegeräts auf den Speicher, ohne dabei den Zentralprozessor zu belasten.
Euklidischer Algorithmus	Der älteste und damit erste Algorithmus, der den größten gemeinsamen Teiler von zwei ganzen Zahlen berechnet.
ed.serv	educational.services Name des IT-Dienstleisters, der im Szenario des Internet-basierten Wissenstransfers auftritt.
<i>Legacy-Systeme</i>	Systeme, die auf veralteten Technologien basieren aber aus Kostengründen nicht erneuert werden können bzw. sollen. Wörtliche Übersetzung von <i>Legacy</i> : Altlast
i2s	intelligent internet solutions Name des als Kunden einer Schulung auftretenden Unternehmens, das im Szenario des Internet-basierten Wissenstransfers auftritt.
Kursdokument <i>Course Document</i>	Ein im Beispielszenario des Internet-basierten Wissenstransfers auftretendes Objekt. Die Klassenbeschreibung zu diesem Objekt umfasst z.B. als Attribut den Titel (<i>title</i>) und als eine Methode das Lesen des Titels (<i>getTitle()</i>)
Mehrschichtenarchitektur	Architektur einer verteilten Anwendung, in der die Aspekte der Anwendung in mehrere Schichten (insbes. Darstellung, Verarbeitung, Daten) aufgeteilt wird. Englisch: <i>N-Layer Architecture</i>
Modell	Abbildung und Darstellung eines Ausschnitts der Wirklichkeit.

QoS	<i>Quality of Service</i> Die von einem IT-Dienst (z.B. Kommunikationsdienst) garantierten Qualitätseigenschaften, wie z.B. maximale Antwortzeit oder minimale Verfügbarkeit.
Semiotik	Lehre der Zeichen, die sich aus der Syntax, der Semantik und der Pragmatik zusammensetzt.
Signal	Darstellung einer Mitteilung durch die zeitliche Veränderung einer physikalischen Größe.
UML	<i>Unified Modeling Language</i> (vereinheitlichte Modellierungssprache) Sprache, die aus graphischen Elementen besteht und zur semi-formalen Beschreibung von beliebigen Gegenständen (z.B. Software-Systeme oder Geschäftsbereiche) genutzt wird.
von-Neumannscher Flaschenhals	Bezeichnung eines Engpasses, der in der von-Neumann-Architektur aufgrund der im Vergleich zur Prozessorbearbeitungszeit relativ langen Speicherzugriffszeiten entsteht. Synonymer Begriff: von-Neumann-Engpaß

Index

American Standard Code for Information Interchange 11	<i>Legacy-Systeme</i> 28
Architektur 23	Modell 15
Client-Server-Prinzip 29	Rechensysteme 28
Digitalisierung 4	<i>Remote Procedure Call</i> 29
Direct Memory Access 27	Semiotik 6
Euklidische Algorithmus 12	Unified Modeling Language 18
Kommunikationsnetze 27	Verteilte Anwendungen 28
Kursdokument 21	von-Neumannscher Flaschenhals 26

Informationen und Interaktionen

Information 1: GRUNDBEGRIFFE DER INFORMATIK	2
Information 2: Signalklassen.....	3
Information 3: Nachricht und Information	4
Information 4: Wissen.....	5
Information 5: Semiotik	6
Information 6: Alphabet und Codes	10
Information 7: ALGORITHMUS - Informationsverarbeitung.....	11
Information 8: Arten der Berechnung einer Funktion $f:A \rightarrow B$	12
Information 9: Effizienz von Algorithmen.....	14
Information 10: Ordnung und asymptotische Schreibweise	15
Information 11: MODELL – Beziehung zwischen Wirklichkeit und Modell.....	15
Information 12: Arbeiten mit Modellen	17
Information 13: Verantwortungsvoller Umgang mit Modellen	18
Information 14: Modell-Beispiel.....	19
Information 15: Beispielszenario	19
Information 16: Attribute und Operationen.....	21

Information 17: Klasse	22
Information 18: ARCHITEKTUR - Überblick	24
Information 19: Rechnerentwicklung	25
Information 20: Aufbau eines von-Neumann-Rechners	25
Information 21: Speicher und Peripherie	26
Information 22: Verteiltes System	27
Information 23: Client-Server-Prinzip	29
Interaktion 1: INFORMATION – Signal und Inschrift.....	2
Interaktion 2: Interpretationsvorschrift α	4
Interaktion 3: Codierung von Informationen.....	7
Interaktion 4: Shannonsche Informationstheorie	8
Interaktion 5: Entscheidungsgehalt	9
Interaktion 6: Codierung von Zeichen	10
Interaktion 7: Euklidischer Algorithmus.....	13
Interaktion 8: Ablaufsteuerung.....	13
Interaktion 9: Mit Modellen verfolgte Ziele.....	16
Interaktion 10: Auftretende Objekte im Beispielszenario	20
Interaktion 11: Klassendiagramm und Objektdiagramm	22
Interaktion 12: Ergänzung der Typangabe	23
Interaktion 13: Nutzung des Kommunikationsnetzes im Beispielszenario	28
Interaktion 14: Client-Systeme und Server-Systeme im Beispielszenario.....	30

Literatur

- [AL+03] Sebastian Abeck, Peter C. Lockemann, Jochen Schiller, Jochen Seitz: Verteilte Informationssysteme: Integration von Datenübertragungstechnik und Datenbanktechnik: dpunkt.verlag, 2003.
- [BB93] Gilles Brassard, Paul Bratley: Algorithmik – Theorie und Praxis, Wolfram's Verlag, 1993.
- [BG73] Friedrich L. Bauer, Gerhard Goos: Informatik – Eine einführende Übersicht – Erster Teil, Springer-Verlag, 1973.
- [BH03] Bernd Brügge, Christian Herzog: Einführung in die INFORMATIK I, Vorlesungsfolien, TU München, 2003.
- [C&M-AUR] Cooperation&Management, ARCHITEKTUR UND RAHMENWERKE, Kursdokument zur Vorlesung "ENTWICKLUNG VON INTERNET-SYSTEMEN UND WEB-APPLIKATIONEN", <http://www.cm-tm.uka.de/iswa>, Universität Karlsruhe (TH), C&M (Prof. Abeck)
- [C&M-BUS] Cooperation&Management: BITÜBERTRAGUNG UND SICHERUNG, Kursdokument zur Vorlesung "KOMMUNIKATION UND DATENHALTUNG", <http://www.cm-tm.uka.de/kud>, Universität Karlsruhe (TH), C&M (Prof. Abeck)
- [C&M-IÜ] Cooperation&Management: INFORMATIK I IM ÜBERBLICK, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/inf01>, Universität Karlsruhe (TH), C&M (Prof. Abeck)
- [C&M-OP] Cooperation&Management: OBJEKTORIENTIERTE PROGRAMMIER-

UNG, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck)

[C&M-PG] Cooperation&Management: PROGRAMMIERGRUNDLAGEN, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck)

[Go95] Gerhard Goos: Vorlesungen über Informatik, Band 1: Grundlagen und funktionales Programmieren, Springer-Verlag 1995.

[PD00] L. L. Peterson, B. S. Davie, Computernetze - Ein modernes Lehrbuch: dpunkt.verlag, 2000.