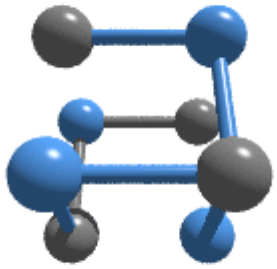


Vorlesung Softwaretechnik
- Strukturierte Analyse / Real Time Analysis -

Prof. Dr. Klaus-Peter Fähnrich

WS 2005/2006

Überblick LE 14

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

LE 14: Strukturierte Analyse

- Das Hierarchiekonzept
- Das Kontextdiagramm
- Verfeinerte DFD
- Data Dictionary -Einträge und Datenintegrität
- Mini-Spezifikation
- Methodik
- Qualitätssicherung
- Wertung

LE 15: Strukturierte Analyse/Real Time Analysis

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Lernziele

1. Vor- und Nachteile der strukturierten Analyse
2. Basiskonzepte und ihre Kombination in SA
3. Zusammenhang zu ER-Modellen und Funktionsbäumen
4. Erstellen eines vollständigen SA-Modell für eine gegebene Problemstellung
5. Einzuhaltende Regeln und methodische Schritte bei der Erstellung eines SA-Modells

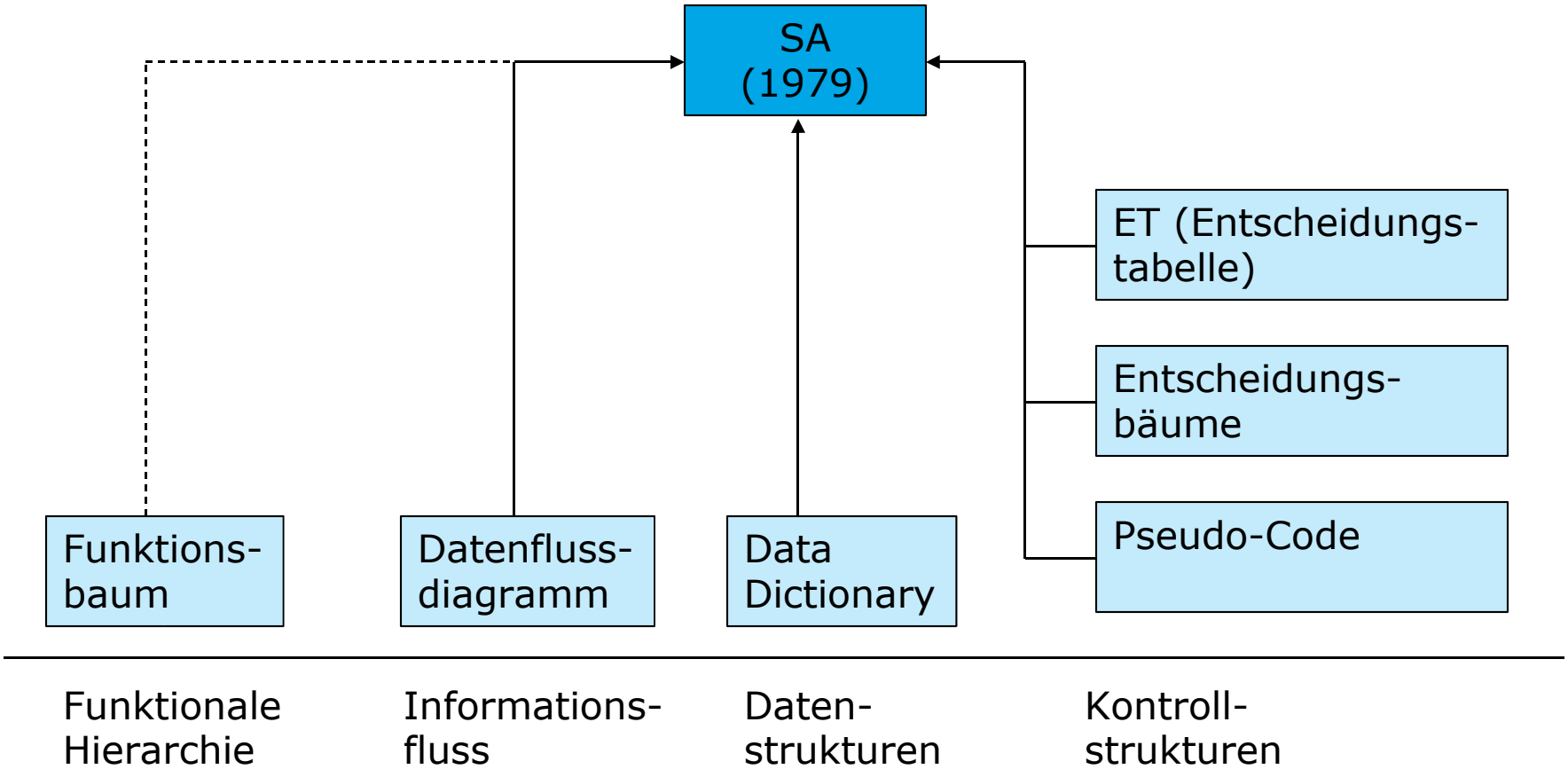
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Definition

- Die **Strukturierte Analyse** (SA) :
  - Beschreibt eine Methodenklasse
  - Wichtige SA-Varianten:
    - Weinberg 1978: »Structured Analysis«
    - Gane/Sarson 1979: »Structured Systems Analysis«
    - McMenamin/Palmer 1984: »Essential Systems Analysis«
    - Yourdon 1989: »Modern Structured Analysis«
  - SA ist Stand der Technik und Industriestandard
  - SA nach DeMarco wird hier vorgestellt.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Kombination von Basiskonzepten SA



Legende:
A ⟶ B: A ist in B enthalten
A ⇢ B: A ist implizit in B enthalten

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

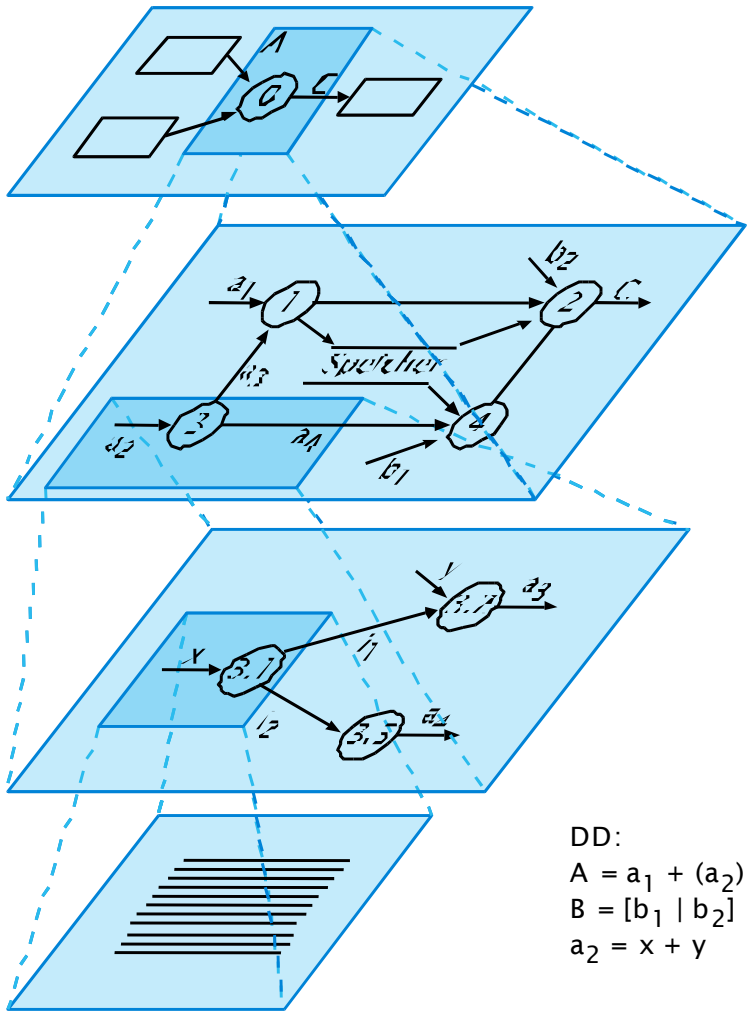
Aufbau

- **Hierarchisierung** zur Steigerung der Übersichtlichkeit
- Datenflussdiagramme (DFD) werden hierarchisch verfeinert
- Kontextdiagramm steht als abstraktes DFD in der obersten Hierarchieebene.
- Jeder Prozess wird in untergeordneten Diagrammen verfeinert.
- Ist keine Verfeinerung mehr möglich, wird eine textuelle Mini-Spezifikation erstellt.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Hierarchiekonzept

- Kontextdiagramm
- Diagramm 0
- Diagramm 3
- MiniSpec 3.1



DD:
$A = a_1 + (a_2)$
$B = [b_1 \mid b_2]$
$a_2 = x + y$

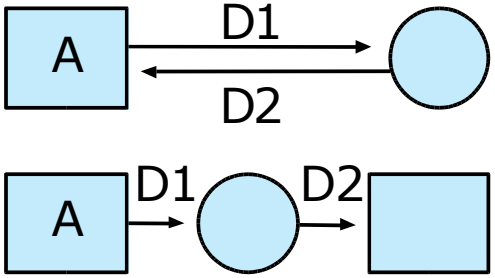
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Kontextdiagramm - Syntaxregeln

- Beschreibt die Schnittstellen des Systems zur Umwelt
- Es nimmt im SA-Modell eine Sonderstellung ein, was Syntax und Semantik betrifft:
  - Enthält nur einen Prozess, der die Nummer 0 enthält und das Gesamtsystem repräsentiert .

  - Enthält mindestens eine Schnittstelle.

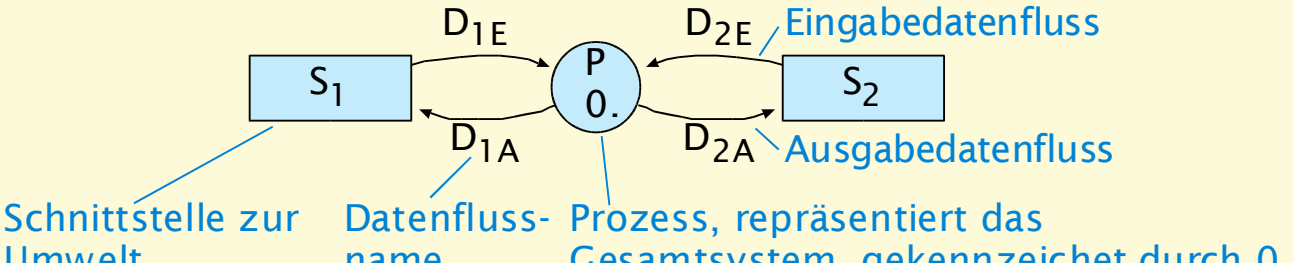
  - Zwischen den Schnittstellen gibt es keine Datenflüsse.
  - Enthält keinen Speicher.
  - Jede Schnittstelle ist i. A. nur einmal vorhanden
    - Ausnahme: Übersichtlichkeit



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

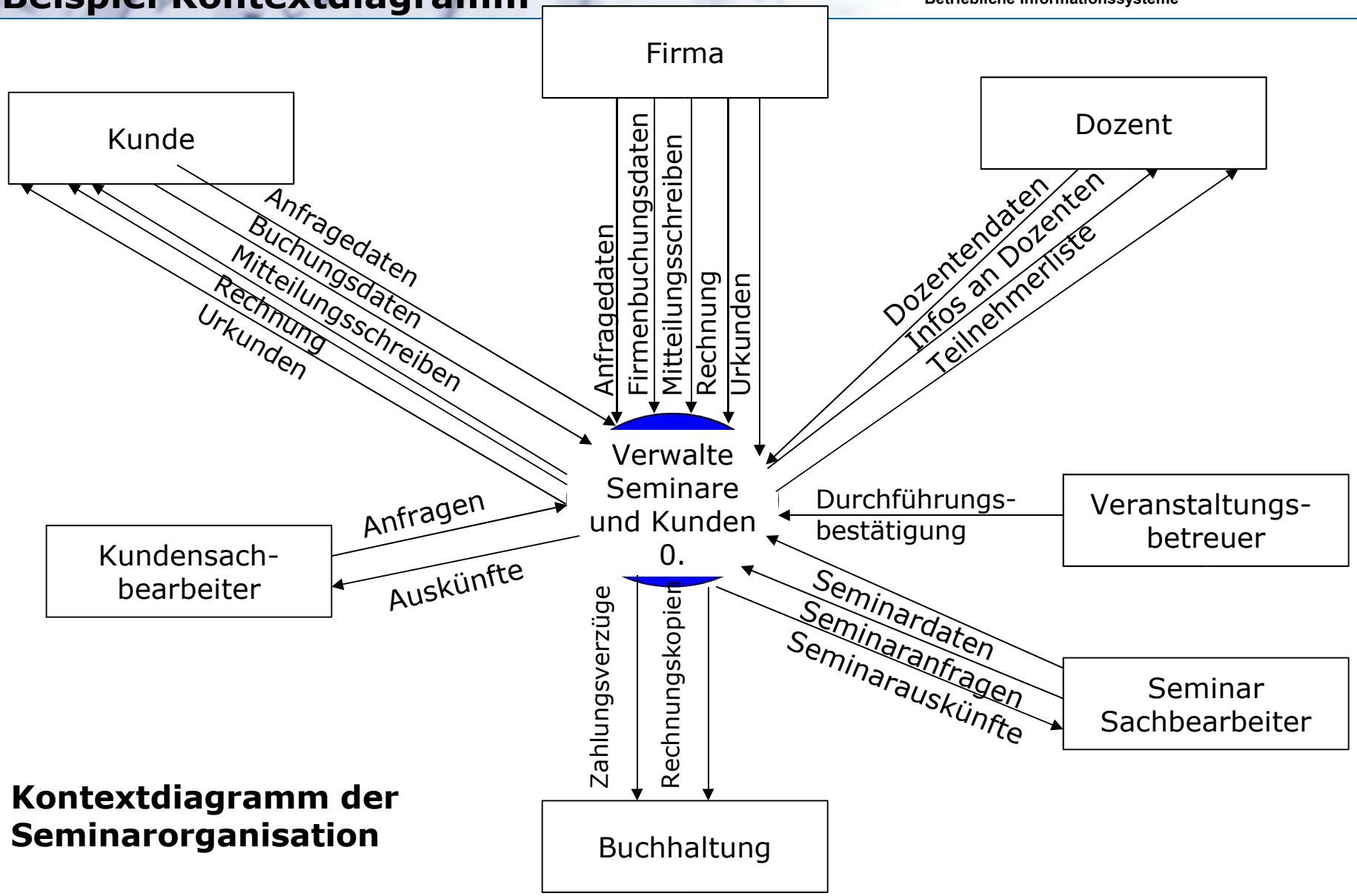
Kontextdiagramm - semantische Regeln

- Beschreibt Anwendungsbereich des zu modellierenden Systems *(problem domain).*
- Zeigt Datenflüsse, die Systemgrenzen passieren.
- Ist die Zusammenfassung von Diagramm 0. (siehe Folie 7)
- Kann es von derselben Schnittstelle mehrere Instanzen geben, wird sie einmal dargestellt.
- Gibt es wenige gleichartige Schnittstellen mit unterschiedlichen Datenflüssen, dann getrennte Darstellung.
- Schnittstelle muss ursprüngliche Quelle oder Senke einer Information angeben
- Wahl einer Schnittstelle abstrahiert von der konkreten Eingabe oder Ausgabe.
  - d. h. Tastatur und Drucker i. A. keine Schnittstelle.



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Beispiel Kontextdiagramm

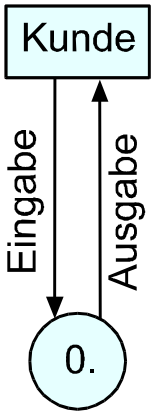


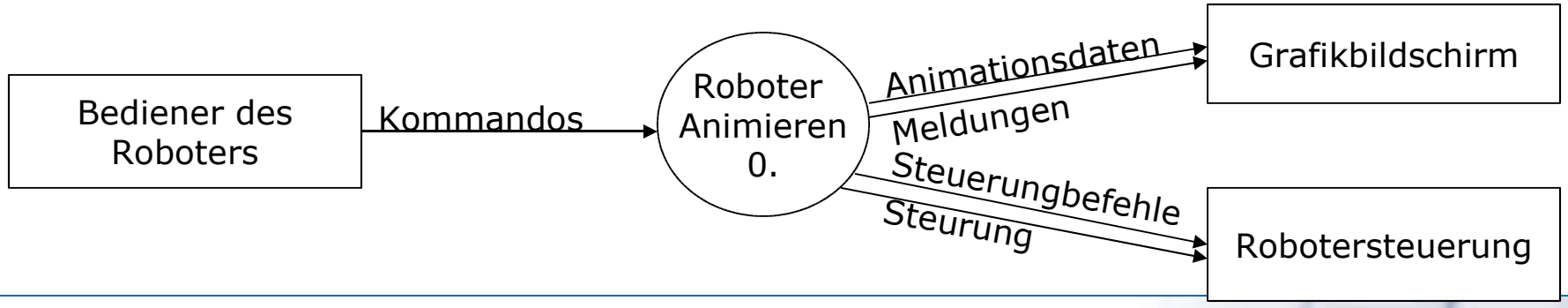
Kontextdiagramm der Seminarorganisation

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Abstraktion

- Datenflüsse auf angemessenem Abstraktionsniveau beschreiben
- Zu detailliert: Dann unübersichtlich, überladen
- Zu abstrakt: Dann nichtssagend
- Richtschnur:
  - Anhand des Kontextdiagramms müssen für einen Außenstehenden die wesentlichen Informationen über die Umwelt erkennbar sein.
  - Namensgebung muss problembezogen sein.
  - Alle Datenflüsse auf demselben Abstraktionsniveau.
- Beispiel »Animation eines Roboter«:
  - Da die Animation im Mittelpunkt steht, ist Grafikbildschirm eine eigene Schnittstelle.
  - Modelliert man auch die Ansteuerung eines Roboters, kommt die Schnittstelle Robotersteuerung hinzu.

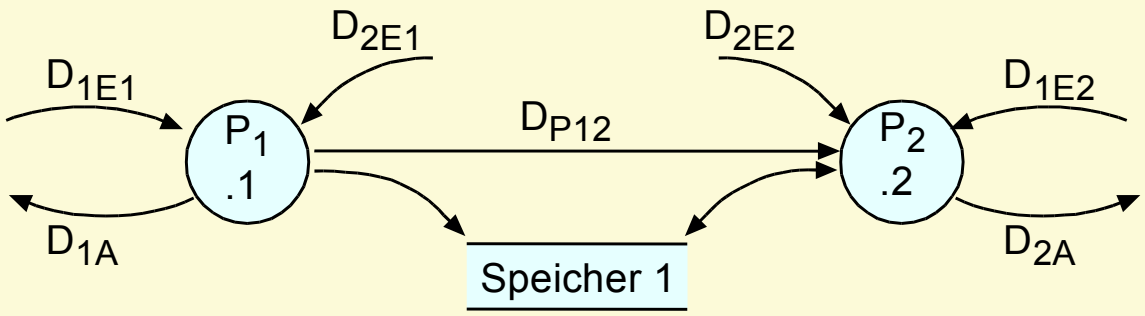




UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

DFD 0 = Verfeinerung des Kontextdiagramms

- Regeln
  - Prozess 0 wird in Teilprozesse (hier: P_1 und P_2) zerlegt
  - Datenflüsse werden ebenfalls verfeinert (hier: D_{1E} und D_{2E})
  - Speicher werden eingeführt (hier: Speicher 1).



DD:
$D_{1E} = D_{1E1} + D_{1E2}$
$D_{2E} = (D_{2E1}) + D_{2E2}$

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

DFD: Eigenschaften und Regeln

Verfeinerte DFD's

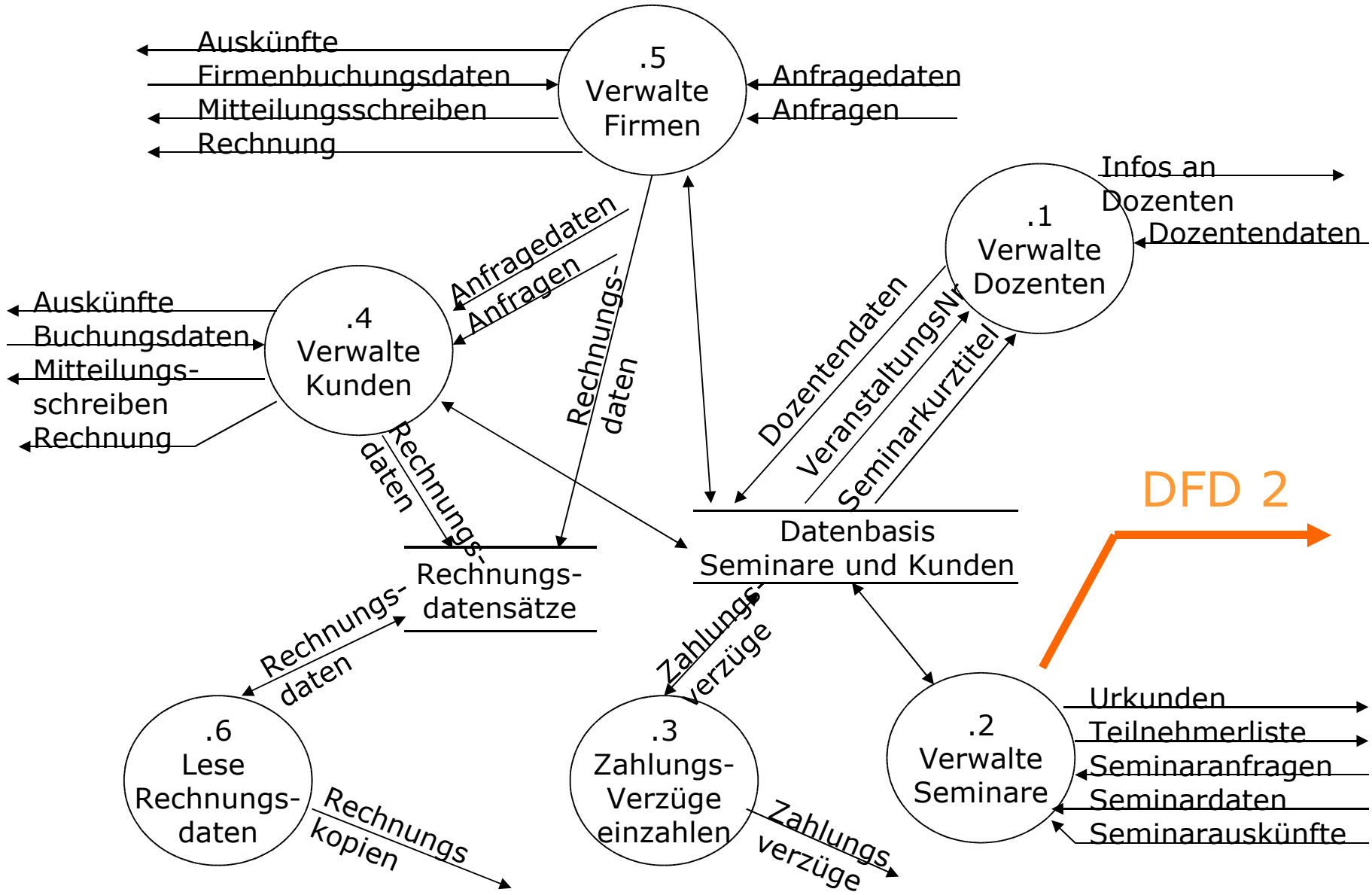
- Prozess 0 im Kontextdiagramm wird in Teilprozesse gegliedert
- Darstellung erfolgt im Diagramm 0 (DFD 0).
- Jeder Prozess wird fortlaufend nummeriert.
- Es werden Speicher eingefügt
  - Ist die Ermittlung der Anzahl und Struktur der Speicher schwierig, sollte zunächst ein ER-Modell erstellt werden.
  - Evtl. die im ER-Modell gefundenen Speicher für die Darstellung im SA-Modell komprimiert als Datenbasis darstellen.

Regeln für DFDs

- Schnittstellen können nicht verfeinert werden.
- Speicher können nicht verfeinert werden.
- Speicher und Schnittstellen können aber auf allen Verfeinerungen unverändert wiederholt werden.
- Anzahl der Prozesse auf einem Diagramm soll nicht größer als 7 sein.
- Abstraktionsniveau der Prozesse und Datenflüsse sollte gleich sein.
- Prozesse werden von 1 beginnend fortlaufend nummeriert.
- Vor jeder Zahl steht ein Punkt: .1, .2, .3
- Jedes Diagramm trägt die Nummer, die seine Stellung in der Hierarchie angibt
  - DFD 4.3: Verfeinerung des Prozesses 3 des DFD 4
- Zur eindeutigen Kennzeichnung wird vor jeden Prozess die DFD-Nummer gesetzt
  - Prozess 4.3.1: 1. Prozess im Diagramm 4.3
- Die Nummernsystematik wird i. A. vom eingesetzten CASE-Werkzeug verwaltet.

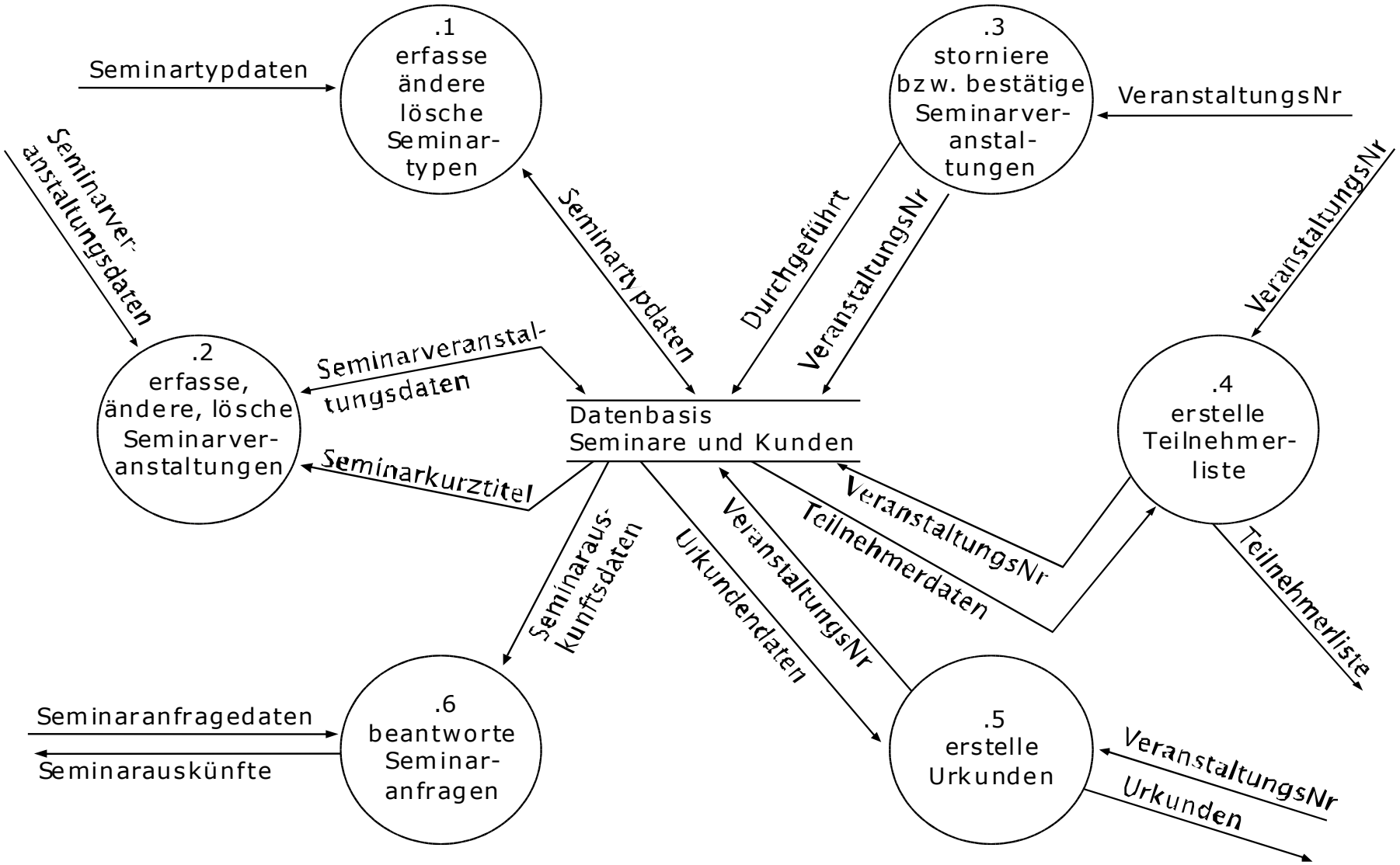
DFD 0: Verwalte Seminare und Kunden der Seminarorganisation

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

DFD 2: Verwalte Seminare in der Seminarorganisation



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Data Dictionary-Einträge und Datenintegrität I

Verfeinerung der Daten

- Parallel zur Verfeinerung der DFDs und der Prozesse werden auch die Daten verfeinert.
- Jeder Datenflusspfeil erhält einen Datenflussnamen
  - Ausnahme: Datenfluss geht zum oder kommt vom Speicher und greift auf den gesamten Inhalt zu.
- Jeder Datenflussname ist im Data Dictionary definiert.
- Jeder Speicher trägt einen Namen.
- Jeder Speichername ist im Data Dictionary definiert.

Beispiel »Seminarorganisation«:

- Beim Übergang vom Kontextdiagramm auf DFD 0 wurden keine Datenflüsse verfeinert.
- Es wurden neue Datenflüsse zwischen Speichern und Prozessen eingefügt.
- In DFD 4 (*Verwalte Kunden*) wurden Datenflüsse verfeinert, dies muss im Data Dictionary festgehalten werden:

```
Anfragedaten  = Personaldaten + (Firmendaten)
Buchungsdaten = Anmeldedaten + Abmeldedaten
```

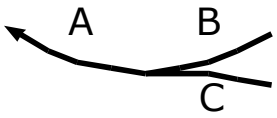
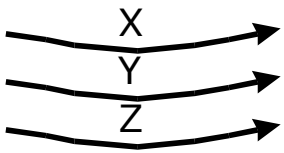
- Sind die Datenflussnamen in mehreren DFDs identisch, dann handelt es sich um die gleichen Datenflüsse.

Data Dictionary-Einträge und Datenintegrität II

- Alle Datenflüsse des untergeordneten DFDs müssen entweder
  - im übergeordneten DFD unter gleichem Namen erscheinen oder
  - Teilkomponente eines Datenflusses sein und damit im DD beschrieben werden.
- Ist diese Eigenschaft zwischen allen Diagrammen erfüllt, spricht man von einem ausbalancierten Datenmodell (balancing).
- Vorteile ausbalancierter Datenmodelle:
  - Einarbeitung in das System wird erleichtert, da nach dem Kontextdiagramm keine neuen Datenflüsse hinzukommen.
  - Stellt man auf tieferer Ebene einen fehlenden Datenfluss fest, dann kann er neu eingezeichnet werden.
    - Kann man den neuen Datenfluss als Teil eines bestehenden identifizieren, braucht nur der DD-Eintrag ergänzt zu werden.
    - Ist dies aus fachlicher Sicht nicht sinnvoll, dann muss auf allen übergeordneten Diagrammen der neue Datenfluss nachträglich eingezeichnet werden.
- SA-Hierarchiekonzept ist ein Makromechanismus
  - Alle Datenflussdiagramme müssen innerhalb der Hierarchie ineinander substituierbar sein (durch balancing sichergestellt).
  - Es gibt keinen Schutzmechanismus
    - Namen müssen global eindeutig sein.

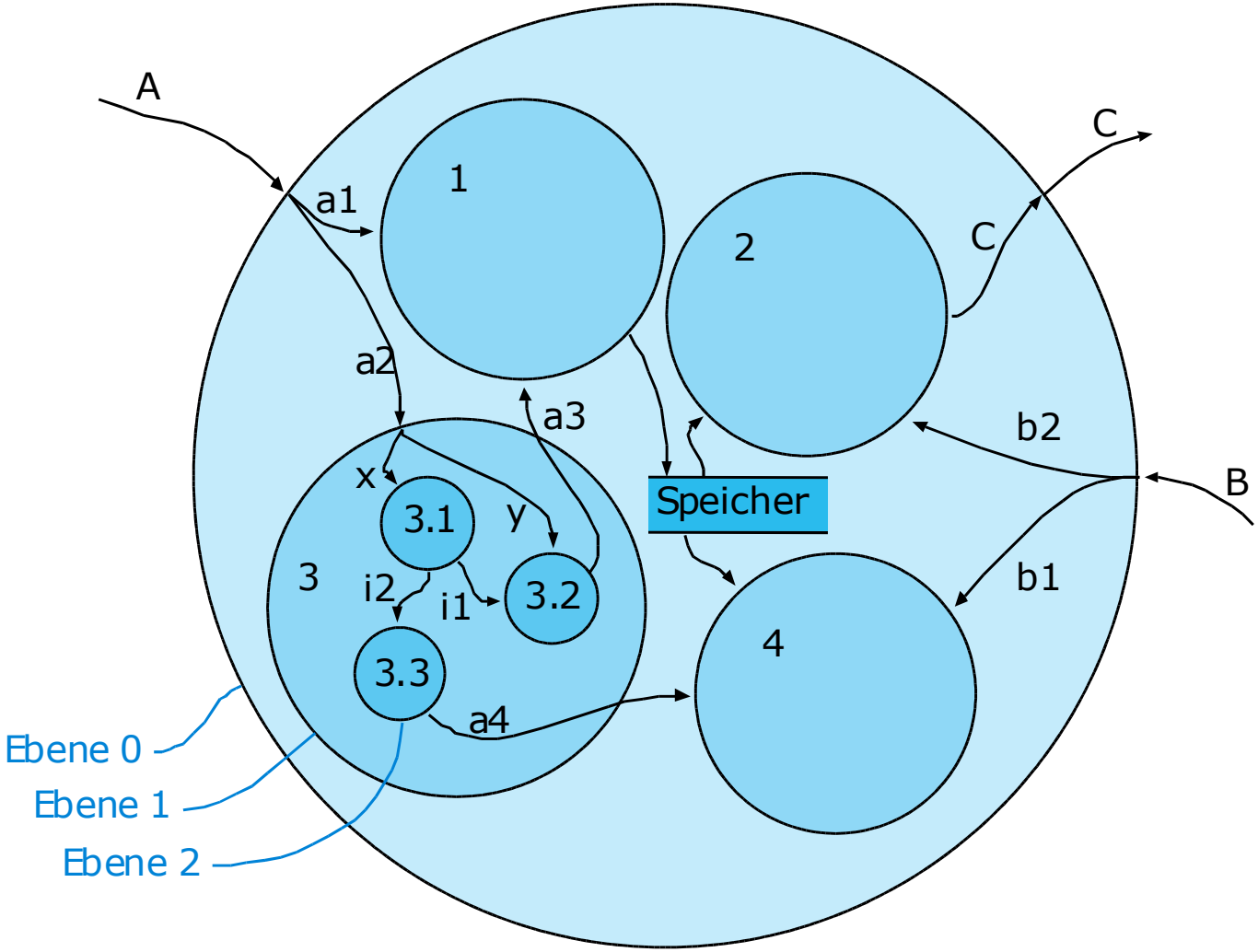
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Darstellungsmöglichkeiten von Datenflüssen

Symbolik	Erklärung
	A fließt von links nach rechts
	A wird in beide Zweige kopiert
	A teilt sich in die Komponenten B und C. Impliziert, dass A keine weiteren Komponenten hat. DD: A=B+C
	A entsteht aus den Komponenten B und C. Impliziert, dass A keine weiteren Komponenten hat. DD: A=B+C
	A fließt beide Wege entlang des Pfeils (meistens bei schreibendem und lesendem Speicherzugriff)
 Kurzform für: 	X, Y und Z fließen getrennt entlang des Pfeils (von Werkzeugen i.A. nicht unterstützt)

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Substituierte Hierarchie

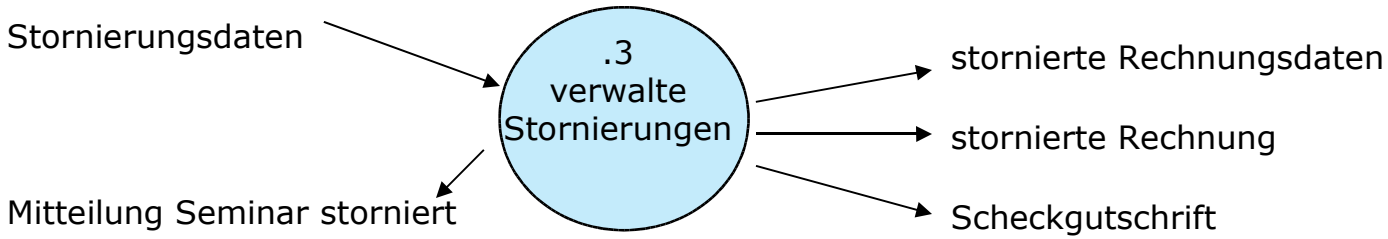


UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Mini-Spezifikationen

- Jeder Prozess, der nicht weiter verfeinert wird, muss durch eine MiniSpec beschrieben werden.
- Jede MiniSpec muss beschreiben, wie die Eingaben in die Ausgaben transformiert werden.
- Eine MiniSpec darf keine Implementierungsvorschriften enthalten.
- MiniSpecs können Pseudocode, ET oder Entscheidungsbäume sein.

- Beispiel: MiniSpec von Prozess 4.3.3 der »Seminarorganisation«




```
Stornierungsdaten aus der Datenbank lesen;
An alle Teilnehmer die Mitteilung Seminar storniert /F110/ versenden
if Teilnehmer hat Rechnung bereits gezahlt
then stornierte Rechnung und Scheckgutschrift mitversenden
end if
Stornierte Rechnungsdaten in Speicher Rechnungsdatensätze eintragen.
```

Methodik

2. Festlegung der Schnittstellen zur Umwelt.

4. Identifizierung aller Eingabe- und Ausgabedatenflüsse von den Schnittstellen zum Prozess 0. Vorläufige Datenflussnamen wählen.

6. Prozesse und Funktionen ermitteln, die Eingaben in Ausgaben transformieren. Vorläufige Prozessnamen wählen!

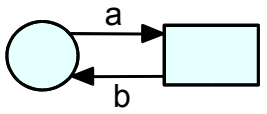
8. Anzahl und Art der Speicher ermitteln, evtl. ER-Modell erstellen

10. Verfeinern der ermittelten Datenflüsse und

12. Zuordnung zu Prozessen und speichern der Datenflüsse und Speicher im DD

Schnittstellen

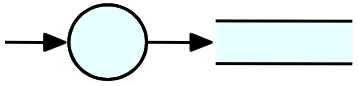
Datenflüsse



Aufteilung in Prozesse

Speicher, u.U.ER-Modell

Datenflüsse verfeinern



Methodik

7. Überarbeiten des Modells. Dabei ist zu beachten:
   - Initialisierung und Terminierung ignorieren
   - Kontrollflüsse entfernen
   - Triviale Fehlermeldungen weglassen
   - Sorgfältige und eindeutige Namenswahl der Datenflüsse. Evtl. neue Strukturierung des Systems
   - Eindeutige Beschriftung aller Prozesse. Der Prozessname soll alle Aktionen des Prozesses erfassen. Evtl. neue Strukturierung des Systems.
2. Ausgehend von der konsolidierten Modellierung des Kontextdiagramms und des Diagramms 0 schrittweise Erarbeitung weiterer Ebenen

4. MiniSpecs für alle nicht weiter verfeinerten Prozesse erstellen

10. Verfeinerung beenden, wenn jeder Prozess überschaubar ist. Nicht unnötig tief verfeinern!

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Hinweise zur Methodik

- Kontextdiagramm und Diagramm 0 können zunächst als ein Diagramm entwickelt werden.
- Erst zum Schluss wird aus diesem Diagramm das Kontextdiagramm abstrahiert.
- Kontextdiagramm und Diagramm 0 sind die entscheidenden Diagramme.
- Sie sollten zunächst interaktiv ohne CASE-Werkzeug im Team entwickelt werden.
- Bei umfangreichen Problemen zwei getrennte Teams einsetzen und die Ergebnisse diskutieren.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Qualitätssicherung

- Vielfältige Analysen zur Qualitätssicherung können durchgeführt werden.
- Jedes der Basiskonzepte kann separat überprüft werden.
- Syntax wird durch CASE-Werkzeuge geprüft.
- Semantik kann mit einer Checkliste überprüft werden.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Bewertung der Strukturierten Analyse

- Vorteile
  - Geschickte Kombination bewährter Basiskonzepte
  - Durch hierarchisch gegliederte DFD Verbesserung der Übersichtlichkeit
  - Viele analytische QS-Möglichkeiten durch Quervergleiche der in den Basiskonzepten beschriebenen Aspekte
  - Leicht erlernbar
  - Erlaubt eine top-down-Einarbeitung in ein System
  - Guter Zusammenhang zu ER-Modellen über Speicher herstellbar.

- Nachteile
  - Schnittstellen können nicht verfeinert werden
    - Bei umfangreichen Schnittstellen gibt es dann Darstellungsprobleme
  - Speicher können nicht verfeinert werden
    - Ausweg: Globale Datenbasis verwenden, Verfeinerung durch ein ER-Modell beschreiben
  - Es entsteht ein Strukturbruch, wenn ein SA-Modell in einen datenabstraktionsorientierten Entwurf transformiert werden soll.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Zusammenfassung

Die **Strukturierte Analyse** (SA, **structured analysis**) besteht aus einem **Hierarchiemodell**, das die einzelnen DFD als Baum anordnet. Wurzel des Baumes ist das **Kontextdiagramm**. Blätter des Baumes sind DFD, die nicht weiter verfeinert werden können.

Prozesse dieser DFD werden durch **Mini-Spezifikationen** (**MiniSpecs**) beschrieben. Innerhalb eines SA-Modells, d. h. von der Baumwurzel bis zu den Baumblättern, muss die **Datenintegrität** (**balancing**) sichergestellt sein, d. h., die DFD müssen zwischen Kind- und Elterndiagramm ausbalanciert sein.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Literatur

- Balzert H., Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag,Heidelberg, 2000.

- DeMarco T. Structured Analysis and System Specification, Yourdon Press, Englewood Cliffs, 1978

- Gane C., Sarson T., Stuctured Systems Analysis: Tools and Techniques, Prentice-Hall, Englewood Cliffs, 1979

- McMenamin S. M., Palemer J. F., Essential Systems Analysis, Yourdon Press, Englewood Cliffs, 1984

- Weinberg V., Structured Analysis, Yourdon Press, Englewood Cliffs, 1978

Überblick LE 15

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

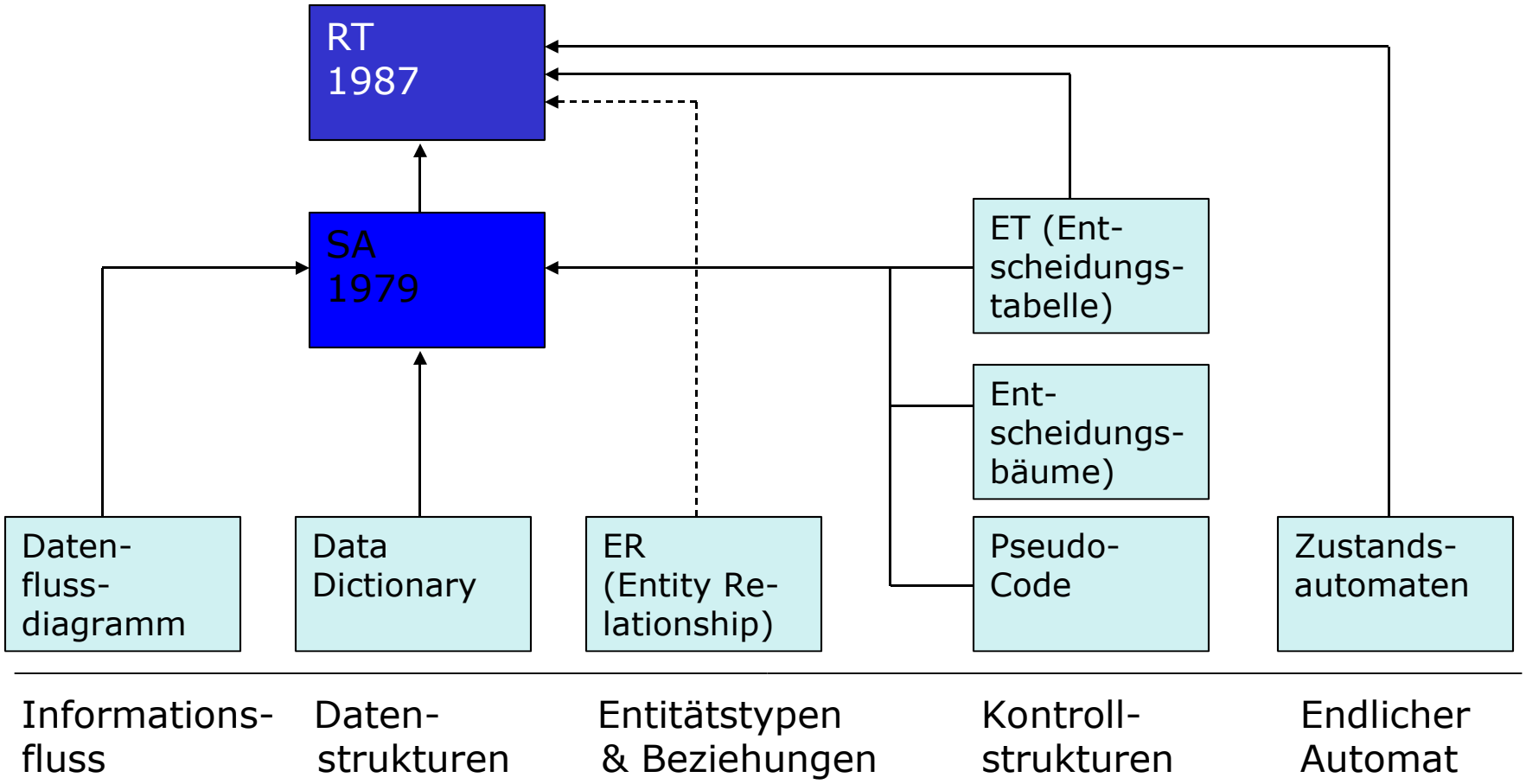
LE 14: Strukturierte Analyse

LE 15: Strukturierte Analyse/Real Time Analysis

- Vergleich SA und SA/RT
- Datenflüsse vs. Kontrollflüsse
- Zeitspezifikation
- Wertung

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

SA/RT und seine Basiskonzepte



Legende:
A ⟶ B: A ist in B enthalten
A ⇢ B: A ist implizit in B enthalten

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Vergleich SA und SA/RT

- Defizite SA
  - SA verzichtet bewusst auf die Beschreibung der Initialisierung und Terminierung eines Systems
  - Zeitanforderungen können in SA nicht festgelegt werden
- RT erweitert SA so, dass ...
  - ereignisgesteuerte Systeme mit
  - Zeitanforderungen und
  - komplexen Prozessaktivierungen
  - modelliert werden können.
- SA/RT
  - ist eine Methodenklasse.
- SA/RT ist
  - Industriestandard.
  - Erste Werkzeuge waren schon ein Jahr nach Vorstellung der Methode (1987) verfügbar.
- Echtzeitbegriff in SA/RT
  - Es ist nur die Festlegung externer Zeitanforderungen möglich
    - Diese werden keiner Analyse unterzogen
    - Sie werden nur dokumentiert
  - Zeitanforderungen müssen keine Echtzeitanforderungen sein
  - Prinzipiell können auch kommerzielle Anwendungssysteme modelliert werden.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Datenflüsse vs. Kontrollflüsse

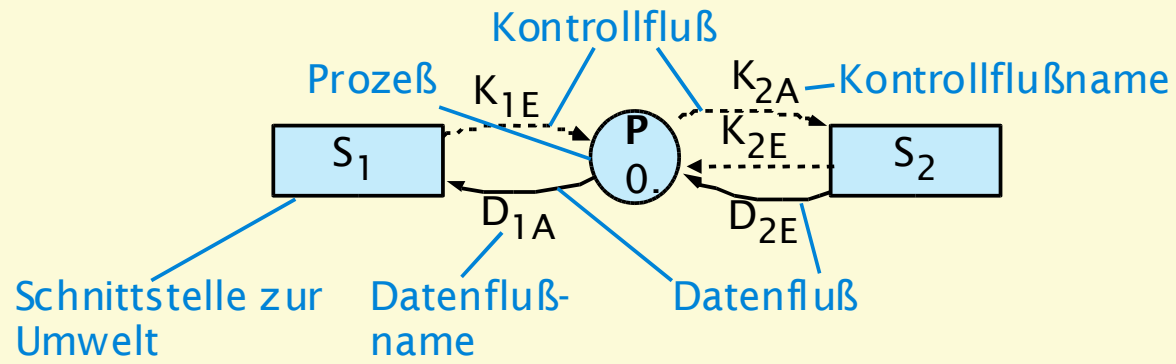
- Datenflüsse (aus SA)
  - Werden in den Prozessen verarbeitet bzw. transformiert.
  - Werden als durchgezogene Linien dargestellt.
  - Elementare Datenflüsse sind meistens kontinuierliche Signale.
  - Kontinuierliches Signal kann einen beliebigen Wert innerhalb eines Wertebereichs annehmen
    - Beispiel: 0..250 km/h
  - Ein Datenfluss kann auch ein diskretes Signal sein, wenn es selbst verarbeitet wird.
- Kontrollflüsse (neu in RT)
  - Steuern die Verarbeitung
  - Werden als gestrichelte Linien dargestellt
  - Sind immer diskrete Signale
    - Nehmen eine endliche Anzahl bekannter Werte an
      - Beispiel: Druckknopf = [gedrückt | nicht gedrückt]
  - Werden wie Datenflüsse im DD definiert (aber als Kontrollflüsse gekennzeichnet)
    - DD heißt dann Requirements Dictionary (RD)
  - Unterscheidung zwischen Kontroll- und Datenflüssen oft schwierig.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

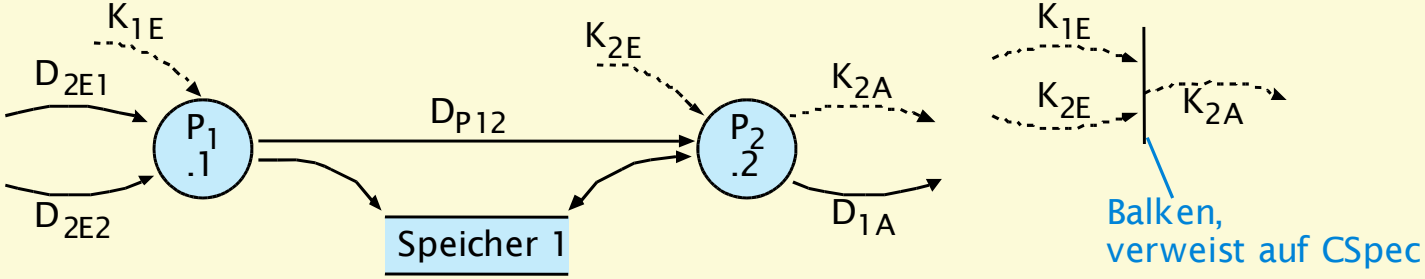
Flussdiagramm

- Flussdiagramm (FD):
  - Enthält sowohl Daten- als auch Kontrollflüsse

Beispiel:



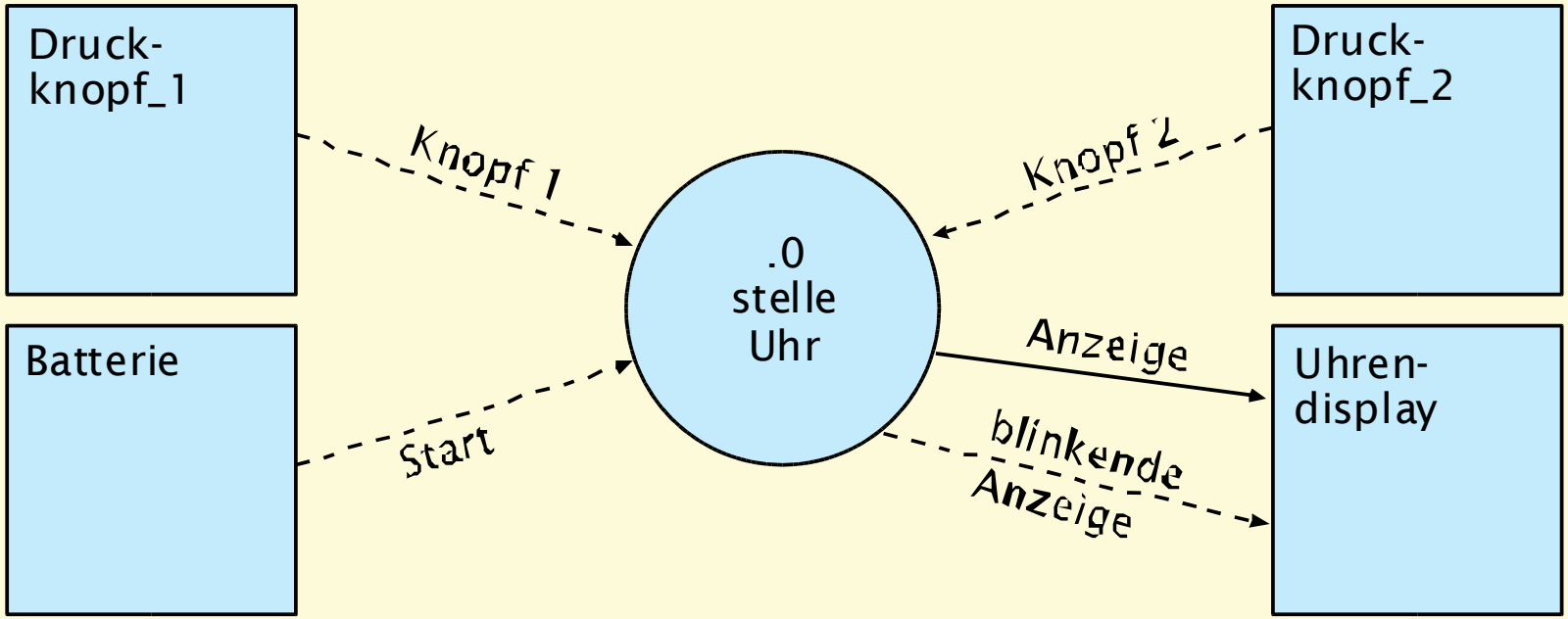
FD0 = Verfeinerung des Kontextdiagramms



RD:
$D_{2E} = D_{2E1} + D_{2E2}$

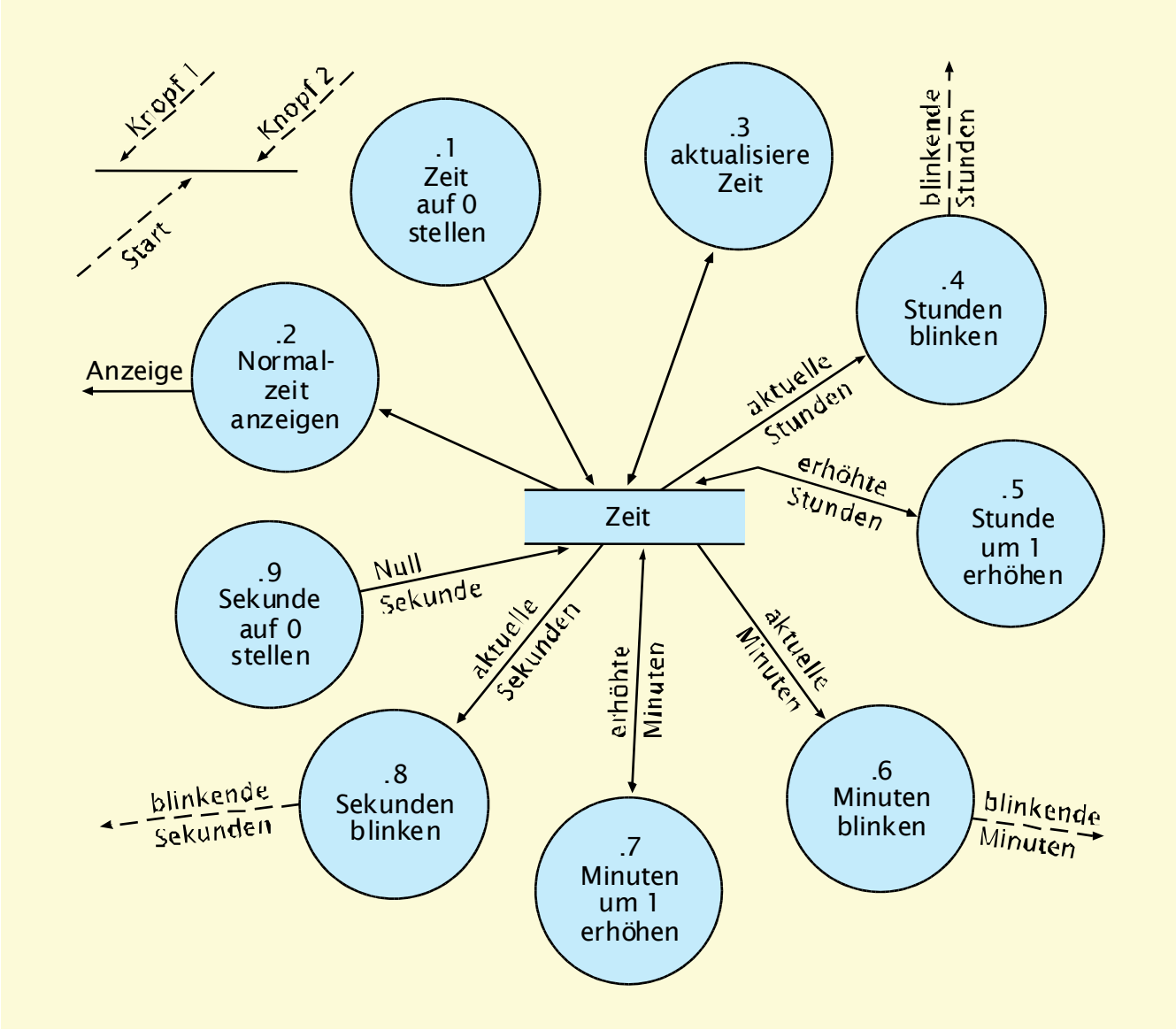
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

RT-Kontextdiagramm – Beispiel: Digitaluhr



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Flussdiagramm 0 (FD0) Beispiel: Digitaluhr



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Requirements Dictionary

- RD (Kontrollflüsse blau) - Beispiel Digitaluhr

Knopf 1	**= [gedrückt | nicht gedrückt]**
Knopf 2	**= [gedrückt | nicht gedrückt]**
Start	**= [Strom vorhanden | Strom nicht vorhanden]**
Anzeige	= Stunden + Minuten + Sekunden
Blinkende Anzeige	**= [blinkende Stunden | blinkende Minuten | blinkende Sekunden]**
Zeit	= Stunden + Minuten + Sekunden
aktuelle Stunden	= Stunden
erhoehte Stunden	= Stunden
aktuelle Minuten	= Minuten
erhoehte Minuten	= Minuten
aktuelle Sekunden	= Sekunden
Nullsekunde	= Sekunden
Stunden	= 0 .. 23
Minuten	= 0 .. 59
Sekunden	= 0 .. 59
blinkende Stunden	**= 0 .. 23**
blinkende Minuten	**= 0 .. 59**
blinkende Sekunden	**= 0 .. 59**

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Kontrollspezifikationen

- Ermöglicht es über ET und/oder Zustandsautomaten, Prozesse eines FD zu aktivieren und zu deaktivieren.

- Jedem FD kann eine Kontrollspezifikation (CSpec) zugeordnet werden:
  - Enthält ET und/oder
  - Zustandsautomaten
- Zusammenhang zwischen CSpec und FD:
  - Wird durch eine Balkennotation (bar) im FD beschrieben
  - Kontrollflüsse, die Eingangsgrößen für eine CSpec sind, werden mit der Pfeilspitze zum Balken gezeichnet
  - Kontrollflüsse, die eine CSpec verlassen, führen vom Balken weg.

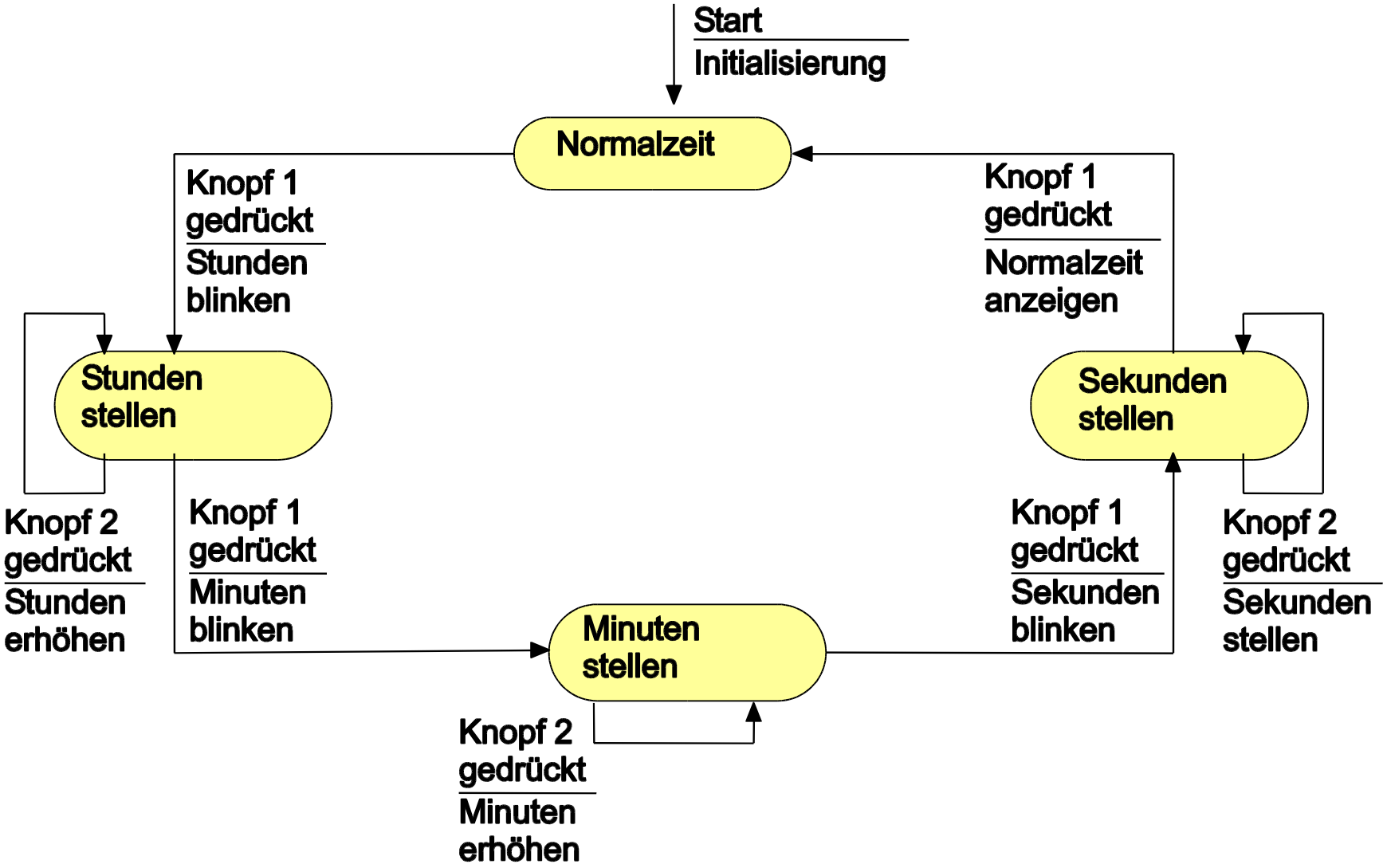
Kontrollspezifikationen – Beispiel: Digitaluhr

- Nach Einlegen der Batterie:
  - Zeitspeicher soll mit der Zeit 0 initialisiert werden, d.h. zuerst muss der Prozess „Zeit auf 0 stellen“ aktiviert werden.
- Ist die 0 im Speicher eingetragen, dann kann der Prozess „Normalzeit anzeigen“ aktiviert werden.
- Wird Knopf 1 gedrückt, erfolgt ein Übergang in der Zustand „Stunden stellen“

- Die Kontrollspezifikation (CSpec 0) besteht aus einem Zustandsautomat und einer ET.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Zustandsautomat von CSpec 0



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Entscheidungstabelle von CSpec 0

- Prozessaktivierungstabelle

Kontroll-aktionen	Prozesse							
	.2	.1	.5	.4	.7	.6	.9	.8
	Normalzeit anzeigen	Zeit auf Null stellen	Stunde um 1 erhöhen	Stunden blinken	Minuten um 1 erhöhen	Minuten blinken	Sekunden auf Null stellen	Sekunden blinken
Initialisierung	2	1	0	0	0	0	0	0
Stunden erhöhen	0	0	1	2	0	0	0	0
Minuten erhöhen	0	0	0	0	1	2	0	0
Sekunden stellen	0	0	0	0	0	0	1	2

Zeitspezifikation

- Zeitanforderungen
  - Spezifikation von externen Zeitanforderungen
  - Beziehen sich auf die Signale der Schnittstellen
  - Zwei Arten von Zeitspezifikationen:
    - Wiederholungszyklen
      - Für externe, elementare Ausgabesignale
      - Wiederholungszyklen werden im RD (Requirement Dictionaries) durch das Attribut Rate festgelegt
    - Eingabe-Ausgabe-Antwortzeiten
      - Legen den erlaubten Antwortzeitbereich für jedes Eingabeereignis und das resultierende Ausgabeereignis fest
- Regeln für Zeitspezifikationstabelle
  - Eingabeereignisse treten außerhalb des Systems ein
  - Ausgabeereignisse sind Aktionen, die vom System ausgeführt werden
  - Ein- und Ausgabeereignisse sind im RD definiert
  - Es darf kein Signal oder Wert in der Zeitspezifikationstabelle erscheinen, das nicht im RD enthalten ist
  - Jedes externe Signal, das im RD aufgeführt ist, sollte in der Zeitspezifikationstabelle erscheinen, selbst wenn die Zeit unkritisch ist.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Zeitspezifikation – Beispiel Digitaluhr

- Wiederholungszyklen
  - Für externe, elementare Ausgabesignale
  - Wiederholungszyklen werden im RD durch das Attribut *Rate* festgelegt

```
Stunden = 0..23          Rate: Alle 100 msec
Minuten = 0..59          Rate: Alle 100 msec
Sekunden= 0..59          Rate: Alle 100 msec
```

- Eingabe-Ausgabe-Antwortzeiten
  - Legen den erlaubten Antwortzeitbereich für jedes Eingabeereignis und das resultierende Ausgabeereignis fest
  - Zeitspezifikationstabelle

Eingabesignal	Ereignis	Ausgabesignal	Ereignis	Antwortzeit
Knopf1	gedrückt	blinkende Stunden	Zahl anzeigen	<100 msec
Knopf1	gedrückt	blinkende Minuten	Zahl anzeigen	<100 msec
Knopf1	gedrückt	blinkende Sekunden	Zahl anzeigen	<100 msec
Knopf2	gedrückt	blinkende Stunden	Zahl anzeigen	<200 msec
Knopf2	gedrückt	blinkende Minuten	Zahl anzeigen	<200 msec
Knopf2	gedrückt	blinkende Sekunden	Zahl anzeigen	<200 msec.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Bewertung von SA/RT

- Vorteile
  - Gut geeignet zur Modellierung ereignisgesteuerter Systeme
  - Komplexe Steuerungszusammenhänge können beschrieben werden
  - Zur Zeit keine vergleichbare bessere Methode zur Modellierung dieser Anwendungsklasse verfügbar
  - Gute Werkzeugunterstützung

- Nachteile
  - Schwieriger als SA zu erlernen und zu verstehen
  - Wird leicht unübersichtlich, da eine CSpec aus mehreren Seiten bestehen kann.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Zusammenfassung

Real-Time Analysis (**RT**) erweitert SA um die Möglichkeit, Prozesse zu aktivieren und zu deaktivieren. Außerdem können **Zeitspezifikationen** beschrieben werden. Um dies zu ermöglichen, gibt es neben den Datenflüssen auch **Kontrollflüsse**, die Ereignisse repräsentieren. DFD werden zu Flussdiagrammen verallgemeinert, die zusätzlich Kontrollflüsse enthalten können. Die Prozess-Steuerung der Prozesse, die sich auf einem Flussdiagramm befinden, erfolgt durch einen zugeordnete **Kontrollflussspezifikation** (**Cspec**). Die in der Kontrollflussspezifikation verwendeten Ein- und Ausgabensignale werden durch eine **Balken-Notation** (**bar**) im zugeordneten Flussdiagramm aufgeführt. Elementare Prozesse werden durch eine **Prozess-Spezifikation** (**PSpec**) beschrieben (anderer Name für Mini-Spezifikationen). Alle Flüsse und Speicher werden im **Requirements Dictionary** (**RD**) definiert (anderer Name für Data Dictionary).

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Literatur

- Balzert H., Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag,Heidelberg, 2000.

- Hatley D. J., Pirbhai I. A. , Strategies for real-time system specification, Dorset House Publishing, New York, 1987

- Ward P. T., Mellor S. J., Structured Development for Real-Time Systems, Volume 1: Introduction and Tools, Yourdon Press, 1985

- Ward P. T., Mellor S. J., Structured Development for Real-Time Systems, Volume 2: Essential Modelling Techniques, Yourdon Press, 1985

- Ward P. T., Mellor S. J., Structured Development for Real-Time Systems, Volume 3: Implementation Modelling Techniques, Yourdon Press, 1986