

Übersicht

- Einführung und begriffliche Grundlagen
- Formalismen und Konzepte
- Strukturierte Methoden: SA & Co
 - Strukturierte Methoden – allgemein
 - SA
 - MSA
 - SA/RT
 - Strukturierte Methoden – Werkzeuge
 - Schlussbemerkungen
- Objektorientierte Methoden: UML 2
- Zusammenfassung und Ausblick

Lernziele

- Die Grundidee strukturierter Methoden beherrschen
- Wissen aus welchen Bestandteilen SA zusammengesetzt ist und wie diese zusammenhängen
- Die jeweiligen Prinzipien der Erweiterungen von SA kennen
- Strukturierte Methoden hinsichtlich ihrer Vor- und Nachteile, insbesondere hinsichtlich praktischer Brauchbarkeit, beurteilen können

Merkmale

- Gemeinsame Grundlage: Beschreibung der funktionalen Zerlegung (in Prozesse) durch *Datenflussdiagramme* ➡
- Methodisches Grundprinzip: Top-down-Vorgehen mit *schrittweiser Verfeinerung* ➡
(Verfeinerung von Prozessen durch Datenflussdiagramme)
- Modellierung der nicht verfeinerten Bestandteile von Datenflussdiagrammen
 - Prozesse: *Prozessspezifikation* ➡
 - Daten: *Datenlexikon* ➡
- Erweiterungen zur Modellierung weiterer Systemaspekte
 - Statische Struktur (System- und Komponentenstrukturen)
 - Dynamisches Verhalten (Kontrollaspekte, Steuerung)

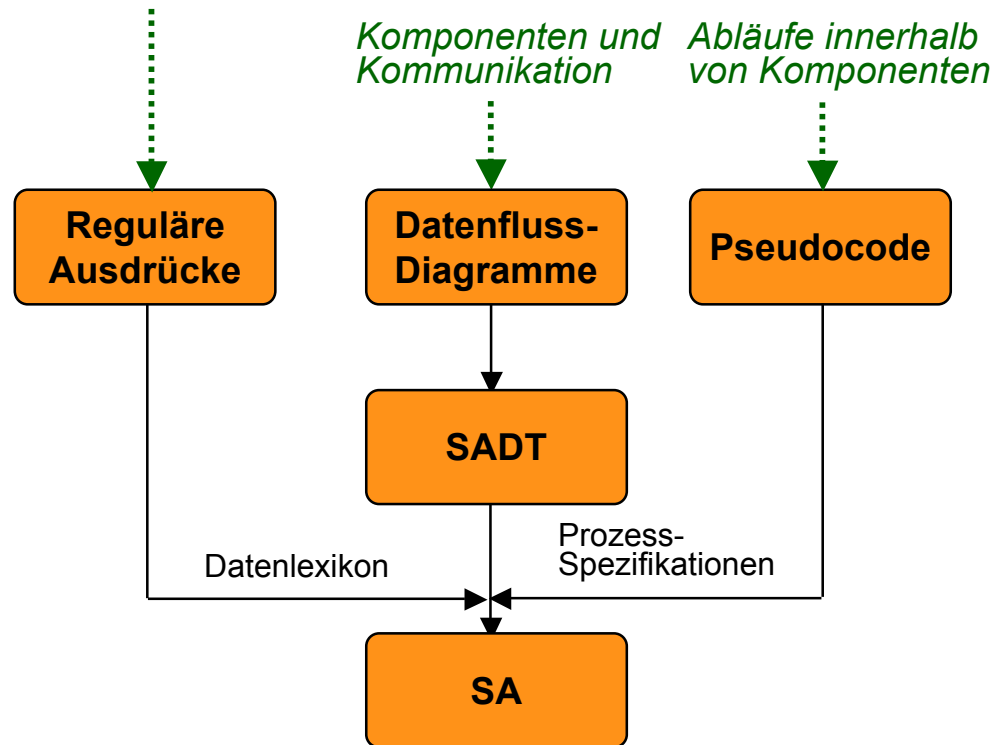
Spezielle Methoden

- SADT „Structured Analysis“, Tom de Marco, Ende 70er
- SA „Modern Structured Analysis“, Ed Yourdon, Ende 80er
- MSA „Structured Analysis / Real Time“, Hatley/Pirbhai, Ende 80er
- SA/RT



System- und Komponentenstrukturen

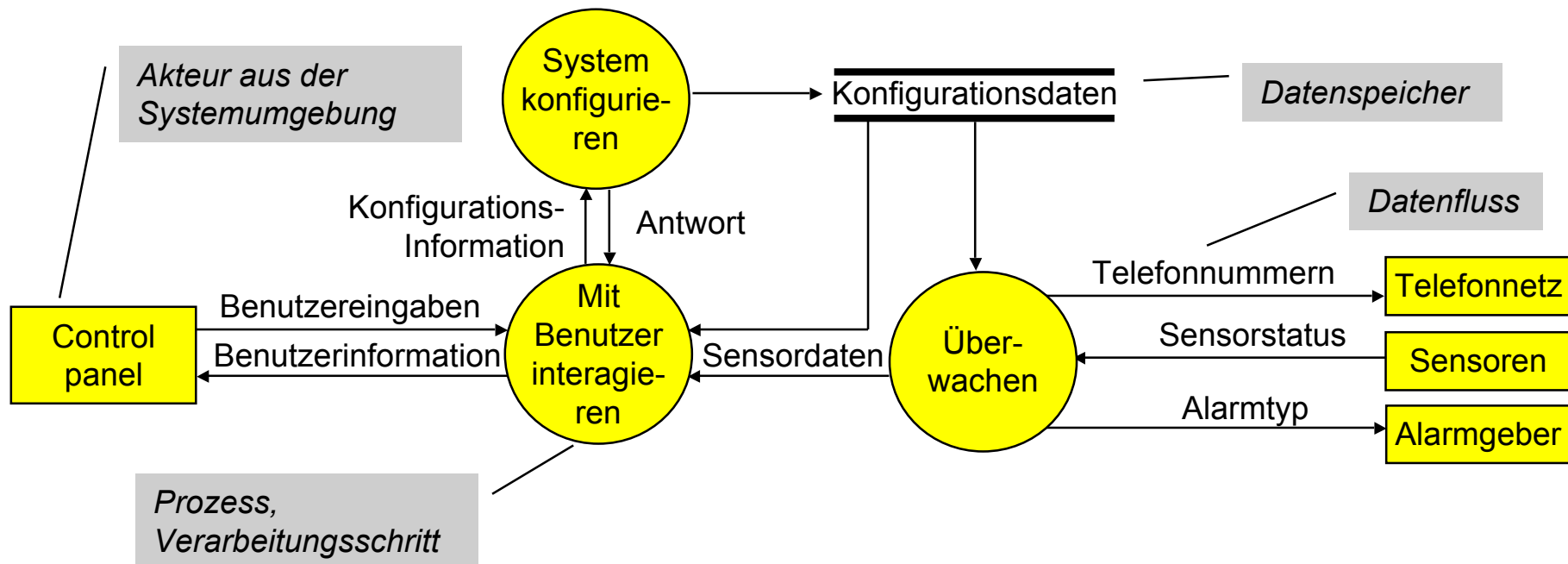
Funktionales Verhalten



Datenflussdiagramm (DFD, Synonyme: Bubble Chart, Prozessmodell, Funktionsmodell)

- Endlicher, zusammenhängender, knoten- und kantenmarkierter Graph mit 3 Arten von Knoten (  )

Beispiel (Datenflussdiagramm der Alarmanlage, siehe Grundlagenteil)



Rolle der Bestandteile von Datenflussdiagrammen

- **Endknoten, Begrenzer**



- Modellieren Umgebungsteile, die mit dem System kommunizieren, als „black boxes“

- **Prozesse**



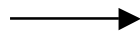
- Beschreiben einzelne Funktionalität eines Systems
- Transformieren Eingangsdaten in Ausgangsdaten („Datentransformationsprozesse“)

- **Datenspeicher, Lager**



- Stellen statische oder temporär abgelegte Daten (eines bestimmten Typs) dar
- Sind über Datenflüsse mit Prozessen verbunden
 - Fluss in ein Lager: schreibender Zugriff
 - Fluss aus einem Lager: lesender Zugriff

- **Datenflüsse**

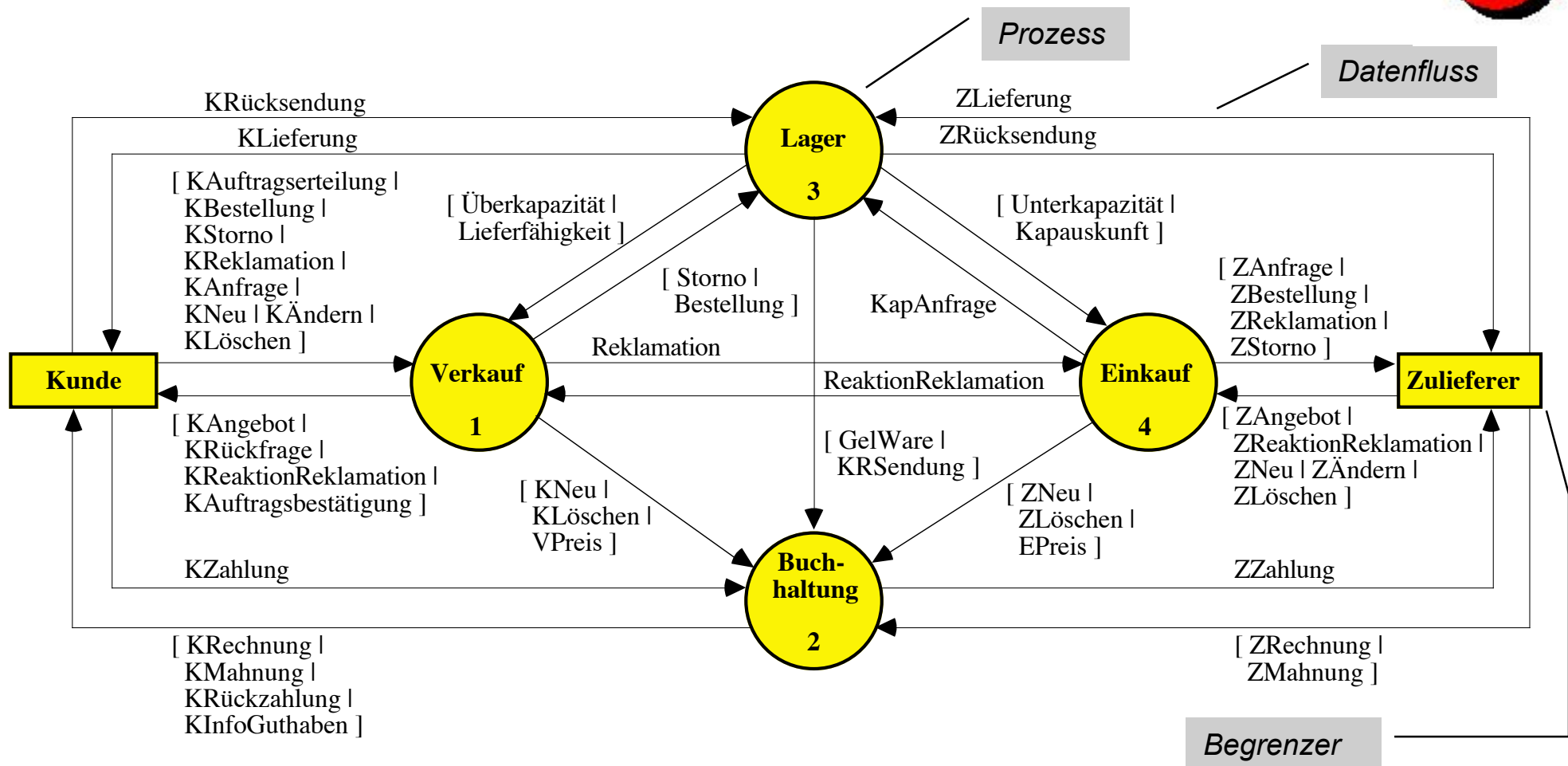


- Verbinden Begrenzer, Datenspeicher und Prozesse miteinander (1 Ende muss Prozess sein)
- Beschreiben Transportwege von Daten (eines bestimmten Typs)
- Sind stets (mit dem Typ der zu transportierenden Daten) benannt
(Ausnahme: Flüsse von und zu Datenspeichern)





Beispiel (Datenflussdiagramm der Vertriebsorganisation)



Strukturierte Darstellung von Datenflussdiagrammen

- Hierarchische Zerlegung (*Verfeinerung* ) von Prozessen in neue DFD
 - Mit neuen Prozessen, (internen) Datenflüssen und möglicherweise neuen Speichern
 - Evtl. „Aufspaltung“ von Datenflüssen
 - Aber: keine neuen Verbindungen zur „Außenwelt“

- „oberstes“ Diagramm: **Kontextdiagramm**

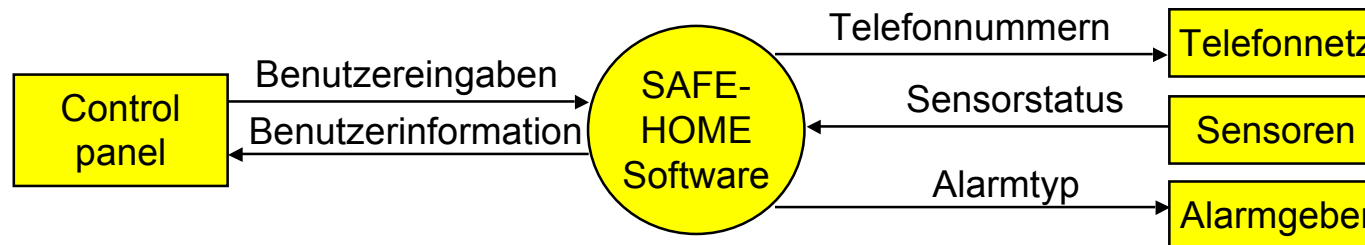
Wurzel des „Verfeinerungsbaums“

- Nicht weiter verfeinerter Prozess: **elementarer Prozess**

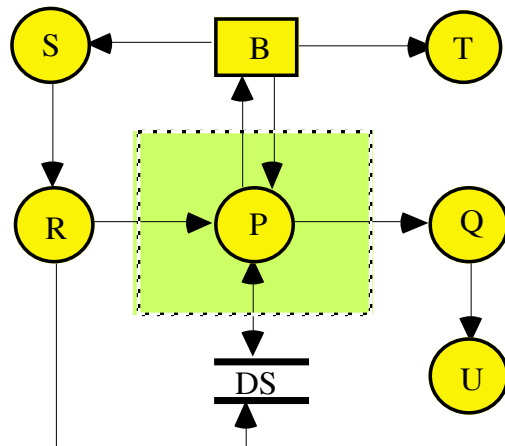
Blatt des „Verfeinerungsbaums“

- Zusammenhang zwischen Diagrammen verschiedener Verfeinerungsebenen durch einfaches Nummerierungsschema (Dezimal-Klassifikation)
- Konsistenz der Verfeinerungsebenen durch entsprechende Regeln (Grundidee: Schnittstellen müssen erhalten bleiben)

Beispiel (Kontextdiagramm für die Alarmanlage)



Prinzip der Verfeinerung



Ausschnitt aus Diagramm D

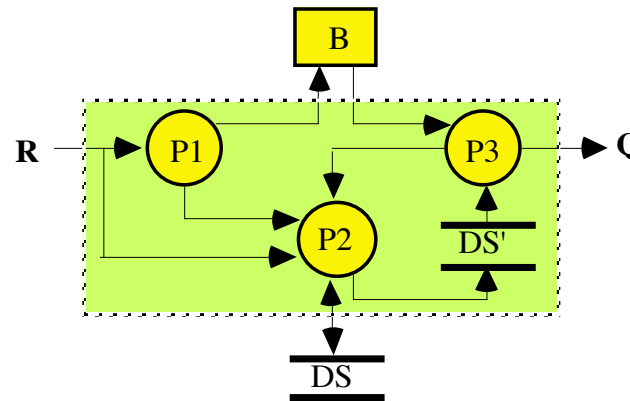


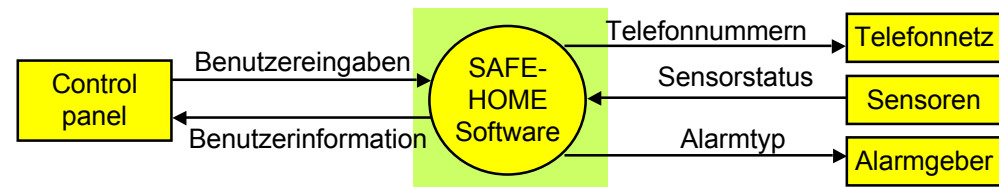
Diagramm D': Verfeinerung von Prozeß P

Verfeinerungsprozess

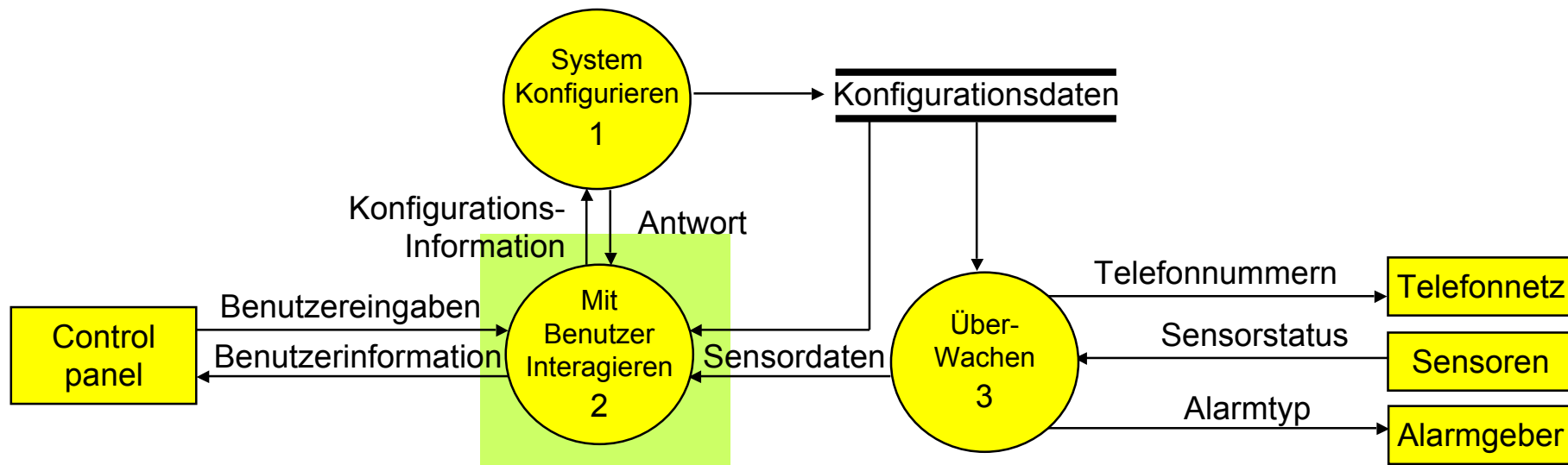
- Nur sehr „weiche“ Kriterien für das Ende der Verfeinerung (in der einschlägigen Literatur)
 - Prozess ist durch Prozessspezifikation von ca. 1 Seite darstellbar
 - Anzahl ein- und ausgehender Datenflüsse eines Prozesses ist hinreichend klein
 - Prozess lässt sich durch einfache Ein-/Ausgaberation beschreiben
- Bemerkungen
 - Diagramme sollten nicht zu groß sein (7 ± 2 Prozesse pro Diagramm)
 - Anzahl der Verfeinerungsstufen sollte nicht zu groß sein (≤ 8)

$$7^8 = 5\,764\,801 !$$

Beispiel (1. Verfeinerung der Alarmanlage)

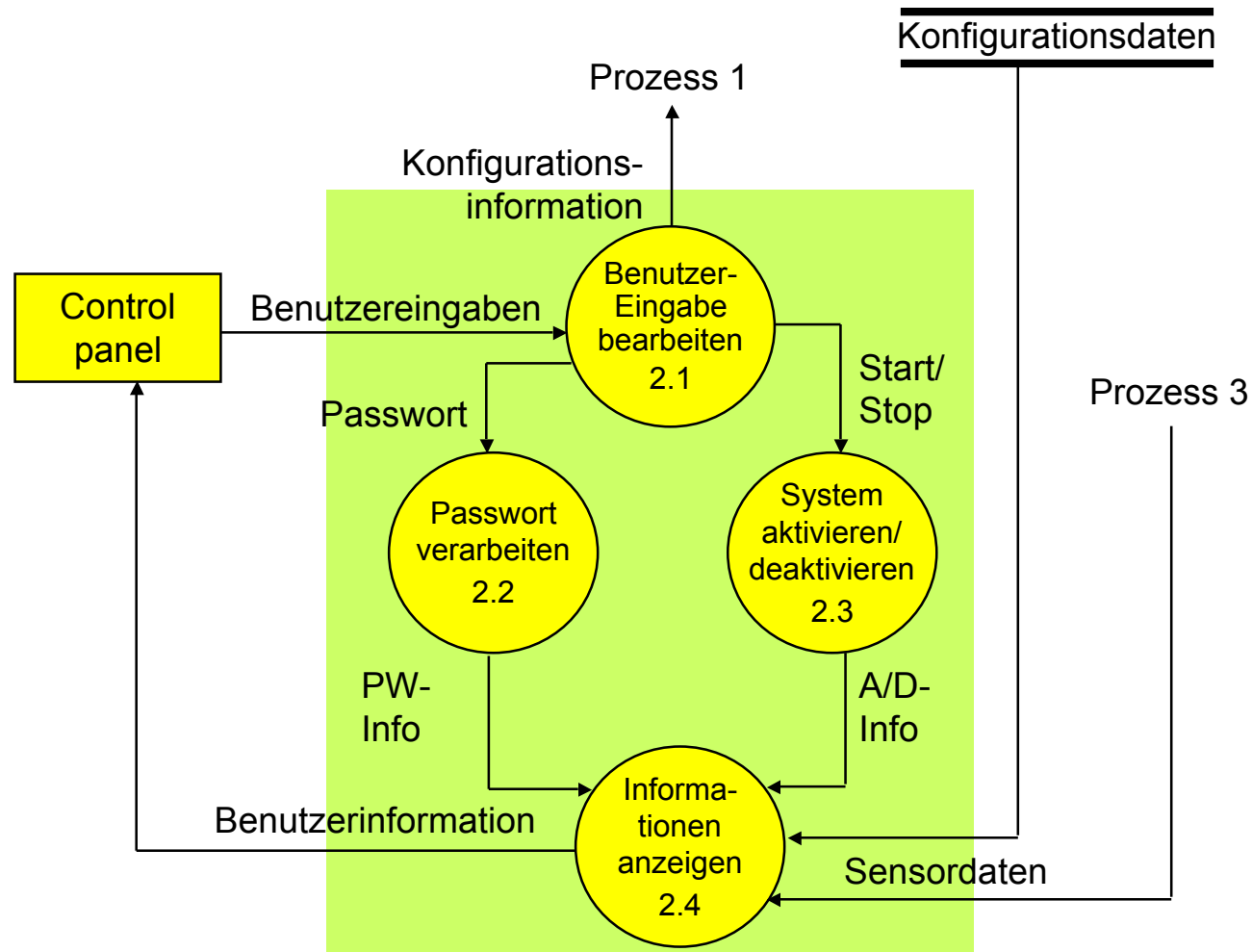


Kontextdiagramm



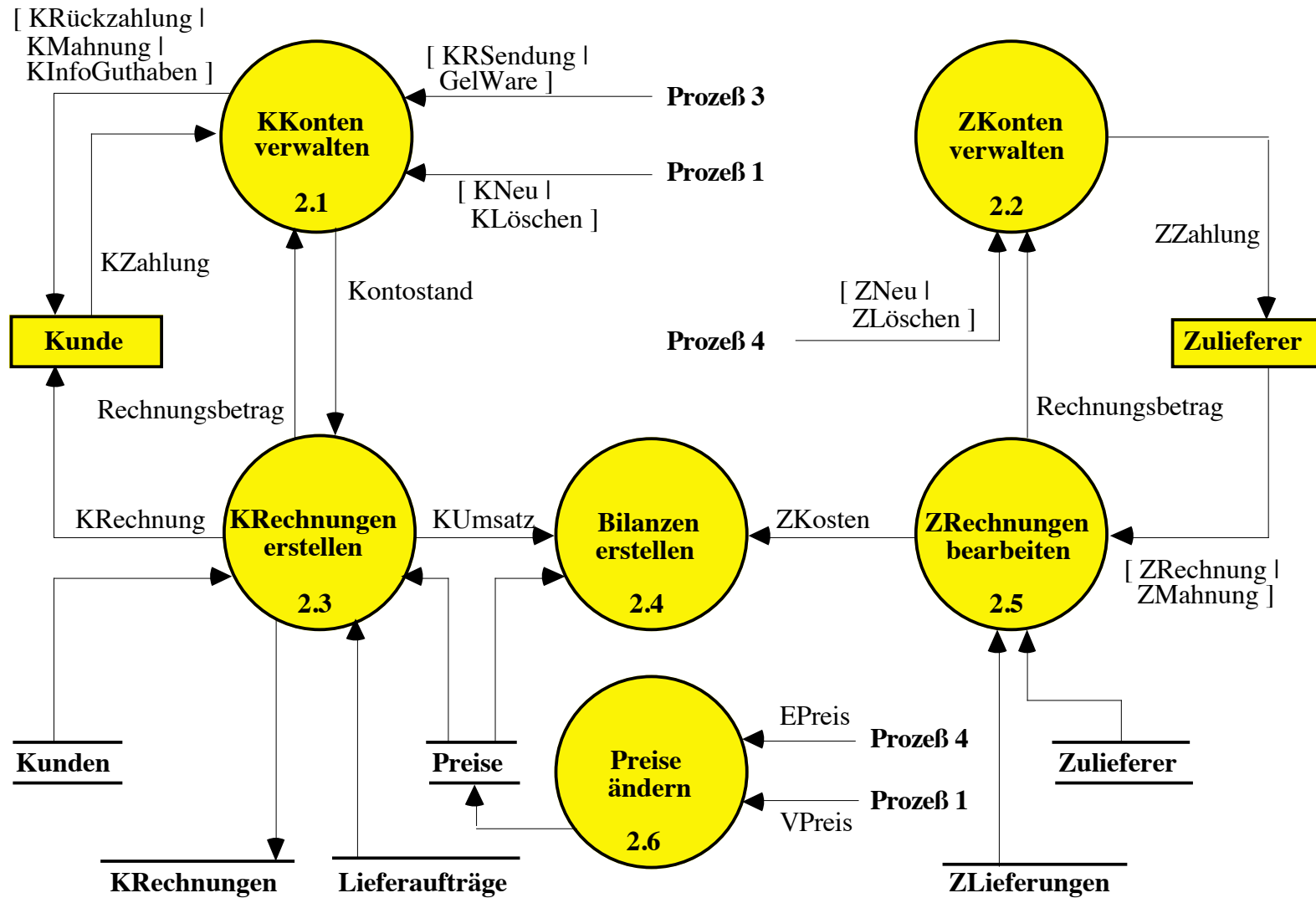
1. Verfeinerung

Beispiel (Alarmanlage: Verfeinerung des Prozesses „mit Benutzer interagieren“)



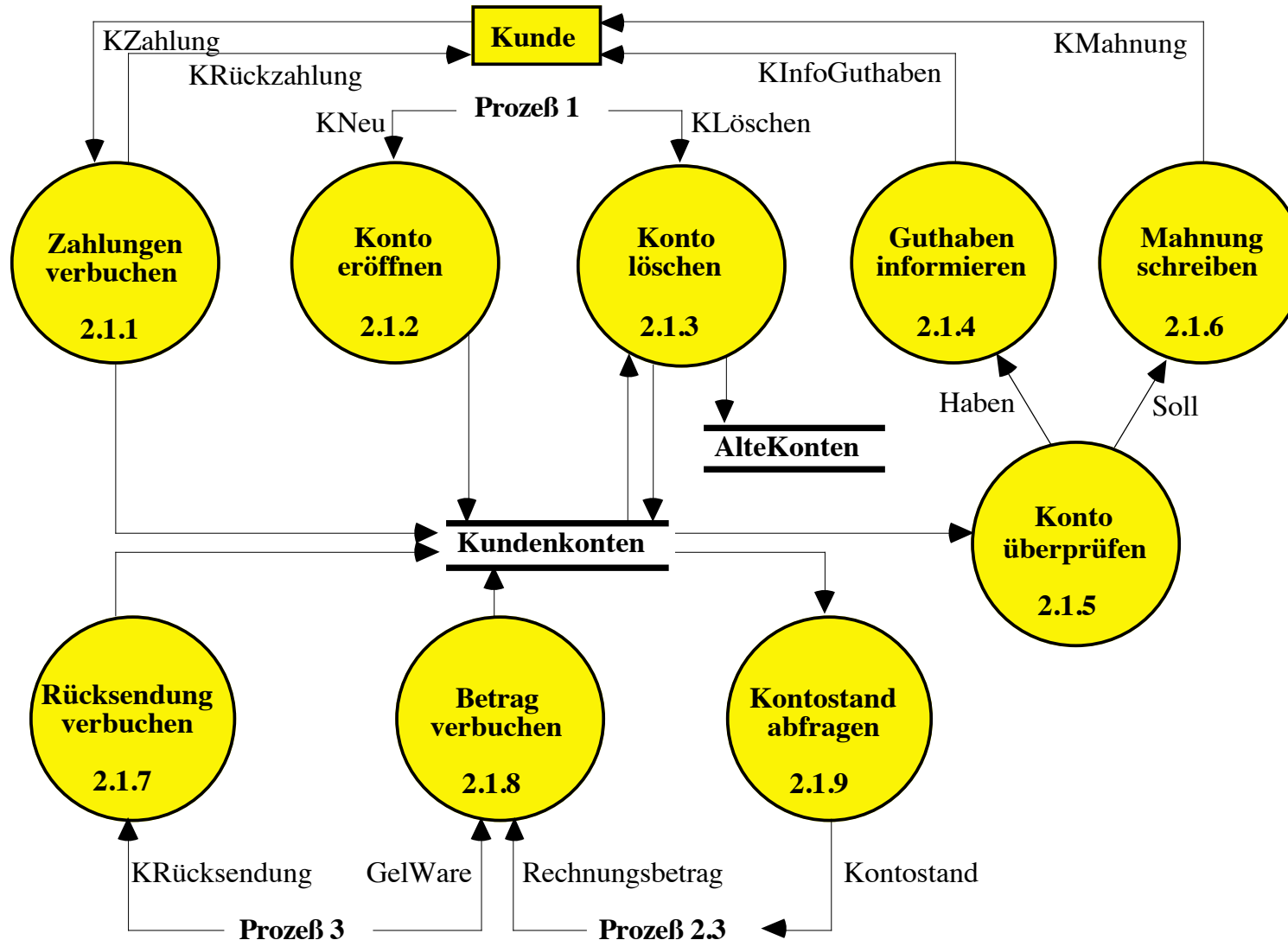


Beispiel: Verfeinerung des Prozesses „Buchhaltung“





Beispiel: Verfeinerung des Prozesses „KKonten verwalten“



Datenlexikon

- Beschreibung der Struktur („Datentyp“) von Datenspeichern und -flüssen
- Definition durch Text oder reguläre Ausdrücke gebildet mit

=	ist definiert als
+	Sequenz („und“)
	Auswahl („oder“)
[]	Klammerung einer Auswahl
{ }	beliebige Wiederholung
{ } ⁿ	n-malige Wiederholung
()	optionale Angabe
@	Schlüssel
**	Kommentar

*In eigener Zeile; auch möglich: * ... * in regulären Ausdrücken*

Beispiel (Auszug aus dem Datenlexikon für die Alarmanlage)

Konfigurationsinformation = Passwort + System-ID + Aktivierungs-Status

Passwort = {[Buchstabe | Ziffer | Sonderzeichen]}⁸

System-ID = Buchstabe + {Ziffer}⁴

Aktivierungs-Status = [alle | {Zone + Ziffer}*]

Sonderzeichen = ** Alle auf der Tastatur vorhandenen Zeichen mit Ausnahme von # und *;



Beispiel: Auszug aus dem Datenlexikon für die Vertriebsorganisation



Adresse =	Straße + Hausnummer + (Länderkennung + -) + PLZ + Ort + (Land)	Kundennummer =	[1 2 3 4 5 6 7 8 9] ⁸
Anmerkungen =	{[Buchstabe - ' ']} ¹⁰⁰ ** Textfelder der Länge 100; ** ' ' symbolisiert das Leerzeichen ** Bemerkungen zum Zahlungsverhalten, ** Beschwerden u.ä.	Länderkennung =	[D F NL GB AU CH I ...]
Datum =	Tag + Monat + Jahr	Land =	Name
Firma =	@Firmenname + Adresse	letzte Bestellung =	Datum
Firmenname =	Buchstabe + {[Buchstabe - . & ' ']} ²⁹	Monat =	[Januar Februar ... Dezember]
Geburtsdatum =	Datum	Nachname =	Name
Hausnummer =	[1 2 ... 999]	Name =	Buchstabe + {[Buchstabe - ' ']} ¹⁹
Jahr =	[1950 1951 ... 2010]	Ort =	Name
Kontostand =	[S H] + {Ziffer} ⁷ + . + Ziffer + Ziffer ** Soll oder Haben	Person =	@Nachname + @Vorname + Adresse + @Geburtsdatum
Kunde =	[Person Firma] + @Kundennummer + letzte Bestellung + Anmerkungen	PLZ =	{Ziffer} ⁵
Kundenkonto =	@Kundennummer + Kontostand	Straße =	Name
		Tag =	[1 2 ... 31]
		Vorname =	Name
		:	



Prozessspezifikation (Synonyme: Pspec, Transformationsbeschreibung, Kurzinfo)

- Betrifft nur elementare Prozesse — *d.h. ein Prozess hat entweder Verfeinerung oder Pspec*
- Beschreibt, wie elementarer Prozess eingehende Daten in Ausgaben transformiert
- Bevorzugte Formalismen: Text, Pseudocode, Vor-/Nachbedingungen

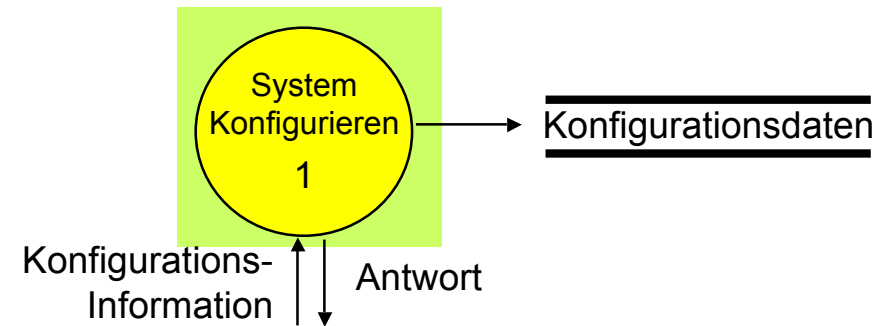
Beispiel (Prozessspezifikation in Pseudocode)

PROCESS **System Konfigurieren**
INPUT **Konfigurationsinformation**
OUTPUT OK | Konfigurierungsfehler
BEGIN

 prüfe Passwort und System-ID;
 IF Passwort oder System-ID nicht akzeptiert
 THEN gib Konfigurierungsfehler zurück
 ELSE Speichere Aktivierungsstatus in **Konfigurationsdaten**;
 gib OK zurück

 END-IF

END





Beispiele: Prozessspezifikation

```
PROCESS Bestand-prüfen
INPUT KBestellung
OUTPUT (Fehlteile?, Fehlliste)
BEGIN
    lege leere Fehlliste an;
    FOR alle Artikel A aus KBestellung DO
        IF Bestellanzahl(A) > Lageranzahl(A)
            THEN füge (A, Bestellanzahl(A) – Lageranzahl(A)) zu Fehlliste hinzu
        END-IF
    END-FOR;
    RETURN (Fehlliste ≠ [], Fehlliste)
END
```

Pseudocode

```
PROCESS Bestand-prüfen
INPUT KBestellung
OUTPUT (Fehlteile?, Fehlliste)
PRECONDITION
    keine
POSTCONDITION
    (Fehlteile? = FALSE ∧  $\forall$  Artikel A: A ∈ KBestellung  $\Rightarrow$  Bestellanzahl(A) ≤ Lageranzahl(A)) ∨
    (Fehlteile? = TRUE ∧
         $\forall$  Artikel A: (A ∈ KBestellung ∧ Bestellanzahl(A) > Lageranzahl(A))  $\Rightarrow$ 
            (A, Bestellanzahl(A) – Lageranzahl(A)) ∈ Fehlliste)
```

Vor- und Nachbedingungen

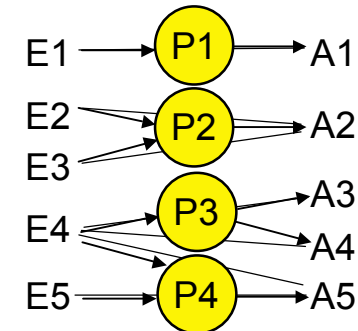


Entwicklung eines SA-Modells (häufige Vorgehensweise: „*middle-out-Strategie*“)

- Entwicklung des initialen Datenflussdiagramms

(„*event partitioning*“)

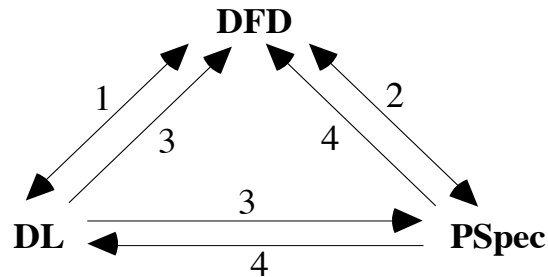
- Ermittlung der Systemschnittstellen
(ein-/ausgehende Flüsse zu Begrenzern)
- Verbindung „zusammengehörender“
Ein-/Ausgabe-Flüsse durch Prozesse



- Ableitung höherer Verfeinerungsebenen (inkl. Kontextdiagramm) aus dem initialen Diagramm durch Abstraktion zusammengehörender Teildigramme in Prozesse („*upward leveling*“)
- Entwicklung der DFD-Hierarchie durch schrittweise Verfeinerung („*downward leveling*“)
- Definition elementarer Prozesse durch Prozessspezifikationen
- Parallel zur DFD-Entwicklung:
Definition von Flüssen und Speichern im Datenlexikon
- Überprüfung des Modells auf Konsistenz (*Balancierung*) nach vorgegebenen Regeln



Überprüfung der Beschreibung: Balancierung



DFD: Datenflußdiagramm
DL: Datenlexikon
PSpec: Prozeßspezifikation

Bestandteile der Datenflussdiagramme müssen miteinander verträglich sein:

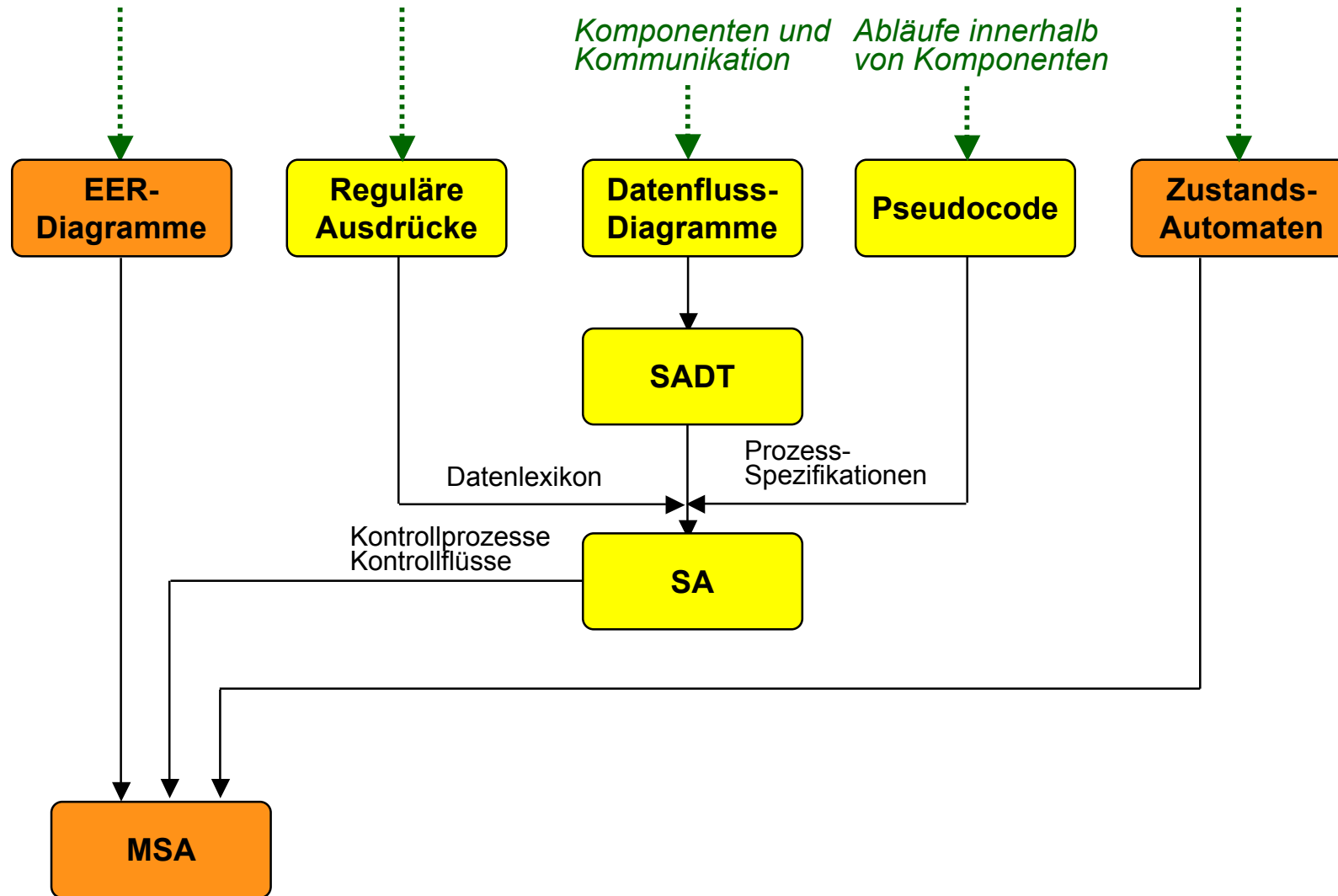
- 1) Jeder Datenfluss und Datenspeicher, der in einem Diagramm vorkommt, muss im DL definiert sein (und umgekehrt)
- 2)
 - Jeder Prozess im DFD hat Verfeinerungs-DFD oder PSpec (aber nicht beides)
 - Jeder PSpec entspricht Prozess in nicht verfeinertem DFD
 - Ein- und Ausgaben (in DFD und PSpec) müssen konsistent sein
- 3) Jeder DL-Eintrag muss referenziert sein durch PSpec, DFD oder DL-Eintrag
- 4) Für jeden Datenbezug in PSpec muss eine der folgenden Bedingungen erfüllt sein
 - passt zu Datenfluss oder Datenspeicher der mit dem mit PSpec assoziiertem Prozess verbunden ist,
 - ist lokaler Term der PSpec
 - ist Komponente eines verbundenen Datenflusses oder Datenspeichers



System- und Komponentenstrukturen

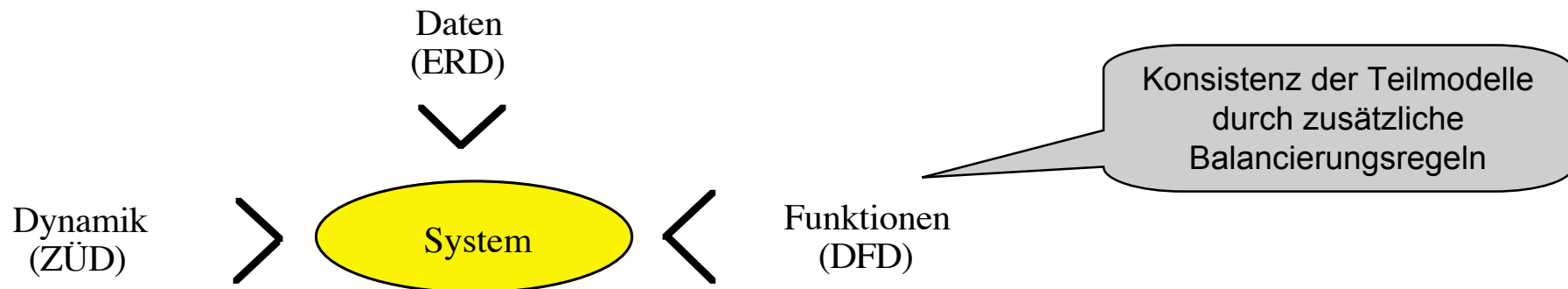
Funktionales Verhalten

Kontrollaspekte, Steuerung



MSA (Modern Structured Analysis, Yourdan 89)

- Logische Weiterentwicklung von SA um Kontroll- und Datenaspekte
- Analyse und Definition eines Problems aus unterschiedlichen Blickwinkeln

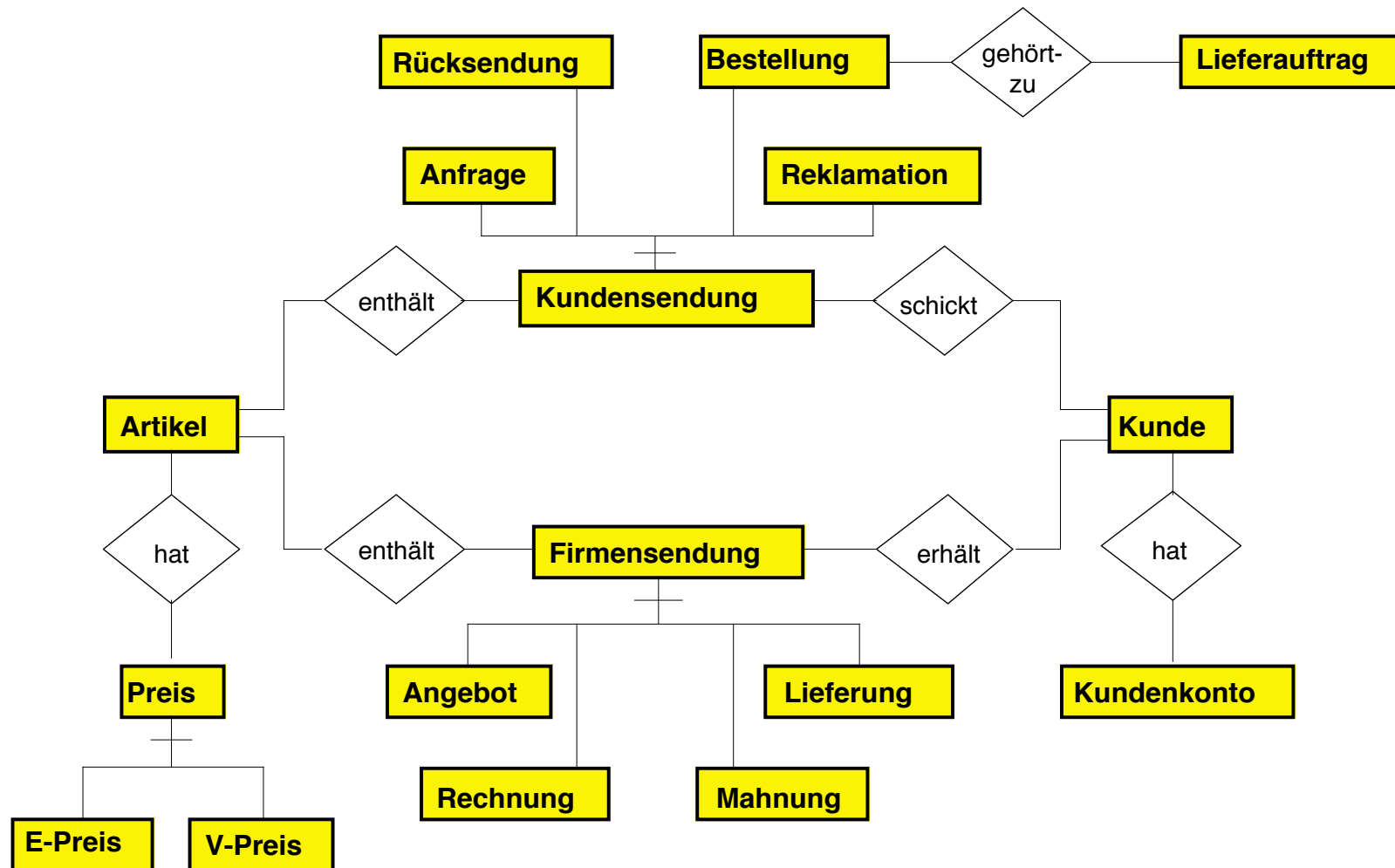


- **DFD** (mit Datenlexikon und Prozessspezifikationen) für funktionale Zerlegung
 - Prozesse sind Daten- oder Kontrollprozesse
 - Schnittstellen sind durch Daten- oder Kontrollflüsse beschrieben
- **ER-Diagramm** (ERD) für Informationsstruktur
 - Relationale Beschreibung von (Beziehungen zwischen) Datenflüssen und Datenspeichern
- **Zustandsübergangsdiagramm** (ZÜD) für Verhaltensbeschreibungen
 - Verhalten von „Kontrollprozessen“ durch Zustandsmaschinen beschrieben



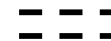
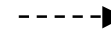


Beispiel: Ausschnitt aus dem ER-Diagramm für die Vertriebsorganisation



Funktionale Modellierung durch Datenflussdiagramme

- I.w. wie bei SA
- Im DFD zusätzlich enthalten
 - Kontrollprozesse (Verhalten über ZÜD definiert, s.u.)
 - Kontrollflüsse (wie Datenflüsse im Datenlexikon definiert)
 - Ereignisspeicher (zur Ablage von Kontrollinformation)



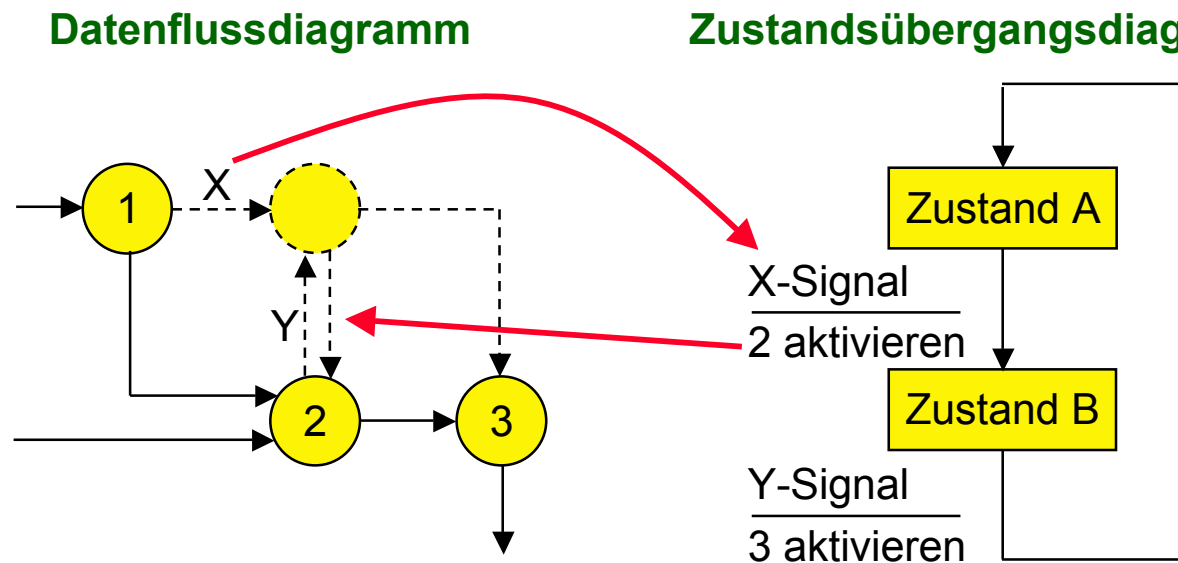
Unterscheidung

*Unterscheidung nur nach inhaltlichen,
nicht nach syntaktischen Kriterien möglich*

- **Datenfluss**: Daten (mit beliebig vielen Werten) oder diskrete Signale, die in einem Datenprozess transformiert werden
- **Kontrollfluss**: diskrete Signale (mit wenigen Werten), die der Steuerung von Prozessen dienen

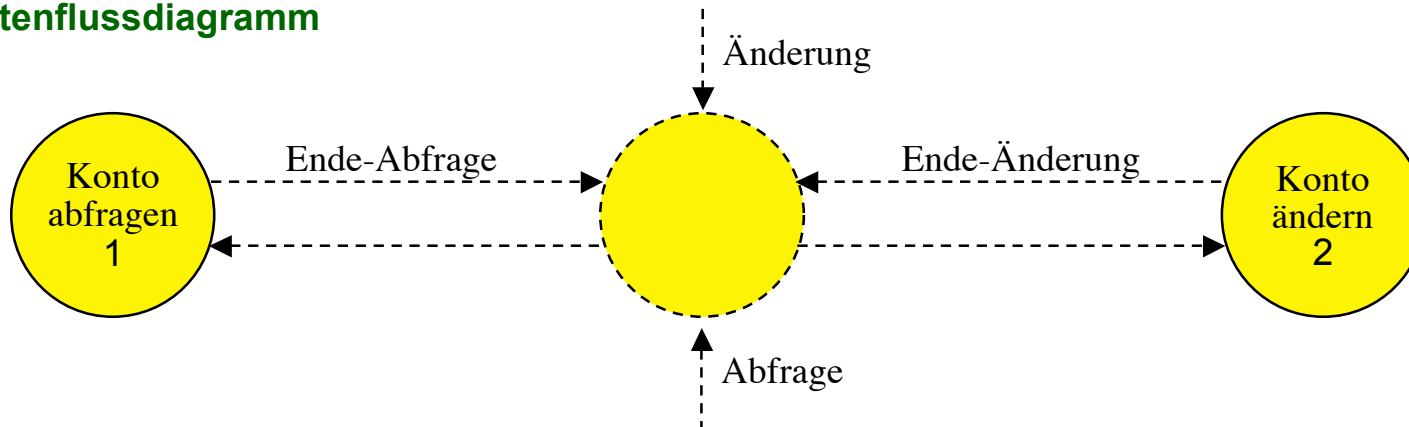
Zusammenhang zwischen Datenfluss- und Zustandsübergangsdiagramm

- Verhalten von Kontrollprozessen wird durch ZÜD beschrieben
- Aufgabe eines *Kontrollprozesses*:
Aktivierung von Datenprozessen in Abhängigkeit vom Systemzustand
- Eingehender Kontrollfluss $\approx$ Bedingung im Zustandsübergangsdiagramm
- Ausgehender Kontrollfluss $\approx$ Aktion im Zustandsübergangsdiagramm

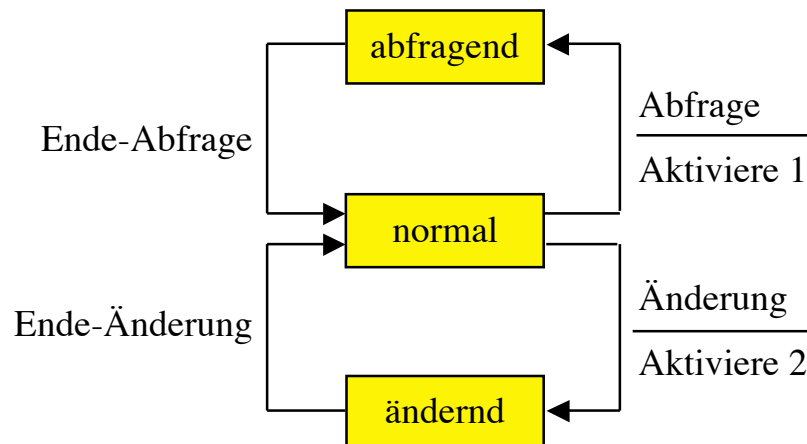


Beispiel: Wechselseitiger Ausschluss von Konto ändern und abfragen

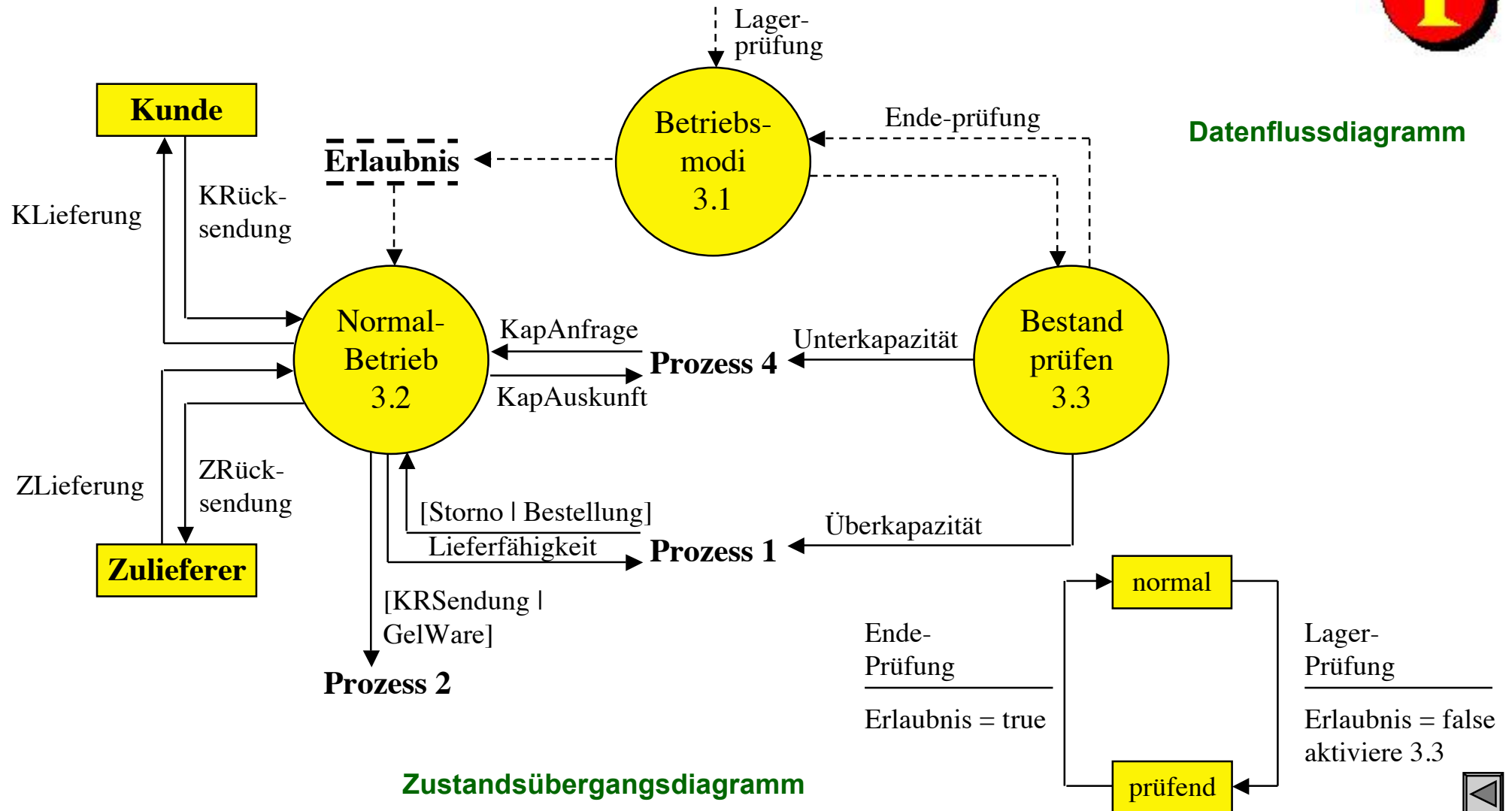
Datenflussdiagramm



Zustandsübergangsdiagramm



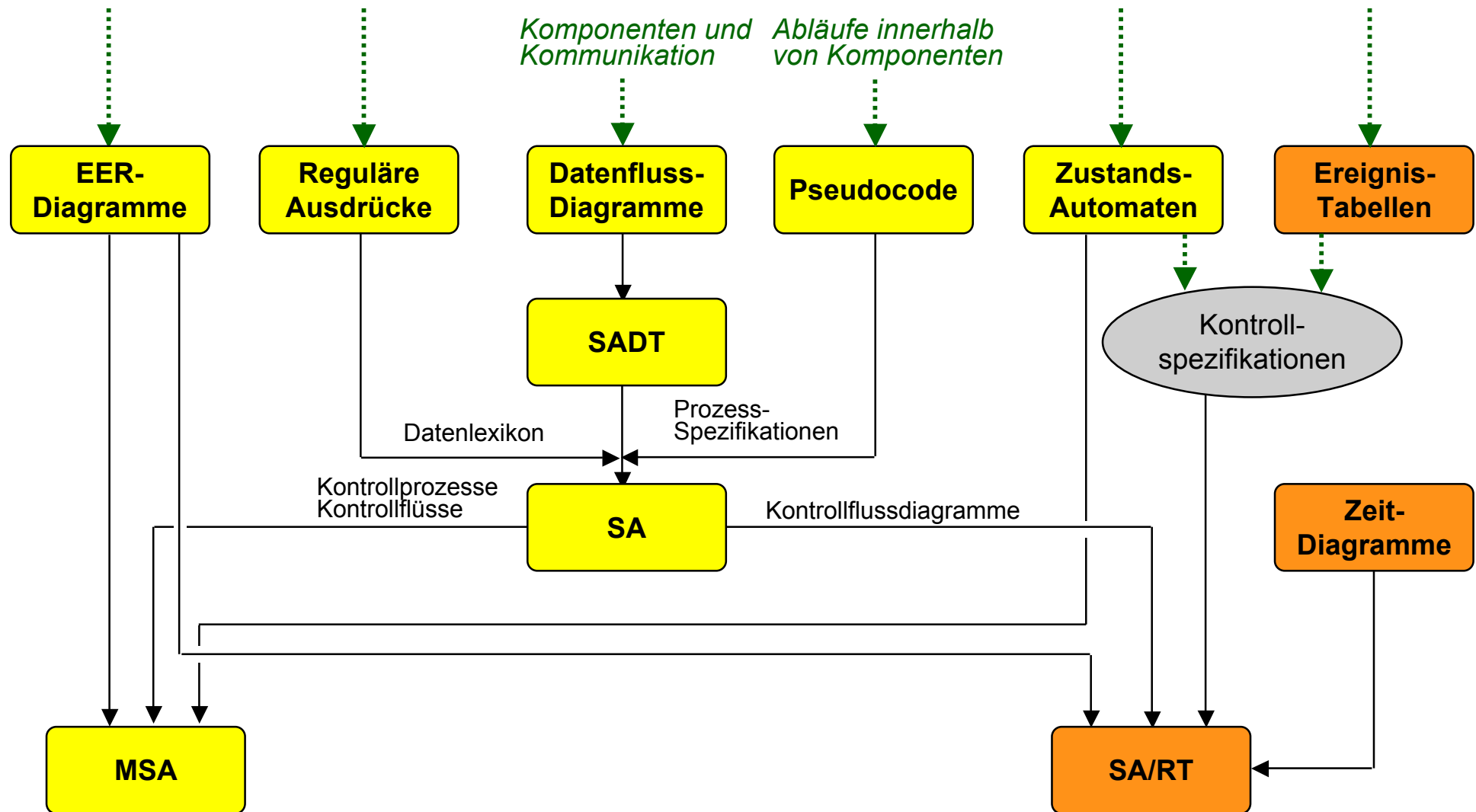
Beispiel: Verfeinerung des Prozesses „Lager“



System- und Komponentenstrukturen

Funktionales Verhalten

Kontrollaspekte, Steuerung

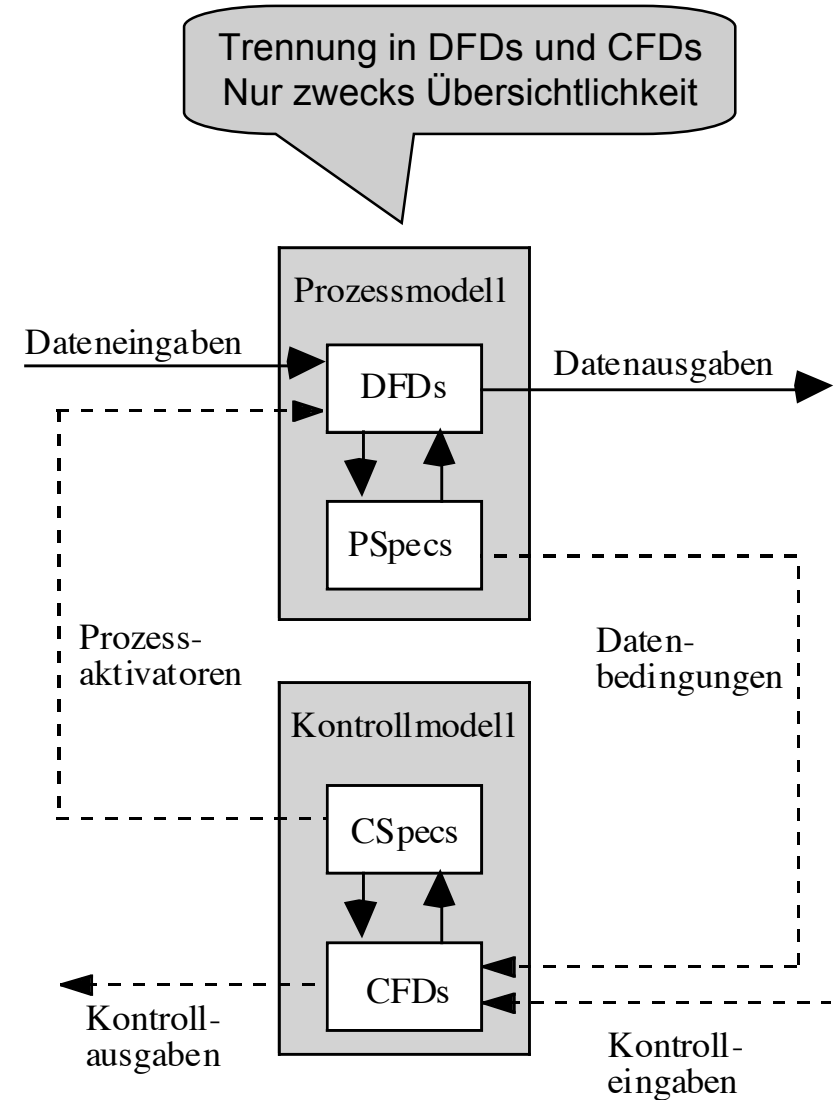


SA/RT

- Logische Weiterentwicklung von SA um
 - Kontrollflüsse und Kontrollprozesse (zur Modellierung ereignisgesteuerter Verarbeitung)
 - Kontrollspezifikationen zur Beschreibung des Verhaltens von Kontrollprozessen
 - Beschreibung von Datenflüssen und Datenspeichern durch Entity-Relationship-Diagramme (ERD)
 - Einbeziehung von Zeitaspekten
- Wie in MSA: Analyse und Definition eines Systems aus den Blickwinkeln
 - Daten (ERD)
 - Funktionen (DFD)
 - Dynamik (ZÜD)
- Verschiedene Ausprägungen
 - Ward, Mellor: MSA
 - Harel: statecharts
 - Hatley, Pirbhai: im folgenden

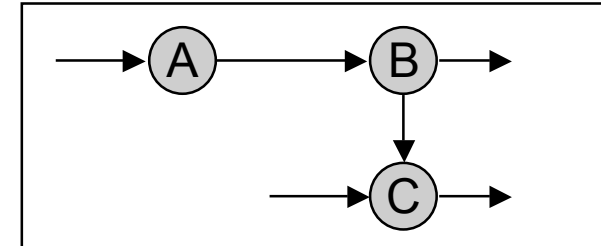
Bestandteile von SA/RT

- (hierarchische) *Datenflussdiagramme* (DFD)
über Datenflüsse verbundene Prozesse
- *Prozessspezifikationen* (Pspec)
Funktionales Verhalten elementarer Prozesse
- (hierarchische) *Kontrollflussdiagramme* (CFD)
über Kontrollflüsse verbundene Prozesse
- *Kontrollspezifikationen* (Cspec)
Definition des Kontrollverhaltens
- *Anforderungslexikon* („requirements dictionary“)
Datenlexikon + Definition von Kontrollflüssen
- *Zeitspezifikationen* (ZS)



DFDs, Datenflüsse und PSpecs

- I.w. wie in SA
- In Pspecs zusätzlich möglich: **Datenbedingungen**
(= Kontrollflüsse, die sich aus Daten ableiten und als Steuerinformation in CFDs eingehen)

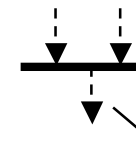
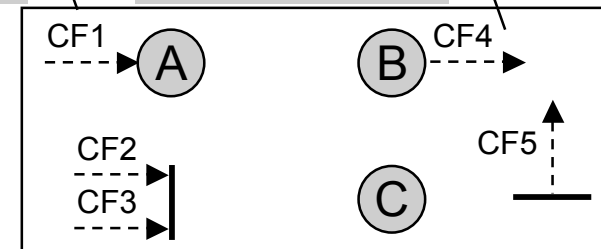


Kontrollflussdiagramme (CFDs)

- Hierarchischer Aufbau wie DFDs
(zu jedem DFD genau ein CFD)
 - Dieselben Prozesse und Speicher, jedoch Kontroll- statt Datenflüsse
 - Evtl. zusätzlich „**Kontrollspeicher**“
 - Darstellung des Kontrollflusses auf verschiedenen Verfeinerungsebenen
- Enthalten auch Kontrollflüsse, die in einem „**Balken**“
(= Ein-/Ausgabeschnittstelle zur Kontrollspezifikation) enden

In Verfeinerung von A verwendet

In Verfeinerung von B erzeugt

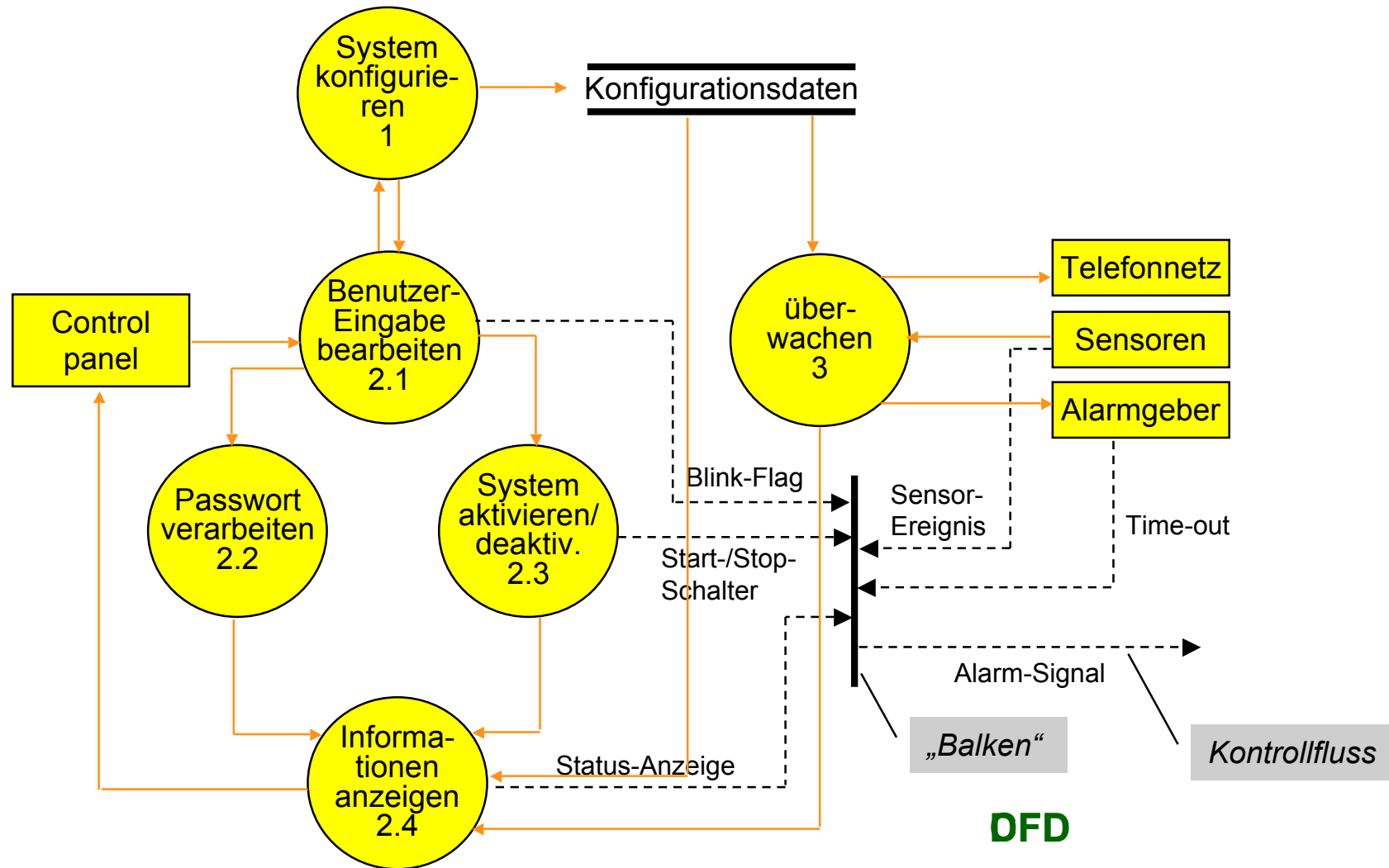


Pro CFD nur ein „Balken“
(der aber „verteilt“ dargestellt werden kann, s.o.)

Zusammenhang zwischen DFDs und CFDs

- Über **Kontrollspezifikationen** ➡
- Auch möglich: DFDs und CFDs integriert

Beispiel (Integriertes Daten- und Kontrollfluss-Diagramm für die Alarmanlage)

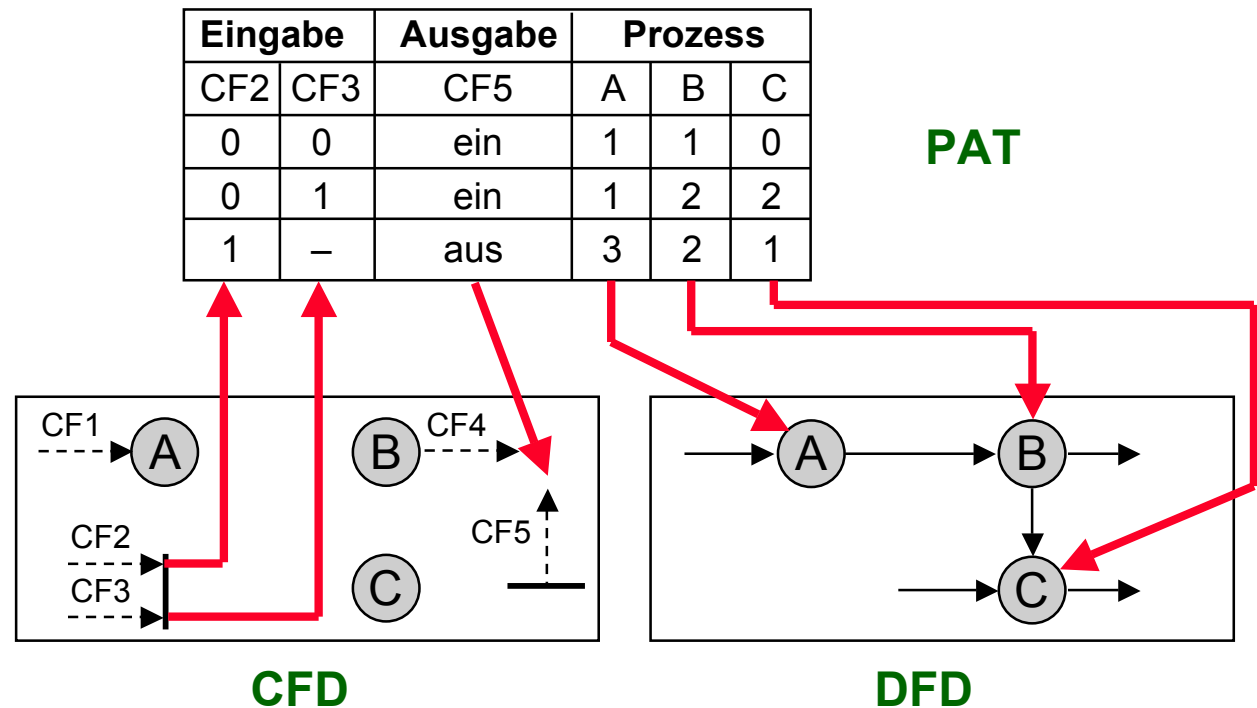


Prinzip der Beschreibung mit Prozessaktivierungstabellen (PAT)

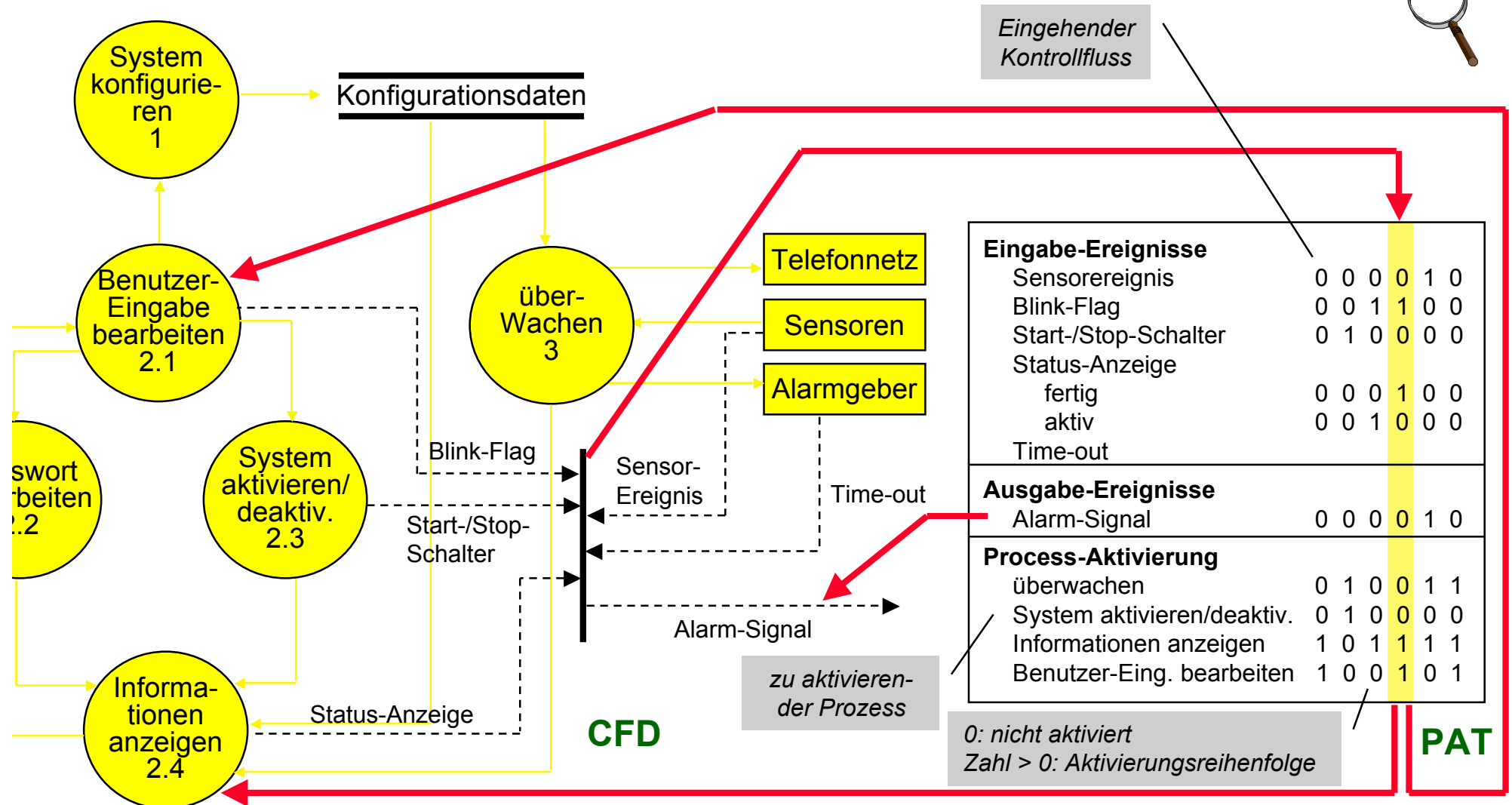
- PAT: Ereignistabelle mit 3 Teilen

- Eingabe-Kontrollflüsse (und deren Belegungen)
- Ausgabe-Kontrollflüsse (und deren Belegungen)
- Prozess-Aktivierungen belegt mit
 - 0: nicht aktiviert
 - Zahl > 0: Aktivierungsreihenfolge

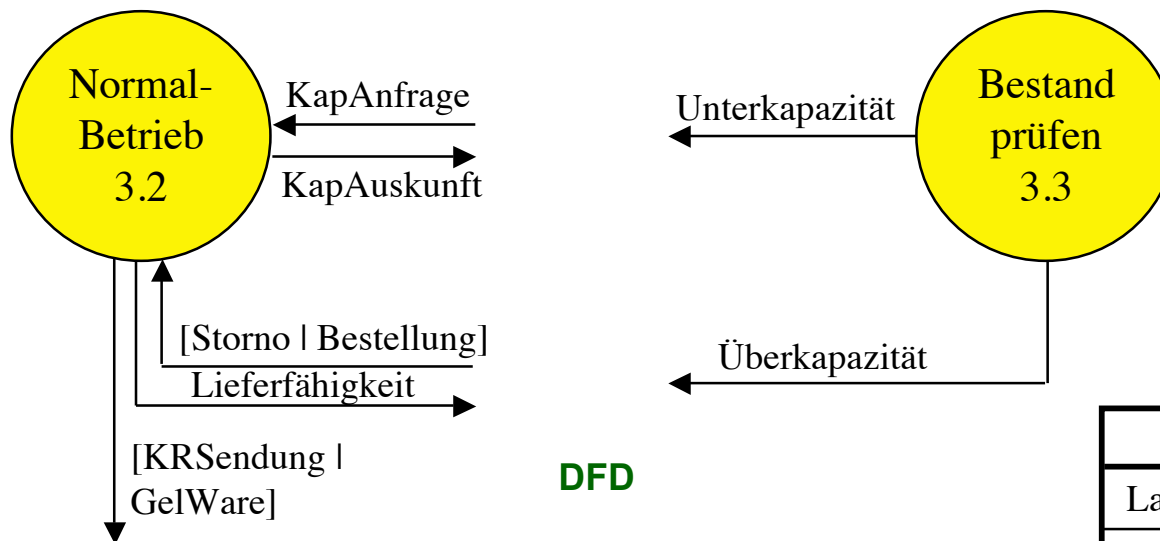
- Bedeutung:
Durch Eingabewerte im CFD werden festgelegt:
 - Ausgabewerte im CFD
 - Aktivierung von Prozessen im DFD (einschließlich Reihenfolge)



Beispiel (CFD und PAT für Prozess „mit Benutzer interagieren“)



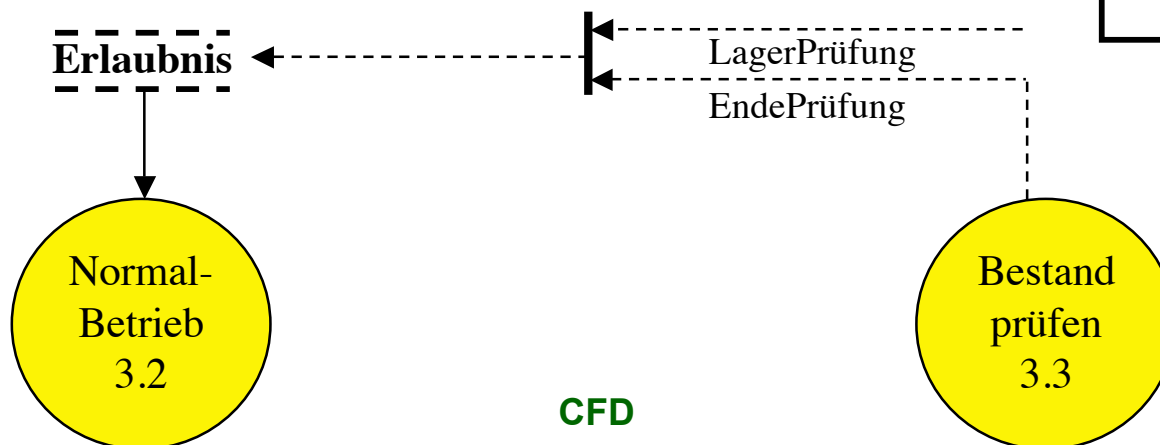
Beispiel: (Vertriebsorganisation: DFD und CFD für den Prozess „Lager“)



DFD

Eingabe		Ausgabe	Prozess	
LagerPrüfung	EndePrüfung	Erlaubnis	3.2	3.3
ein	-	false	0	1
-	ein	true	1	0

PAT

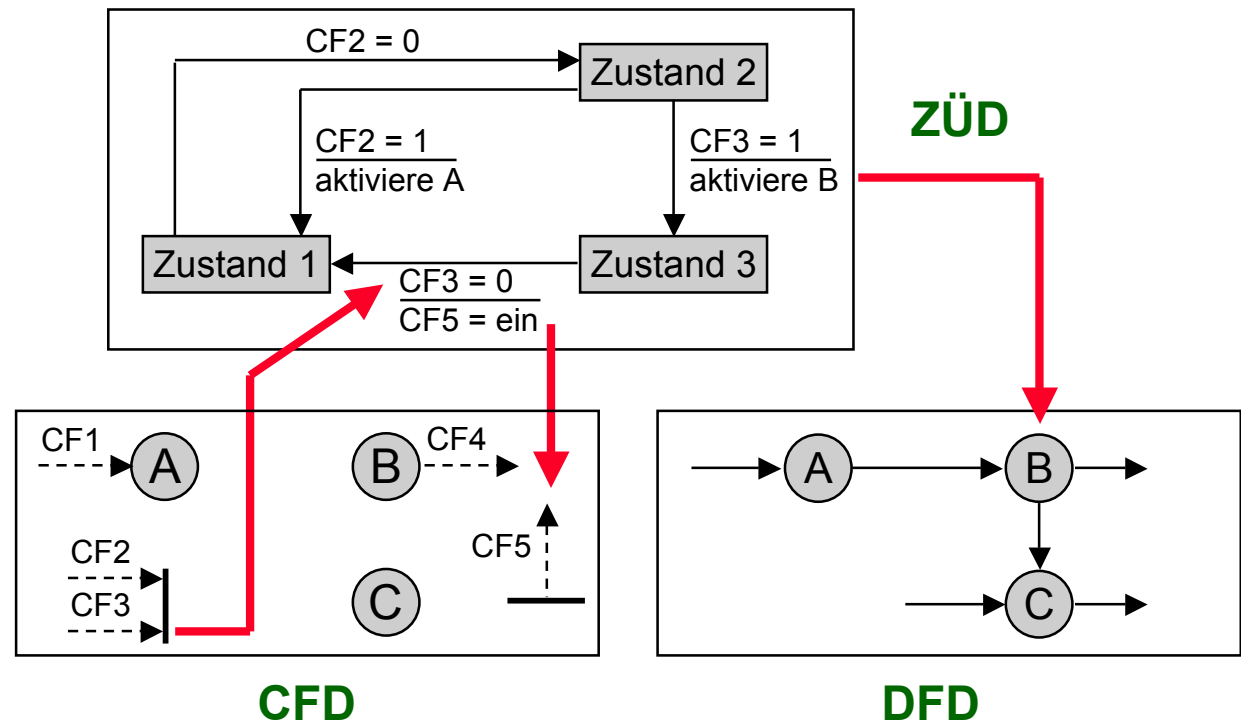


CFD

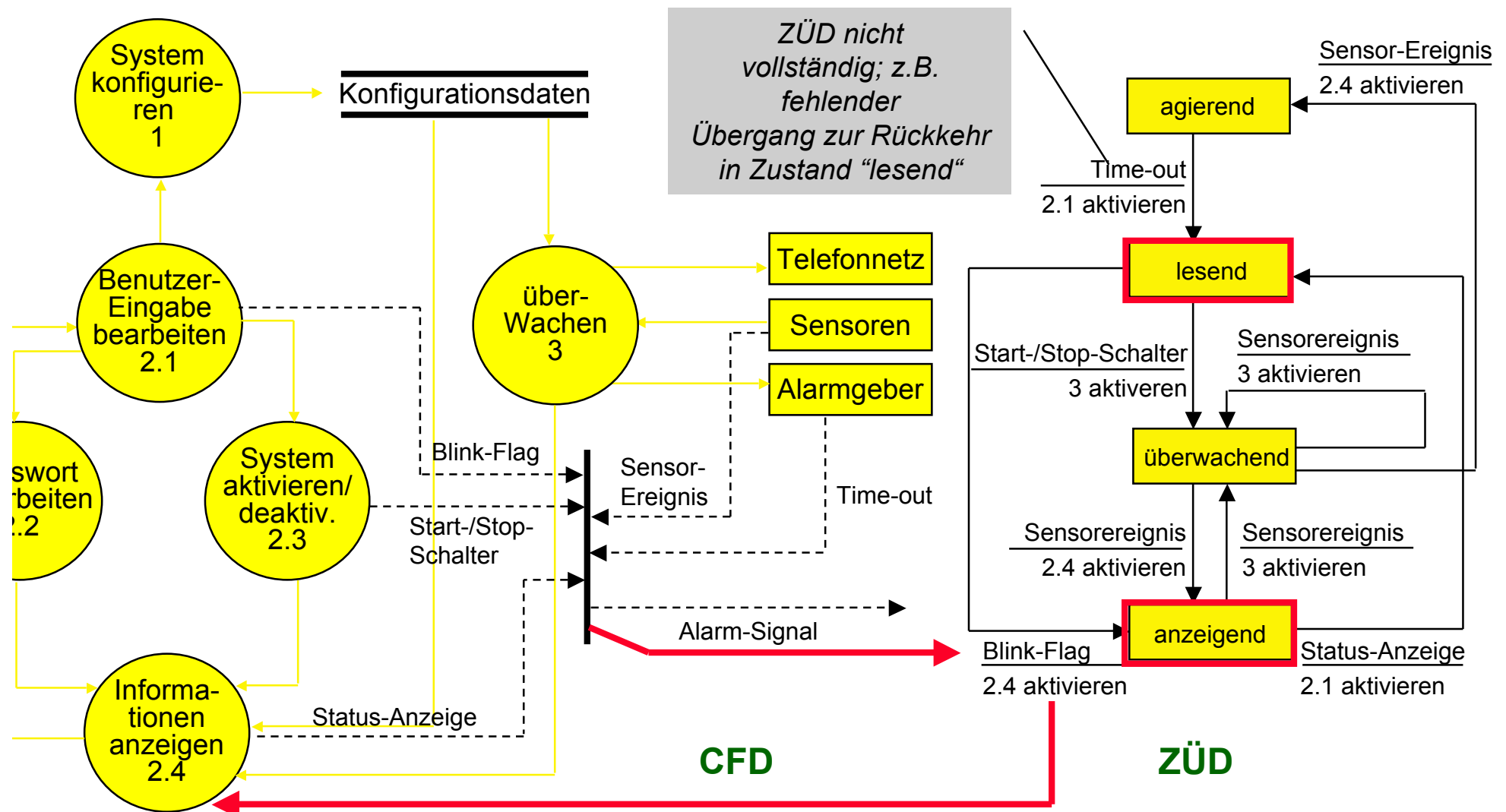


Prinzip der Beschreibung mit Zustandsübergangsdiagrammen (ZÜD)

- ZÜD hat Zustandsübergänge mit Ereignissen und Aktionen
- Ereignisse:
(Werte von)
Eingabe-Kontrollflüsse(n)
- Mögliche Aktionen
 - Prozess aktivieren
 - Kontrollfluss setzen
- Bedeutung:
 - Eingabewerte im CFD = Ereignisse im ZÜD
 - Aktionen im ZÜD bewirken
 - Aktivierung von Prozessen im DFD
 - Belegung von Ausgabewerten im CFD

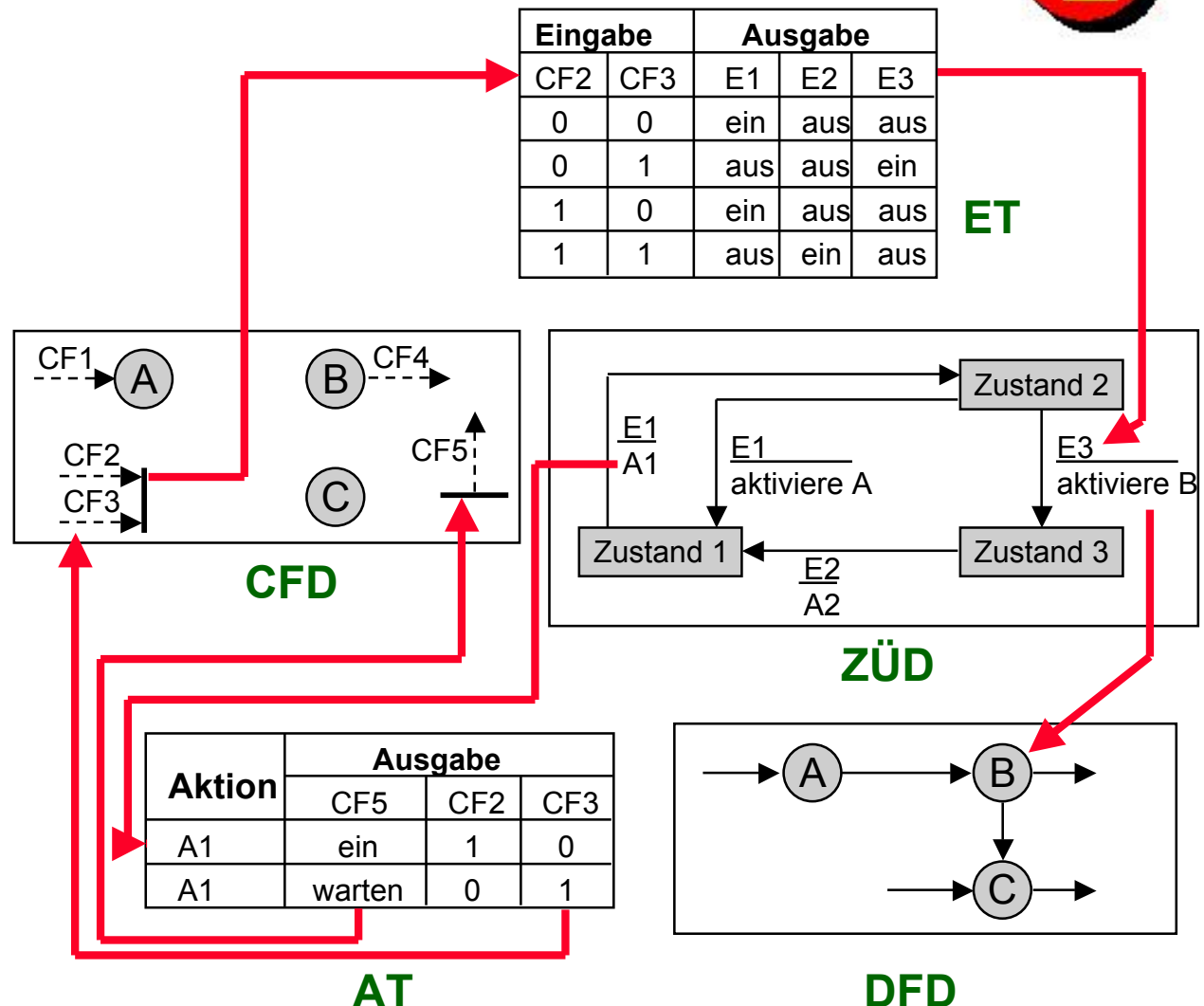


Beispiel (CFD und (rudimentäres) ZÜD und PAT für den Prozess „mit Benutzer interagieren“)



Prinzip der Beschreibung mit Aktionslogik

- ZÜD, Ereignistabelle (ET) und Aktions-/Ereignistabelle (AT)
- Bedeutung
 - Kombination von Eingabewerten im CFD legen (Vorliegen oder Nicht-Vorliegen von) Ereignissen(n) in ET fest
 - Vorliegende Ereignisse bewirken Zustandsübergänge im ZÜD
 - Zustandsübergänge im ZÜD bewirken (als Aktionen)
 - Aktivierung von Prozessen im DFD
 - Komplexe „Aktionen“
 - Komplexe Aktionen im AT bewirken Neubelegung von Steuersignalen im CFD



Kontrolle von Prozessen

- Keine Kontrolle („permanent“)
 - Prozess während gesamter Systemlaufzeit aktiv
 - Kennzeichnung: kommt in keiner CSpec vor
- Aktiviert (durch CSpec)
 - Beendet sich selbst: Schlüsselwort *issue* in zugehöriger Pspec (entspricht „Aktionen“)
 - Automatische Deaktivierung beim nächsten Zustandsübergang (entspricht „Aktivitäten“)

Dabei zu beachten: Verfeinerungshierarchie

- Wird Prozess deaktiviert, dann auch alle seine „Abkömmlinge“
(auch solche ohne eigene Prozesssteuerung)
- Prozess nur dann aktiv, wenn auch alle seine „Vorfahren“ aktiv sind

Trennung von Daten- und Kontrollaspekten

- Keine festen Regeln (insbesondere keine syntaktischen)
- Signale/Daten
 - Signal mit kontinuierlichen Werten → Datenfluss
 - Binäres Signal → fast immer Steuersignal
 - Diskretes Signal mit verschiedenen Werten
 - Zur Steuerung verwendet → Kontrollfluss
 - Zur algorithmischen Verarbeitung verwendet → Datenfluss
- Prozesse
 - Mit nicht ausschließlich diskreten Ein-/Ausgaben → Datenprozess
 - Nur mit diskreten Ein-/Ausgaben → fast immer Kontrollprozess
 - Mit zustandsabhängiger Verarbeitung → Kontrollprozess

Zeitanforderungen

- Bei den Anforderungen nur für externe Signale (im Kontextdiagramm)
- Aspekte
 - Wiederholungsrate, Neuberechnungsrate (im Anforderungslexikon definiert)
 - *Antwortzeitverhalten*
 - Einfache Zusammenhänge: tabellarische Form
 - Komplexe Zusammenhänge: Zeitdiagramme

Beispiel (Zeitspezifikation in tabellarischer Form für einen Verkaufsautomaten)

Eingabe	Ereignis	Ausgabe	Ereignis	Antwortzeit
Objekte	Einwurf	Falschmünzen	Auswurf	2 sek. max
Kunden- auswahl	Eingabe	Produkt verfügbar	Anzeigen	0.5 sek. max
		Produkt	Ausgeben	5 sek. max
Münz- rückgabe- anforderung	Eingabe	zurückgegebene Münzen	Rückgabe	2 sek. max

Unterstützungswerkzeuge

- Einige CASE-Werkzeuge (auf verschiedenen Plattformen) für SA bzw. SA/RT
(z.B: Software through Pictures, Innovator, ...) \

*Werkzeuge trennen meist nicht zwischen
den unterschiedlichen Ansätzen*

Fähigkeiten von Werkzeugen

- Erstellung von Modellen
 - Interaktive graphische Editoren für Diagramme
 - Syntaxgesteuerte Texteditoren für Prozessspezifikationen und Eintragungen ins Datenlexikon
- Analyse
 - Analysator, der mit der zentralen Datenbank kommuniziert
 - Globale Konsistenz- und Vollständigkeitsprüfungen
- Dokumentation
 - Verschiedene Arten von Dokumenten in Benutzer-definierbarem Format
 - Automatische Erstellung organisatorischer Information

Vorteile

- SA leicht erlernbar, gutes Kommunikationsmittel (auch für Nicht-Spezialisten)
- MSA zur Modellierung organisatorischer Systeme gut geeignet
- SA/RT zur Modellierung (auch komplexer) Ereignis-gesteuerter Systeme mit Echtzeitaspekten gut geeignet
- Empfohlene Vorgehensweise unterstützt systematisches Arbeiten
- Brauchbare Werkzeugunterstützung verfügbar

Nachteile

- Zu stark auf funktionales Verhalten ausgerichtet
- Interpretation der Diagramme ohne Zusatzinformation oft schwierig
- Beschreibung der Prozessdynamik (vor allem in SA/RT) gewöhnungsbedürftig
- Schwierigkeiten bei Übergang zu Daten-orientiertem Entwurf

Trotzdem

- MSA und SA/RT erste Methoden, die sich in der Praxis bewährt haben

System- und Komponentenstrukturen

Funktionales Verhalten

Kontrollaspekte, Steuerung

