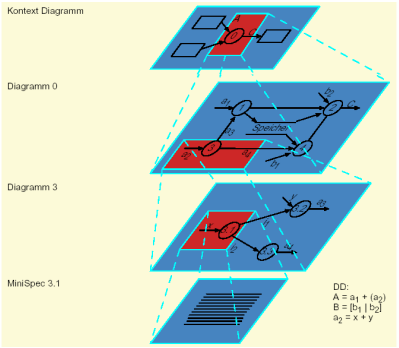
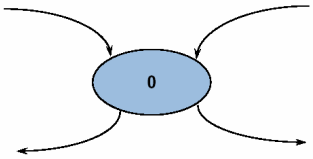
Einführung



- von Tom DeMarco (späte 1970er)
- Erstellung einer formalen Systembeschreibung im Rahmen der Softwareentwicklung (⇒ Analysephase); graphisch
- Resultat: hierarchisch gegliedertes technisches Anforderungsdokument
- Top-Down-Vorgehen
  ⇒ vom Kontextdiagramm zu den Mini-Specs
- Grundform: statische Analyse, erweiterbar zur dynamischen Analyse

Darstellungstechniken

Kontextdiagramm



Ein Kontextdiagramm besteht aus einem einzigen Prozess (dem System) und seinen Verbindungen zur Umwelt.

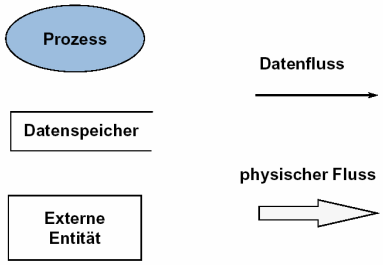
- stellt den Null-Prozess dar
- Wurzel des Analyse-Baums: Aus diesem Knoten sind sämtliche anderen Knoten des Baums erreichbar.
- Komponenten:
  - Kreis: Null-Prozess
  - Rechtecke: Komponenten, die mit dem System interagieren
  - Pfeile vom oder zum Null-Prozess: Daten- oder Kontrollflüsse

  ⇒ Regeln: Nur ein Prozess, keine Datenspeicher, alle Terminatoren!

⇒ offenes System

Datenflussdiagramm DFD

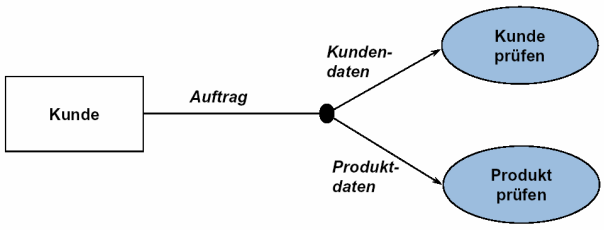
- Übersicht über die ausgeführten Funktionen bzw. Prozesse
- Zeigen, **„was“** geschieht! (nicht „wie“ es geschieht)
- Keine Ablauffolgen, keine zeitlichen/logischen Zusammenhänge!
  Sondern stellt Kommunikation zwischen den Prozessen dar.
  - Auslösung von Prozessen (durch Ereignisse) nicht spezifizierbar
  - Eingehende Datenflüsse bedeuten nicht, dass diese unbedingt zur Ausführung benötigt werden.
    ⇒ Nacheinander folgende Prozesse müssen nicht zwingend durch einen Datenfluss miteinander verbunden sein!
  - Paralleler Ablauf der Prozesse möglich
- Komponenten:



- physischer Fluss: Alle Flüsse, die keine Datenflüsse sind.
- Prozess: auch Funktion genannt; Menge von Aktivitäten
- Datenfluss: Datenbewegung zwischen Quelle und Senke
- Datenspeicher: Persistente Daten (in Dateien, DB)
- Externe Entität/Terminatoren: Schnittstelle zur Umwelt; Bezeichnet mit dem Namen der Schnittstelle (externer Partner); Begrenzungen des betrachteten Systems
- Bei jedem Datenfluss muss mindestens einer der Endpunkte (Quelle und/oder Ziel) ein Prozess sein.

Datenfluss: Ausgetauschte Nachrichten

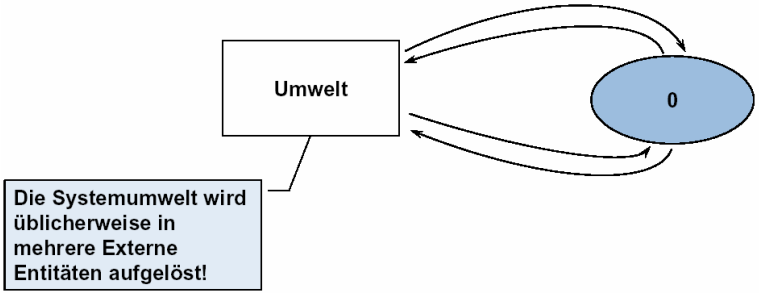
- Gerichteter Kanal, in dem bestimmte Nachrichten von einer Quelle zu einer Senke ausgetauscht werden. Jeder Datenfluss erhält einen Namen. Ausnahme: Datenfluss vom/zum Speicher und greift auf den gesamten Inhalt zu.
  - Prozesse transformieren oder verändern Daten dabei! (Daten „wandern“ nicht einfach durch)
    ⇒ Veränderungen: physisch oder logisch (veränderter Status)
  - Output-Daten müssen immer von seinen Input-Daten abgeleitet werden können
  - Splitten (bzw. Zusammenführen) von Datenflüssen
    - Duplizierte Daten gleichzeitig an verschiedene Teile des Systems gesandt.
    - Zusammengesetzte Datenpakete werden in verschiedene Teilpakete aufgesplittet.



- Die Nachricht beinhaltet festgelegte Daten
  ⇒ Data Dictionary
- Medium des Datenflusses nicht festgelegt
  - direkt mündlich oder fernmündlich
  - schriftlich durch Belege und Formulare
  - elektronisch
- Direkte (zwischen zwei Prozessen) und indirekte Datenflüsse (Datenspeicher als Zwischenstation zwischen zwei Prozessen)
- Regeln
  - Für jeden Datenfluss muss mindestens die Quelle oder die Senke ein Prozess sein.
  - Ausgeschlossen damit:
    - Datenflüsse zwischen zwei Externen Entitäten (d. h. Datenfluss im Umsystem)
    - Datenflüsse zwischen zwei Datenspeichern
    - Datenflüsse zwischen einer Externen Entität und einem Datenspeicher
    - Zulässig hingegen sind Datenflüsse zwischen zwei Prozessen

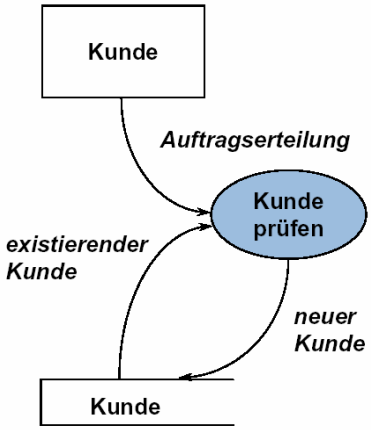
Terminatoren / Externe Entitäten

- Terminatoren repräsentieren Externe Entitäten, mit denen das System kommuniziert.
  - Personen oder Unternehmen ausserhalb des betrachteten Unternehmens
  - Personen oder Abteilungen innerhalb des betrachteten Unternehmens, aber ausserhalb der Kontrolle des zu modellierenden Systems
  - ein anderes System, mit dem das zu modellierende System kommuniziert.
  - Bsp. Benutzer des Systems
- Externe Entitäten können vom Systemanalytiker nicht beeinflusst bzw. verändert werden.



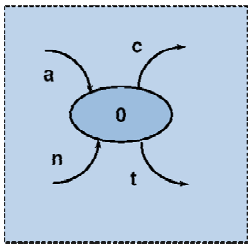
Datenspeicher

- Als Quelle für den Datenspeicher kommen nur Prozesse in Frage
- Bedeutungen: CREATE/INSERT, UPDATE, DELETE bzw. READ
- Regeln
  - Datenspeicher sind keine Senken (d. h. mind. ein Pfeil führt vom Speicher weg ⇒ READ)
  - Datenspeicher werden nur eingezeichnet, wenn sie als Schnittstelle zwischen Prozessen dienen.
- Ausnahme: Keine Schnittstelle zwischen zwei Prozessen



- Datenspeicher als Kommunikationsmittel
- Doppelfunktion des Prozesses vorausgesetzt (Kunden prüfen und Kunden anlegen)

Verfeinerung von DFD



Kontextdiagramm

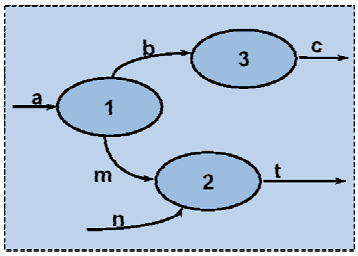


Diagramm der ersten Stufe (Diagramm 0)

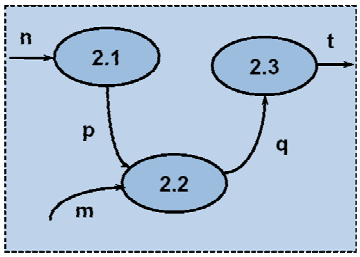


Diagramm der zweiten Stufe (Diagramm 2)

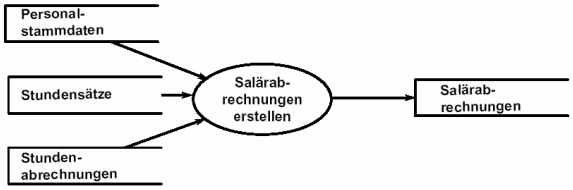
- Kontextdiagramm => Diagramm 0: Keine Verfeinerung von Datenflüssen; typischerweise neue Prozesse sowie Speicher (inkl. möglicher Flüsse) eingefügt.
- Gleicher Datenflussname in den Diagrammen verschiedener Stufen bedeutet gleicher Datenfluss

Mini-Spezifikationen

- Grundelemente zur Darstellung prozeduraler Verfahrensschritte
  - Sequenz: Abarbeiten aller Befehle entsprechend ihrer Anordnung
  - Selektion: Abarbeiten eines Befehls/einer Sequenz hängt von einer Bedingung ab.
  - Iteration: Wiederholen eines Befehls/einer Sequenz in Abhängigkeit einer Bedingung

Textuell: Sequenz, Selektion, Iteration

- Sequenz
  Ergibt sich aus Nacheinanderschreibung einzelner Verfahrensschritte (= Aktionen).
  - Einfach Sequenz: <aktion1>.<aktion2>. …
  - Block: Beginn <aktion1>.<aktion2>. … Ende
- Selektion
  - Einfache Selektion:
    Wenn <Bedingung> dann <Aktion>.
    If <condition> then <action>.
  - Binäre Selektion:
    Wenn <Bedingung> dann <Aktion1> sonst <Aktion2>.
    If <condition> then <action1> else <action2>.
  - Mehrfachselektion:
    Im Falle von
    <Bedingung1>: <Aktion1>. <Bedingung2>: <Aktion2>. …
    Case of
    <condition1>: <action1>. <condition2>: <action2>. …
- Iteration
  - Iteration mit Vorbedingung:
    Solange <Bedingung> erfüllt ist, wiederhole <Aktion>.
    While <condition> do <action>.
  - Iteration mit Nachbedingung:
    Wiederhole <Aktion> solange bis <Bedingung>.
    Repeat <action> until <condition>.
  - Zählschleife:
    Von <Anfangswert> bis <Endwert> <Aktion>.
    For <begin value> to <end value> <action>.

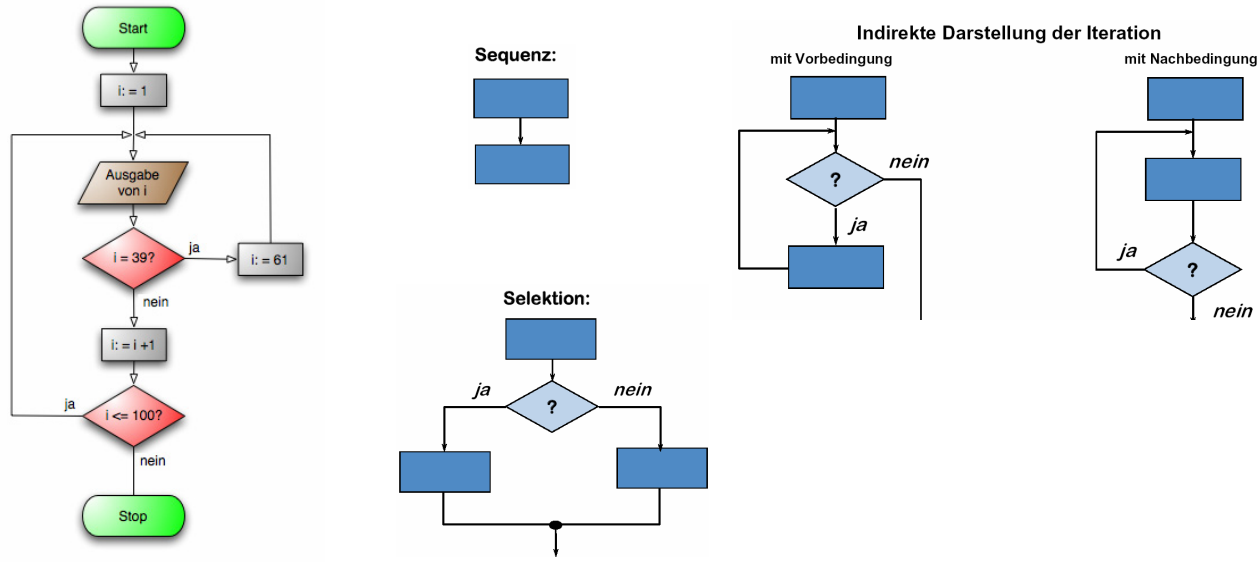


Lese die Stundensätze.
Wiederhole für alle Mitarbeiter
    Lese die Personalstammdaten des Mitarbeiters.
    Lese die Stundenabrechnungen des Mitarbeiters.
    Berechne das Monatssalär des Mitarbeiters.
    Schreibe die Salärabrechnung des Mitarbeiters.
Ende Wiederholung.

Graphisch: Programmablaufpläne

Grundmodell

- Aktionsblock: Rechteck; Beinhaltet Anweisung bzw. Aktion; Genau eine ausgehende Kante
- Fallunterscheidung: Raute; Beinhaltet Bedingung; meist zwei ausgehende Kanten (ja/nein)

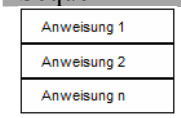


- Nachteile:
  - Wenige Strukturelemente ⇒ Modellieren von Abläufen nur indirekt möglich
  - Gefahr unstrukturierter Entwürfe

  ⇒ Struktogramme als Alternative

Struktogramm (wikipedia)

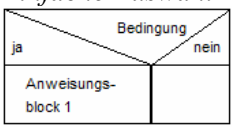
Sequenz



Jede Anweisung wird in einen rechteckigen Strukturblock geschrieben. Die Strukturblöcke werden nacheinander von oben nach unten durchlaufen.
Leere Strukturblöcke sind nur in Verzweigungen zulässig.

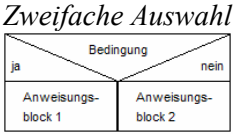
Selektion

Einfache Auswahl



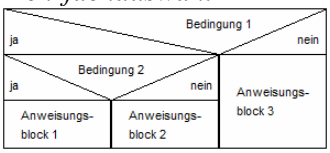
Nur wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Trifft die Bedingung nicht zu (falsch), wird der Durchlauf ohne eine weitere Anweisung fortgeführt (Austritt unten).

Zweifache Auswahl



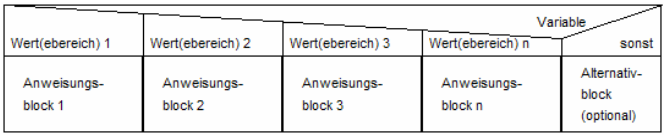
Wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Trifft die Bedingung nicht zu (falsch), wird der Anweisungsblock 2 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Austritt unten nach Abarbeitung des jeweiligen Anweisungsblocks.

Mehrfachauswahl



Auch "verschachtelte" Auswahl genannt, da eine weitere Bedingung folgt. Die Verschachtelung ist ebenso im Nein-Fall (noch) möglich

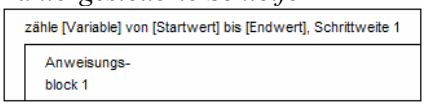
Fallauswahl



Besonders bei mehr als drei abzuprüfenden Bedingungen geeignet. Der Wert von "Variable" kann bedingt auf Gleichheit wie auch auf Bereiche (größer/kleiner bei Zahlen) geprüft werden und der entsprechend zutreffende "Fall" mit dem zugehörigen Anweisungsblock wird durchlaufen.

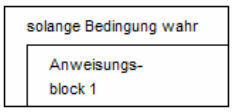
Iterationen

Zählergesteuerte Schleife



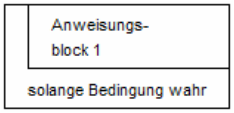
Wiederholungsstruktur, bei der die Anzahl der Durchläufe festgelegt ist. Als Bedingung muss eine Zählvariable angegeben und mit einem Startwert initialisiert werden. Ebenso muss ein Endwert und die (Zähl-)Schrittweite angegeben werden. Nach jedem Durchlauf des Schleifenkörpers (Anweisungsblock 1) wird die Zählvariable um die Schrittweite inkrementiert (bzw. bei negativer Schrittweite dekrementiert) und mit dem Endwert verglichen. Ist der Endwert überschritten, wird die Schleife verlassen.

Iteration mit Vorbedingung („kopfgesteuert“)



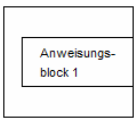
Wiederholungsstruktur mit vorausgehender Bedingungsprüfung. Der Schleifenkörper (Anweisungsblock 1) wird nur durchlaufen, wenn (und solange) die Bedingung zutreffend (wahr) ist.
Diese Symbolik wird auch für die Zählschleife (Anzahl der Durchläufe bekannt) benutzt.

Iteration mit Nachbedingung („fussgesteuert“)



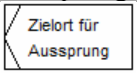
Wiederholungsstruktur mit nachfolgender Bedingungsprüfung. Der Schleifenkörper (Anweisungsblock 1) wird mindestens einmal durchlaufen, auch wenn die Bedingung von Anfang an nicht zutreffend (falsch) war.

Endlosschleife



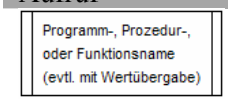
Kann allenfalls durch einen Aussprung (break) verlassen werden.

Aussprung



Der Aussprung (break) stellt das dar, was Nassi und Shneiderman mit den Struktogrammen eigentlich vermeiden wollten: Die Sprunganweisung (unbedingter Sprung Goto).

Aufruf



Symbolik zum Aufruf eines Unterprogramms bzw. einer Prozedur oder Funktion. Nach Durchlauf dieser wird genau zu der aufrufenden Stelle zurückgesprungen und der nächstfolgende Strukturblock durchlaufen.

Geschäftsregeln (Business rules)

- Berücksichtigt bei der Ausführung von Prozessen anfallende Entscheidungssituationen.
  - Bedingungen möglicherweise nicht unabhängig voneinander. (d. h. auch, dass einzelne Bedingung zur Überprüfung überflüssig werden)
  - ⇒ logische Reihenfolge
    ⇒ Entscheidungsbaum
- ECA-Struktur zur Modellierung von Geschäftsregeln (E: Events; C: Conditions; A: Actions)

Textuell

- Strukturiertes Deutsch: WENN (Bedingungskomponente) … DANN (Aktionskomponente) …
- Einleiten der Geschäftsregel: Auslösendes Ereignis explizit nennen.
- Bsp.:
  Bei Vorlage eines Schecks …

WENN (vereinbarte Kreditgrenze überschritten) UND
(bisheriges Zahlungsverhalten einwandfrei) UND
(Überschreitungsbetrag kleiner oder gleich 500)
DANN Scheck einlösen.

WENN (vereinbarte Kreditgrenze überschritten) UND
(bisheriges Zahlungsverhalten einwandfrei) UND
(Überschreitungsbetrag über 500)
DANN Scheck einlösen,
Neue Konditionen vorlegen.

WENN (Kreditgrenze überschritten) UND
(Zahlungsverhalten nicht einwandfrei)
DANN Scheck nicht einlösen

WENN Vereinbarte Kreditgrenze nicht überschritten
DANN Scheck einlösen.

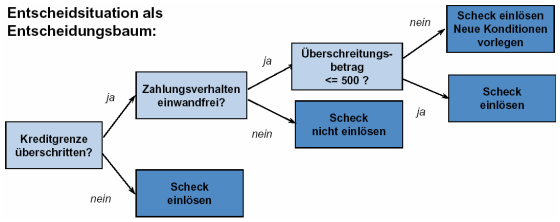
Graphisch

Entscheidungstabelle

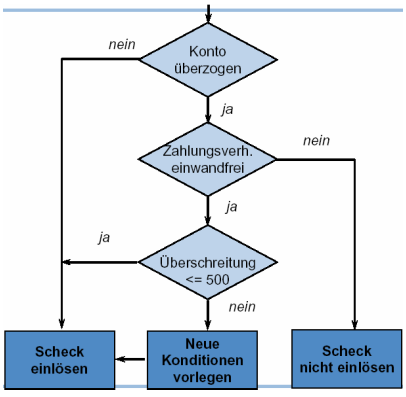
		R1	R2	R3	R4
B1	Kreditgrenze überschritten	J	J	J	N
B2	Zahlungsverhalten einwandfrei	J	J	N	-
B3	Überschreitungs betrag <= 500	J	N	-	-
A1	Scheck einlösen	X	X		X
A2	Scheck nicht einlösen			X	
A3	Neue Konditionen vorlegen		X		

⇒ max. $2^3 = 8$ Möglichkeiten

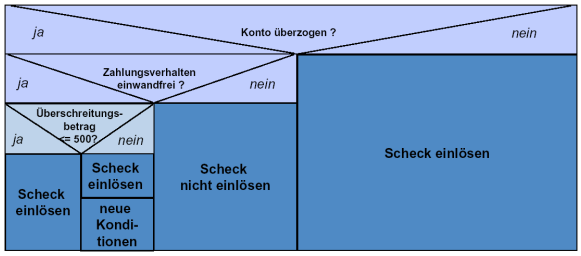
Entscheidungsbaum



Programmablaufplan



Struktogramm



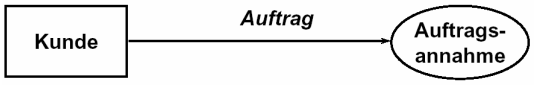
Strukturiertes Deutsch

```
WENN Konto überzogen
DANN
    WENN Zahlungsverhalten einwandfrei
    DANN
        WENN Überschreitungsbetrag <= 500
        DANN
            Scheck einlösen.
        SONST
            Scheck einlösen.
            Neue Konditionen vorlegen.
    SONST Scheck nicht einlösen.
SONST Scheck einlösen.
```

Data Dictionaries

- Definiert jeden Datenflussnamen und jeden Speichernamen genauer:
  Verzeichnis, das Informationen über die Struktur von Daten, ihre Eigenschaften sowie ihre Verwendung enthält. (⇒ Konsistenzüberwachung)
- Syntax

=	Komposition
+	Konkatenation
( )	Optionalität
n{ }m	Iteration
[ ]	Alternativmenge
¦	Separator für die in [ ] aufgeführten Alternativen
**	Kommentar

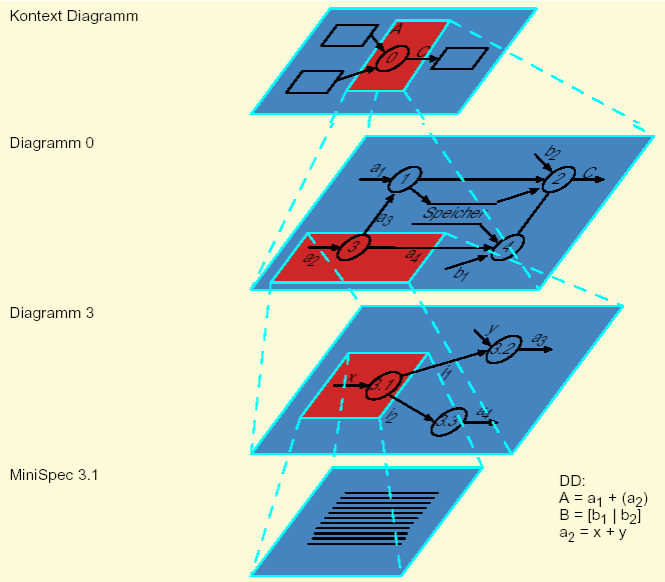


Beispiel für die Anwendung der Notation zur Erstellung eines Data Dictionaries:

- AUFTRAG = AUFNR + AUFDAT + KUNDDAT + 1{AUFPOS}∞ + LWUNSCH
- KUNDDAT = KUNDNR + KUNDNAME + KUNDADRESSE + (GEBDAT)
- KUNDADRESSE = PLZ + ORT + STRASSE
- AUFPOS = POSNR + PRODDAT + MENGE + PREIS
- PRODDAT = PRODNR + PRODBEZ
- LWUNSCH = ['Einzellieferung' ¦ 'Gesamtlieferung']

Dekomposition: Gleichgewicht hierarchischer DFD

Die DFDs eines Systems können hierarchisch in Schichten angeordnet werden.
⇒ Zerlegung eines komplexen Gesamtmodells in Teilmodelle.



- Alle Datenflüsse, die in den übergeordneten Prozess eingehen, müssen in untergeordneten DFD als Terminatoren auftauchen.
- In dem untergeordneten Diagramm darf kein Terminator auftreten, der sich nicht auf einen in den übergeordneten Prozess ein- oder ausgehenden Datenfluss beziehen lässt.
- *Lebenszyklus von Daten*
  Durchlaufen die Daten nie den ganzen Zyklus (Anfangs- und Endterminatoren mindestens 1x gelesen), dann ist das Modell fehlerhaft.