Tom DeMarco
Strukturierte Analyse und System Spezifikation

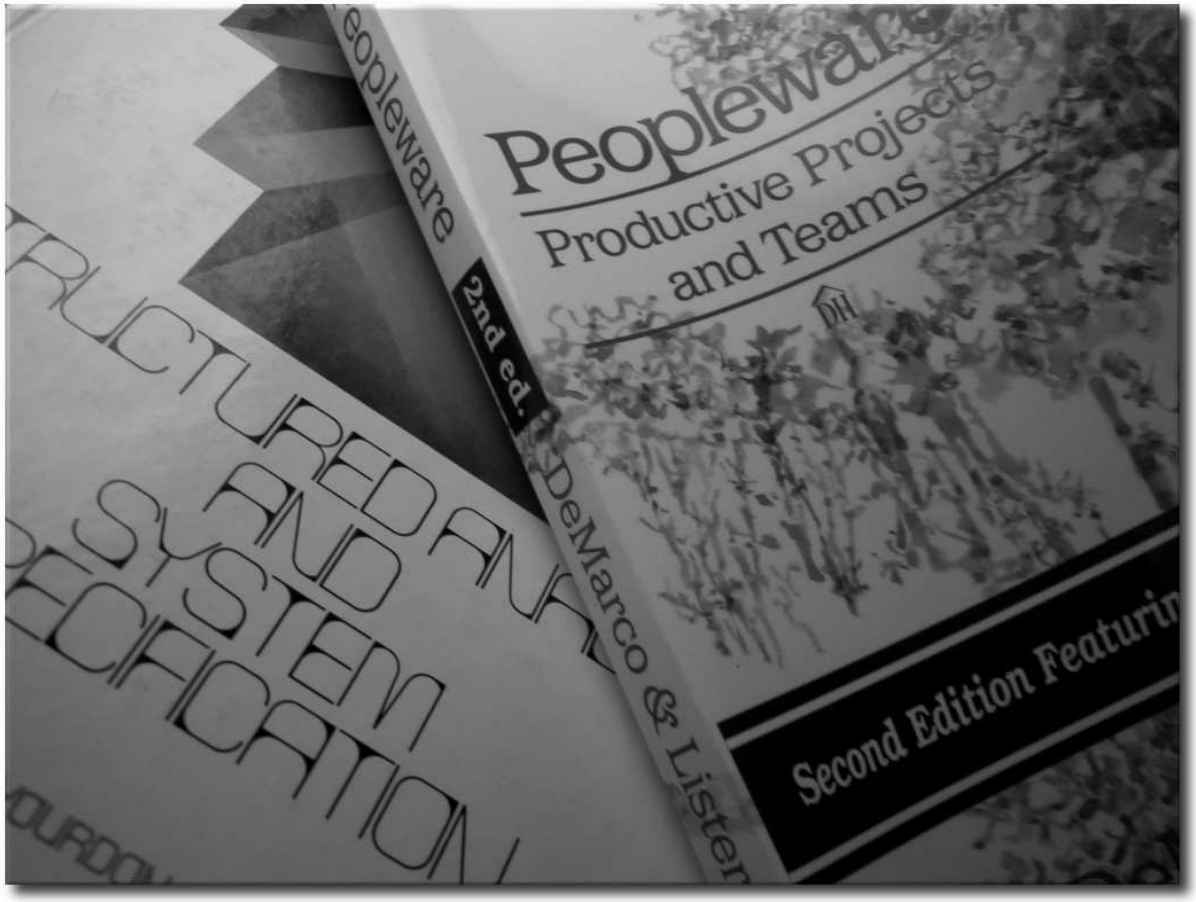
Adrian Kieß, Zweitsemester

Im schönen Sommer des Jahres 2004

Proseminar Wegweisende Arbeiten der Softwaretechnik

Universität Leipzig
Lehrstuhl für angewandte Telematik / e-Business

Prof. Dr. Volker Gruhn & Dipl.-Math. Ralf Laue



“Einen Knoten?” fragte Alice hilfsbereit. “Soll ich dir helfen, ihn zu lösen?”

- *“Alice im Wunderland”*, Lewis Caroll

Inhaltsverzeichnis

1 Einklang

http://www.dilbert.com/

1.1 Zur Person

Tom DeMarco,
bei der “Software Pioneers” Konferenz

Tom DeMarco,

- 1940 in Pennsylvenia, USA geboren
- lebt in New York
- Mitbegründer der Atlantic Systems Guild
- steuert(e) wichtige Beiträge zum Software-Management bei
- erhielt 1986 Warnier Prize für *"lifetime contribution to the field of computing"* und 1999 Stevens Award für *"contribution to the methods of software development"*
- besitzt auch eine Homepage: `http://systemsguild.com/GuildSite/TDM/TDMBio.html`
- bedeutende Veröffentlichungen:
  - *"Structured Analysis and System Specification"*, 1975
  - *"Peopleware"*, 1987
  - *"The Deadline"*, 1997

1.2 Begriffserklärung

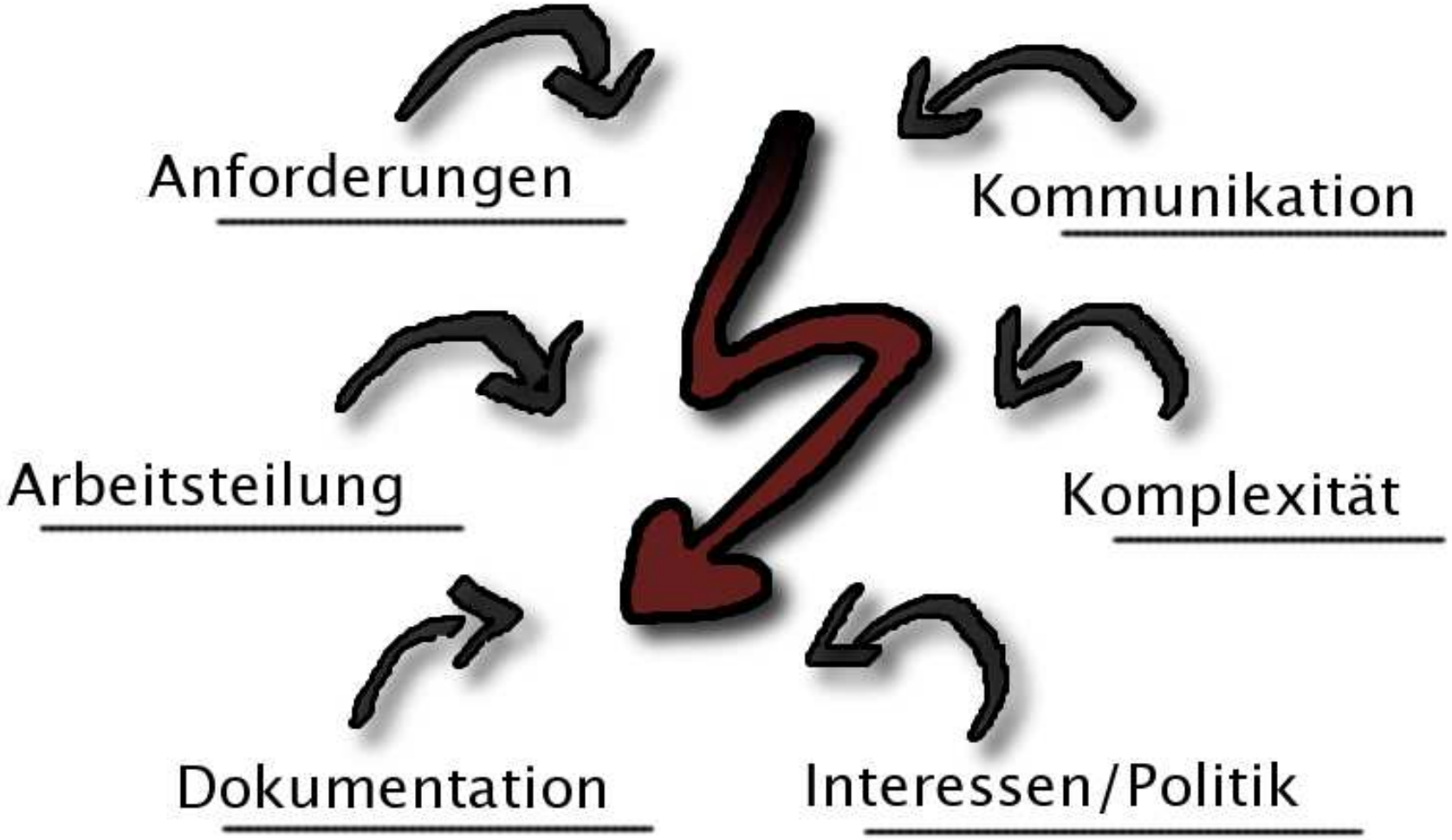
System Ein *System* besteht aus Elementen (Komponenten, Subsystemen), die untereinander in Beziehung stehen. Ein System lässt sich von seiner Umwelt (den übrigen Systemen) abgrenzen. `http://wikipedia.de`

Analyse Eine *Analyse* ist eine systematische Untersuchung, bei der das untersuchte Objekt oder Subjekt zergliedert und in seine Bestandteile zerlegt wird und diese anschließend geordnet und ausgewertet werden. `http://wikipedia.de`

Struktur Unter *Struktur* versteht man die Art der Zusammensetzung eines Systems aus Elementen und die Menge der Relationen bzw. Operationen, welche die Elemente miteinander verknüpfen. `http://wikipedia.de`

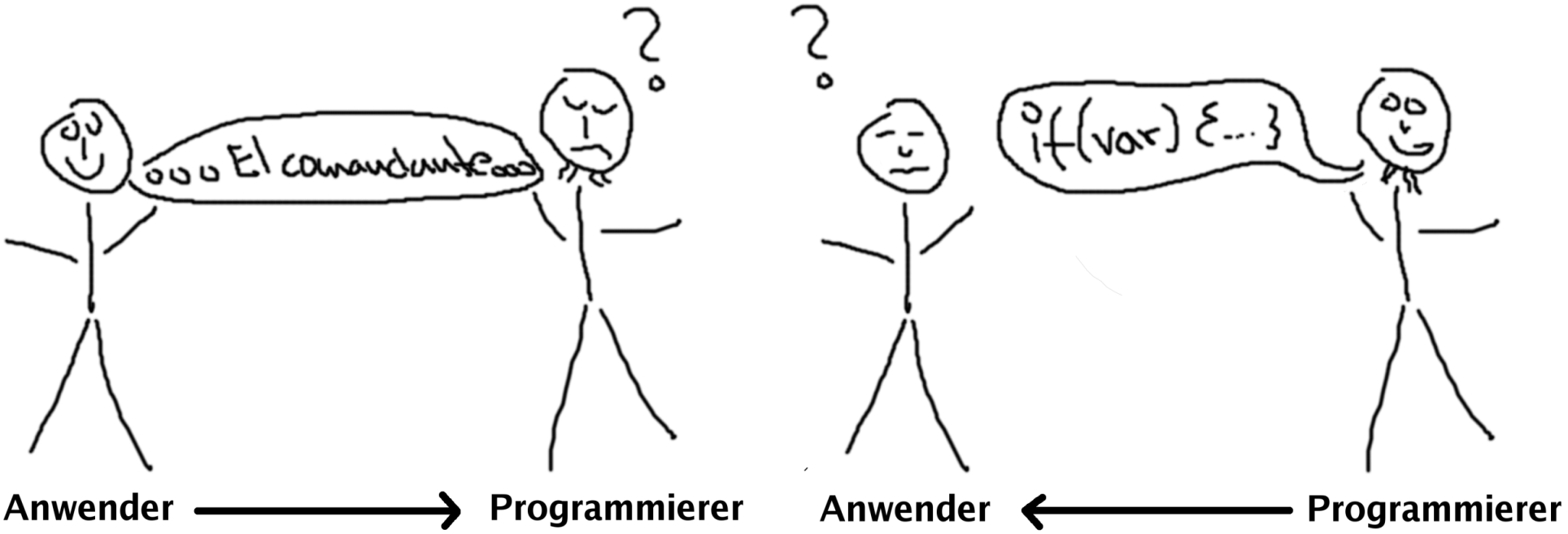
Strukturierte Analyse Die *Strukturierte Analyse* (SA) ist ein datenorientierter Ansatz konzeptioneller Datenmodellierung.

1.3 Wieso, Weshalb, Warum?



1.4 Sichtweisen

Jeder besitzt zuerst nur seine Sicht auf den Stand der Dinge.



> "The major problems of our work are not so much *technological* as *sociological* in nature."
>
> - *"Peopleware"*, page 4

1.5 Varianten

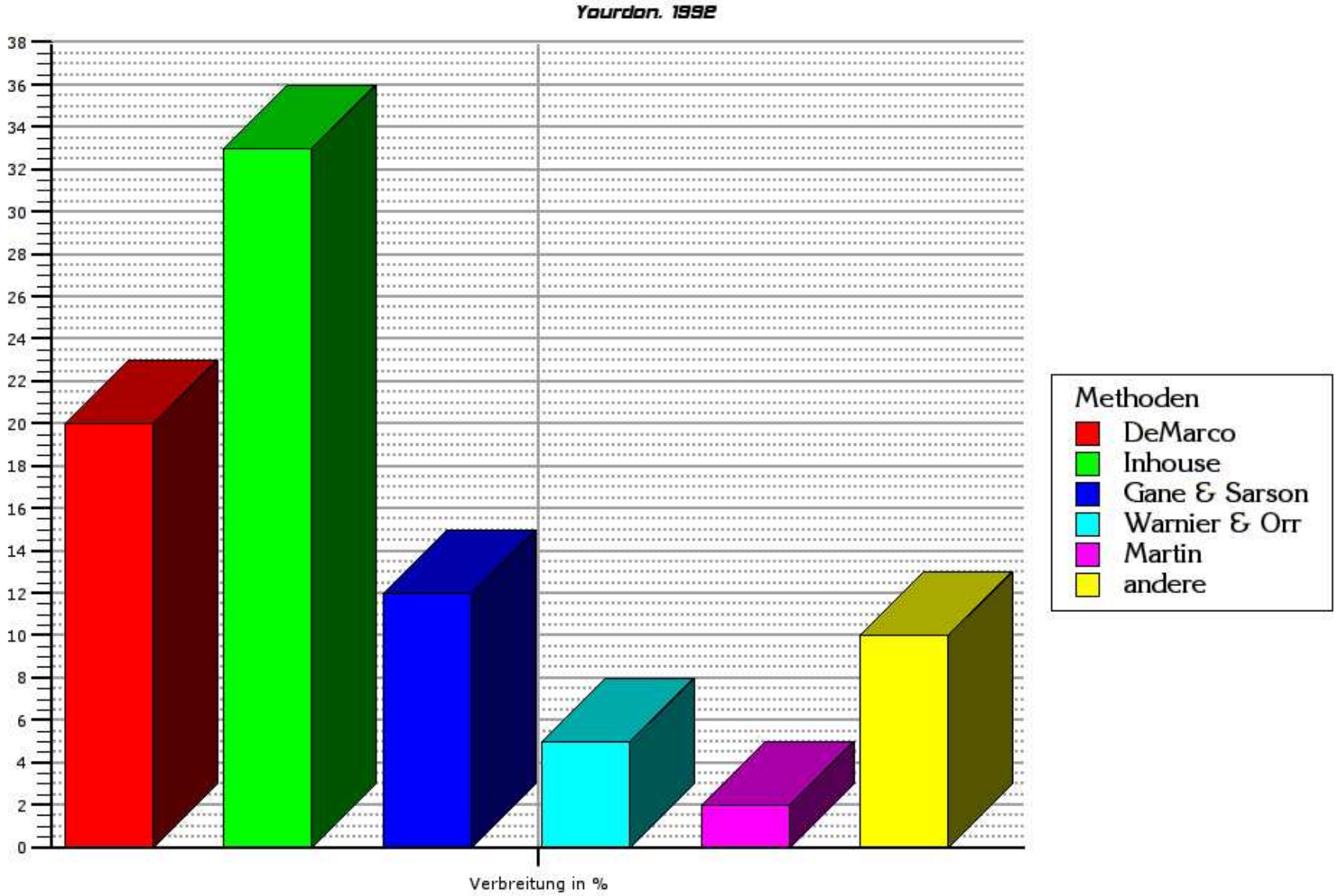
der Strukturierten Analyse:

- Weinberg, 1978: *"Structured Analysis"*
- Gane & Sarson, 1979: *"Structured Systems Analysis"*
- McMenamin & Palmer, 1984: *"Modern Structured Analysis"*
- Yourdon, 1989: *"Modern Structured Analysis"*

  Hier werden die Erfahrungen der Strukturierten Analyse seit dem Erscheinen von DeMarco's Methode gebündelt. Es wird keine neue Methode propagiert, jedoch wird die Strukturierte Analyse im gesamten Umfeld des Software-Engineering betrachtet. (Weiteres unter: `http://www.yourdon.com/`)

Wir halten uns hier natürlich an Methodik von DeMarco.

1.6 Methodenverbreitung



1.7 Definitionen

Was ist was?

Datenflüsse sind vergleichbar mit Pipelines, durch die Daten transportiert werden.

Datenspeicher bilden eine Ablagemöglichkeit für Daten, bei denen sich der Erstellungszeitpunkt vom Gebrauchszeitpunkt unterscheidet.

Prozesse haben die Aufgabe, Eingabedaten in Ausgabedaten zu verarbeiten und enthalten die hierfür notwendigen Algorithmen.

Terminatoren stehen für die Beziehungen des Systemes zur Außenwelt. Sie senden oder empfangen Daten, verarbeiten diese jedoch nicht.

1.8 Konzepte

Die Strukturierte Analyse fußt im Wesentlichen auf drei, miteinander verknüpften Konzepten:

1. Datenflussdiagramm *(Data Flow Diagram (DFD))*
2. Datenlexikon *(Data Dictionary (DD))*
3. Primitive Prozessspezifikation *(MiniSpec)*

2 Datenflussdiagramm

Das *Datenflussdiagramm (DFD)* beschreibt eine hierarchische Betrachtung des Systemes.

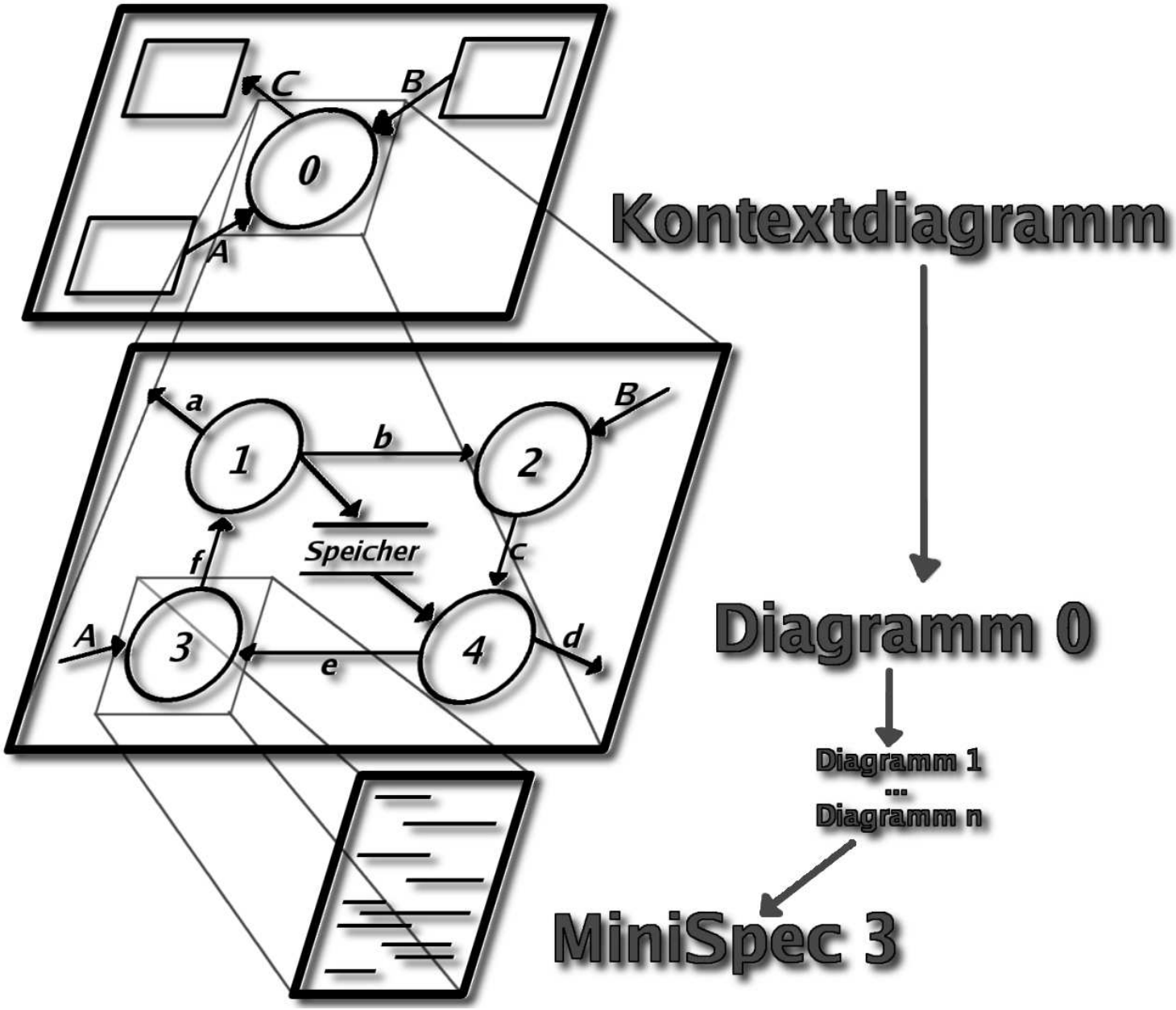
Kontextdiagramm (nullte Ebene): Die Schnittstelle zwischen dem System und der Umwelt.

Diagramm 0 (erste Ebene): Zerlegung des Systemes in Subsysteme; Angabe der Daten, welche zwischen den Subsystemen bzw. Teilfunktionen fließen.

Subdiagramme (n-te Ebene): Zerlegung des vorherigen Subsystemes in weitere Subsysteme.

Prozesse, Datenspeicher und Datenflüsse sollen mit einem möglichst prägnanten Namen versehen werden. Zum Beispiel: “prüfe Passwort”, aber *nicht*: “verarbeite Datei” (Welche Datei? - Wie wird sie verarbeitet?).

2.1 Hierarchiekonzept



2.2 Kontextdiagramm

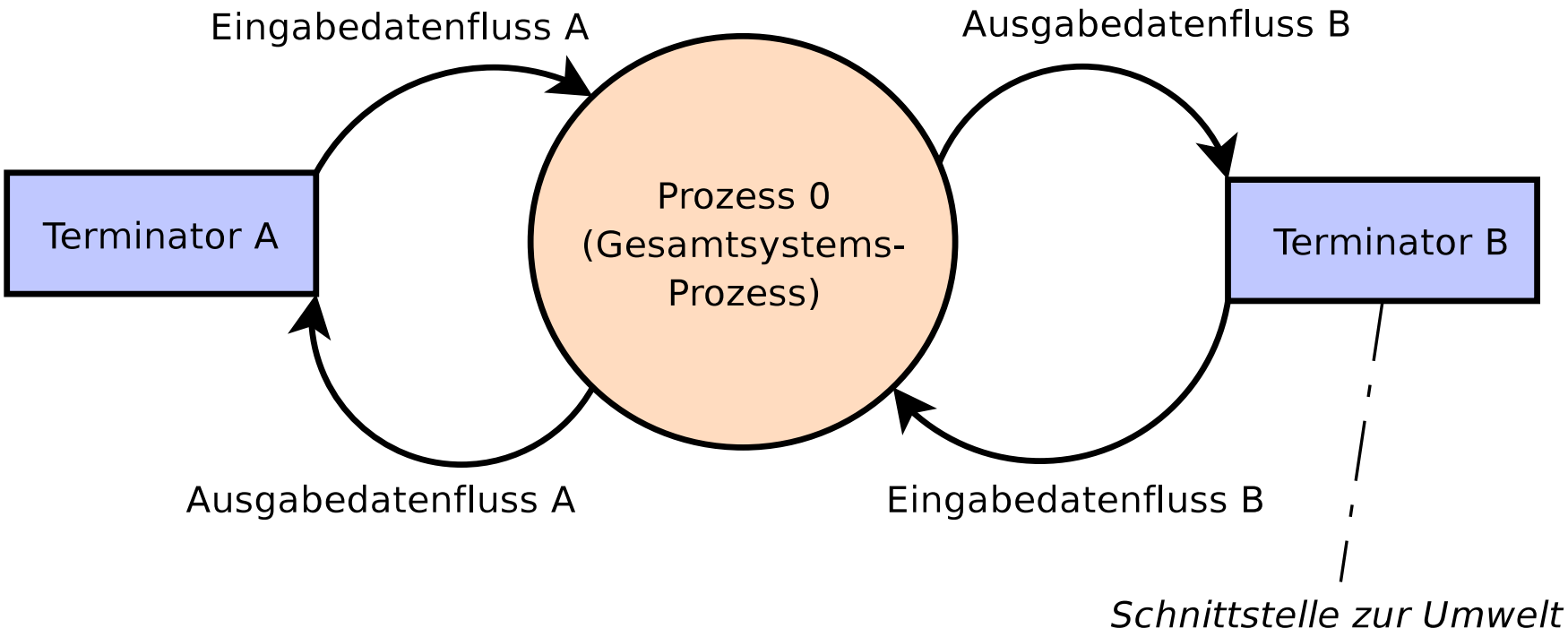
Syntax

- enthält nur einen Prozess, dieser erhält die *Nummer 0* und stellt das Gesamtsystem dar
- verfügt über mindestens eine Schnittstelle

Semantik

- beschreibt den Anwendungsbereich des modellierten Systemes
- zeigt Datenflüsse, welche Systemgrenzen überschreiten
- ist die Zusammenfassung von *Diagramm 0*

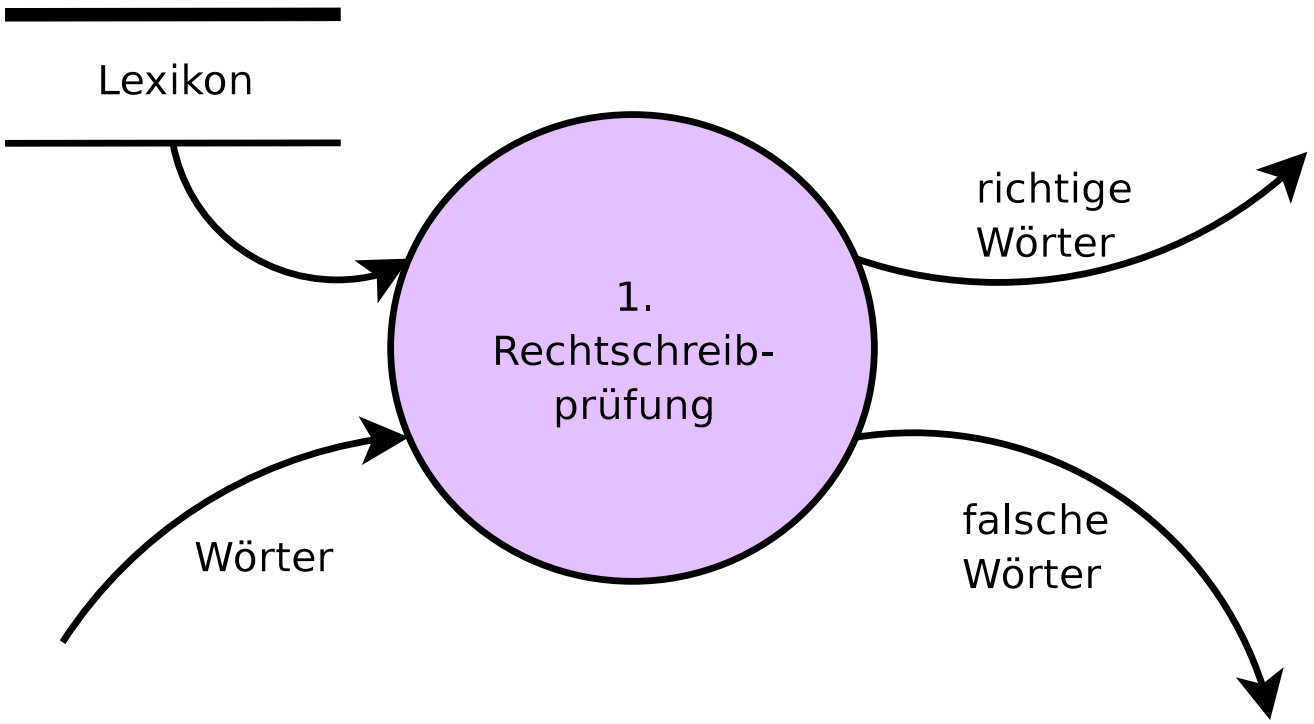
Kontextdiagramm



2.3 Diagramm 0

beschreibt die Verfeinerung des Kontextdiagrammes.

- Der *Prozess 0* aus dem Kontextdiagramm wird in Teilprozesse zerlegt.
- Speicher werden eingefügt.



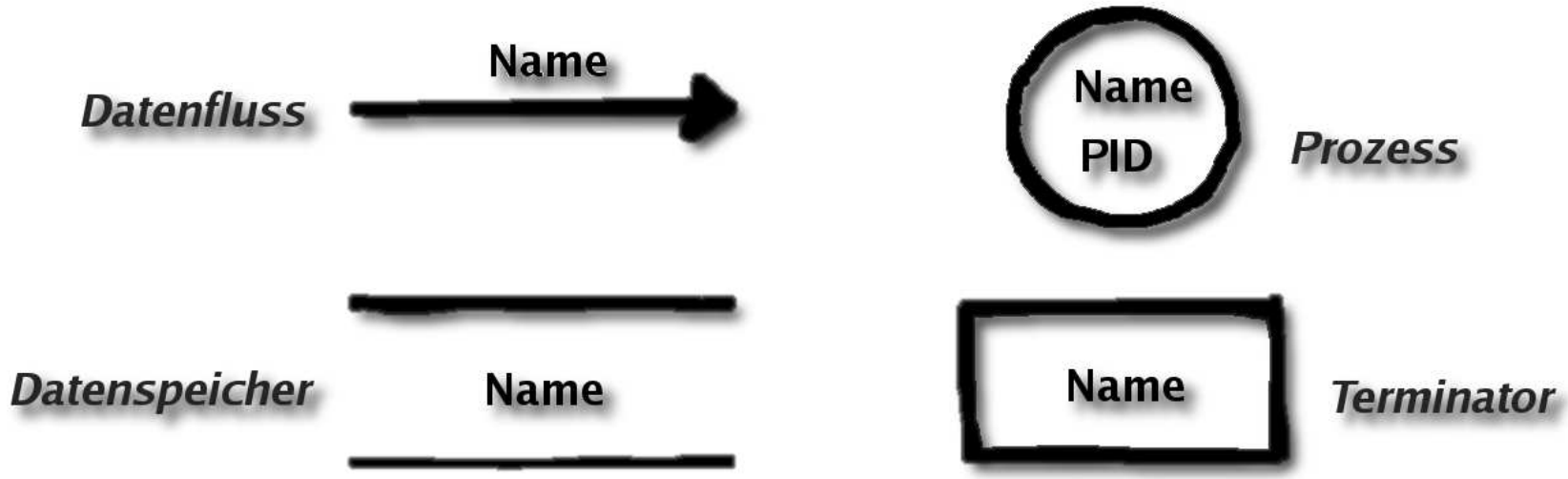
2.4 Prozessverfeinerung

Jeder Prozess kann wieder zu einem neuen Diagramm verfeinert werden, welches folgende Bedingungen erfüllen muss:

- Der Prozess wird in Teilprozesse zerlegt.
- Jeder Prozess wird fortlaufend nummeriert.
- Wurde ein Prozess ausreichend verfeinert, wird er abschließend durch eine MiniSpec beschrieben.
- Die Anzahl der Prozesse pro Diagramm sollte sieben nicht überschreiten.

```
DFD 1, DFD 2, DFD n ... DFD 1.1, DFD 1.2 ... DFD n.m ...
```

2.5 Notation



Datenflüsse werden durch einen Vektorpfeil dargestellt.

Datenspeicher werden durch zwei parallele Striche dargestellt.

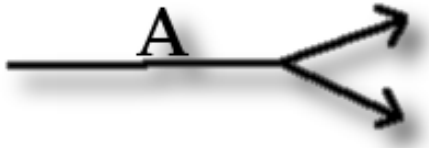
Prozesse werden durch Kreise (Bubbles) dargestellt.

Terminatoren werden durch ein Viereck dargestellt.

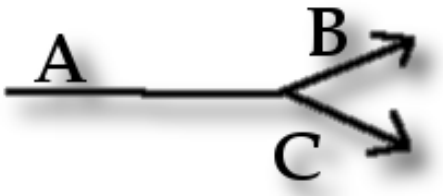
2.6 Datenflüsse

Wie können Daten fließen?

A fließt von links nach rechts



A wird in beide Zweige weitergeleitet



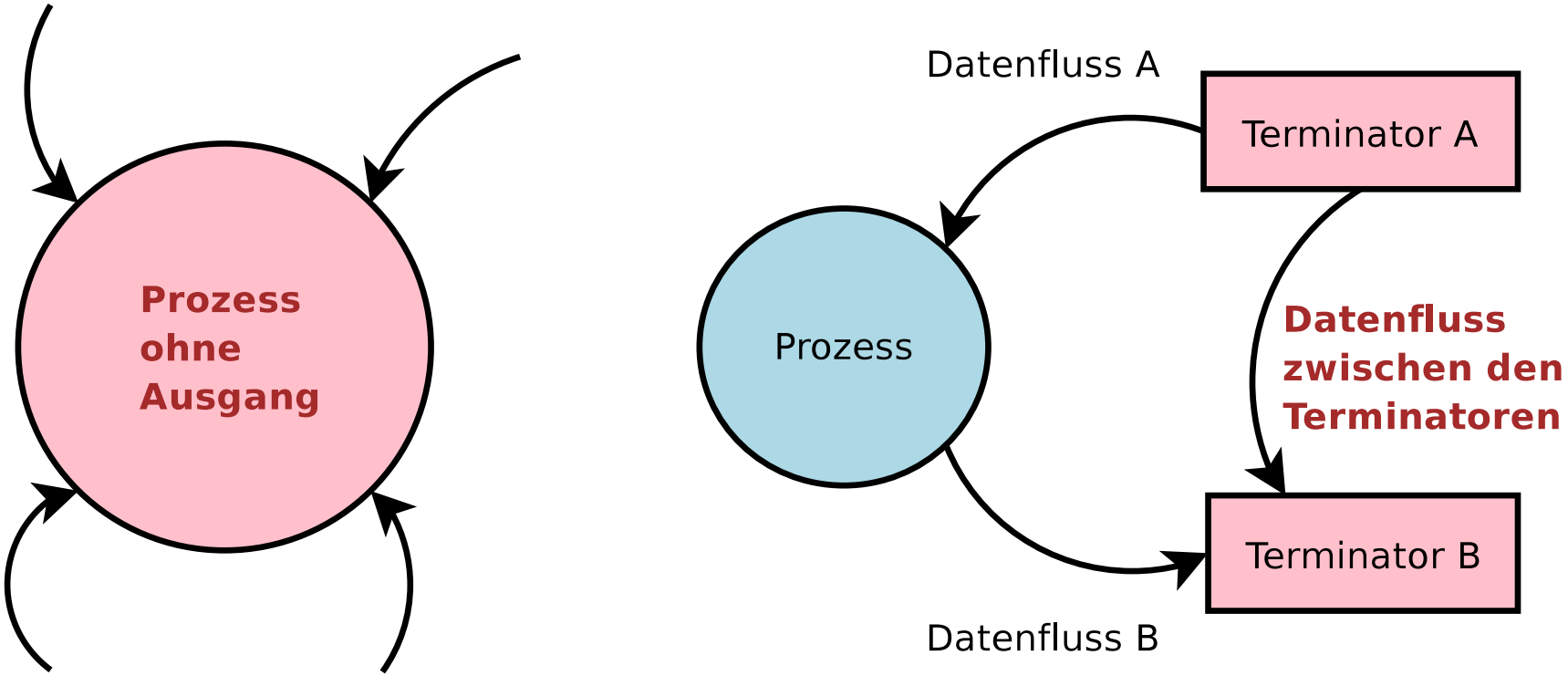
A teilt sich in *B* und *C*; A=B+C

A entsteht aus *B* und C

A fließt in beide Richtungen

2.7 Fehlermöglichkeiten

Was *nicht* funktioniert, oder sinnvoll ist:



3 Datenlexikon

Das *Datenlexikon (DD)* definiert die Strukturen aller verwendeten Informationen und erlaubt die Überprüfung des Datenmodelles auf Redundanz und Widerspruchsfreiheit.

> "A data dictionary is a repository of data about data."
>
> - James Martin, *"Principles of Data-Base Management"*

Beachte:

- Schlüssige Variablen wählen.
- Die Variaben müssen definiert werden.
- Definitionen entsprechen der Notation in Backus-Naur-Form.

3.1 Beispiel

Beispiele für ein Datenlexikon:

```
Artikeldaten = Artikel_id + Erstellungsdatum +
     Titel + Text
Benachrichtigungsschreiben = [Account angelegt |
     Account gelöscht | Nachricht von anderem Nutzer]
Artikelanfrage = User_id + 1{ Artikel_id }20

Kunde = Kunden_id + Name + Adresse
Buch = Buch_id + Autor + Titel + Preis
Buchliste = {Buch}
Einkauf = {Bestellung + Gesamtbetrag}
Kundenliste = {Kunde}
Kundenstatus = [“reich” | “arm”]

Zimmerart = [”Einzelzimmer” | “Doppelzimmer”]
* Ein Kommentar ist immer wichtig. *
```

4 MiniSpec

Bei den *MiniSpecs* handelt es sich um Subsysteme, welche nicht mehr (sinnvoll) durch Datenflussdiagramme aufgeteilt werden können. Sie beschreiben implementierungsunabhängig, wie Eingangsdaten in Ausgangsdaten umgewandelt werden. Die Darstellung erfolgt häufig mit Pseudocode, welcher auf eine DIN A4-Seite passen sollte.

4.1 Beispiel

Unbenutzte User-Accounts löschen:

```
Alle User-Accounts aus Datenbank selektieren;
if letzter Login > 365 Tage
     then verschicke Benachrichtigungsmail
else if letzter Login > 400 Tage
     then lösche Account
```

5 Beispiel

Wir kochen Kaffee.

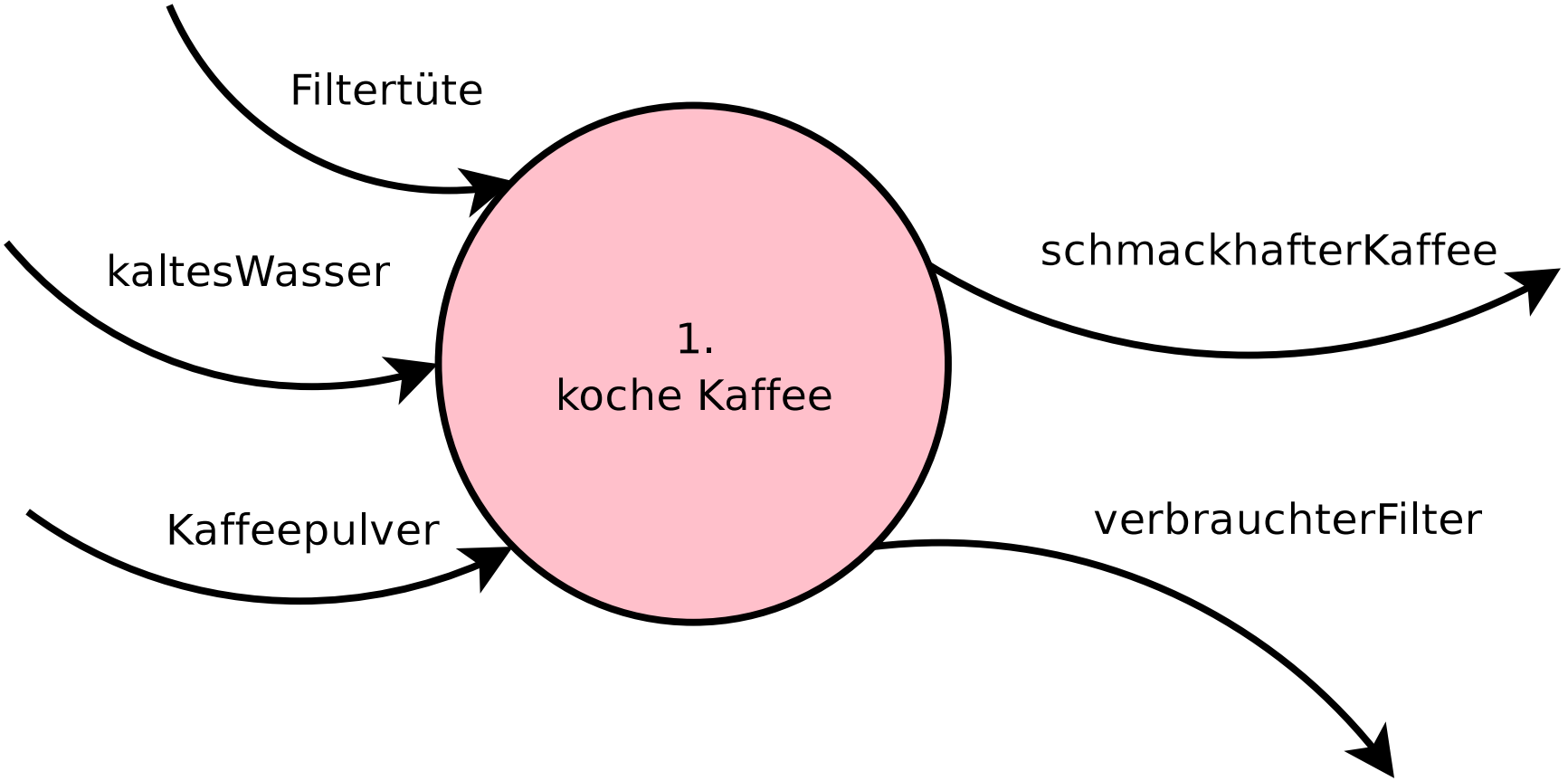
5.1 Erster Schritt

Wir fragen uns: Welche Objekte werden bewegt? Diese Objekte müssen wir nun in das Datenlexikon eintragen.

```
* Datenlexikon: koche Kaffee *
Filtertüte =  [ Naturpapierfilter | gebleichter Filter ]
kaltesWasser = 200{ Milliliter Wasser }1000
Kaffeepulver = 20{ Kaffeebohnen }200
verbrauchterFilter = Filtertüte + (Wasser)
schmackhafterKaffee =  Kaffeepulver + Wasser
```

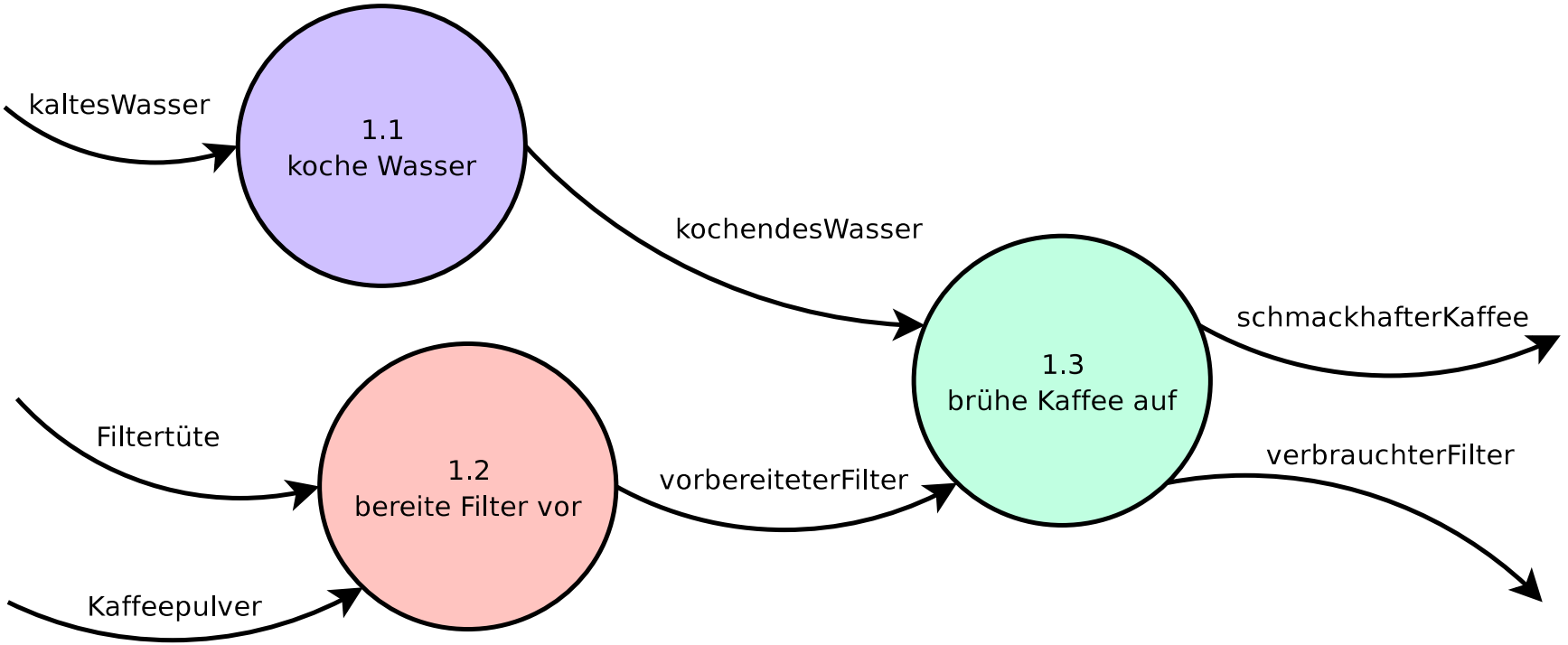
5.2 Zweiter Schritt

Sodann zeichnen wir das Datenflussdiagramm mit den Objekten, welche wir soeben in das Datenlexikon eingetragen haben.



5.3 Dritter Schritt

Danach fragen wir uns: Welche Aktionen und Handlungen treten in dem System auf? Aus diesen Aktionen und Handlungen leiten wir dann das Subdiagramm ab. Die neu eingeführten Datenflüsse werden dem Datenlexikon hinzugefügt.



5.4 Vierter Schritt

Da eine weitere Verfeinerung der Prozesse nicht mehr sinnvoll ist, beschreiben wir die einzelnen Prozesse durch eine MiniSpec.

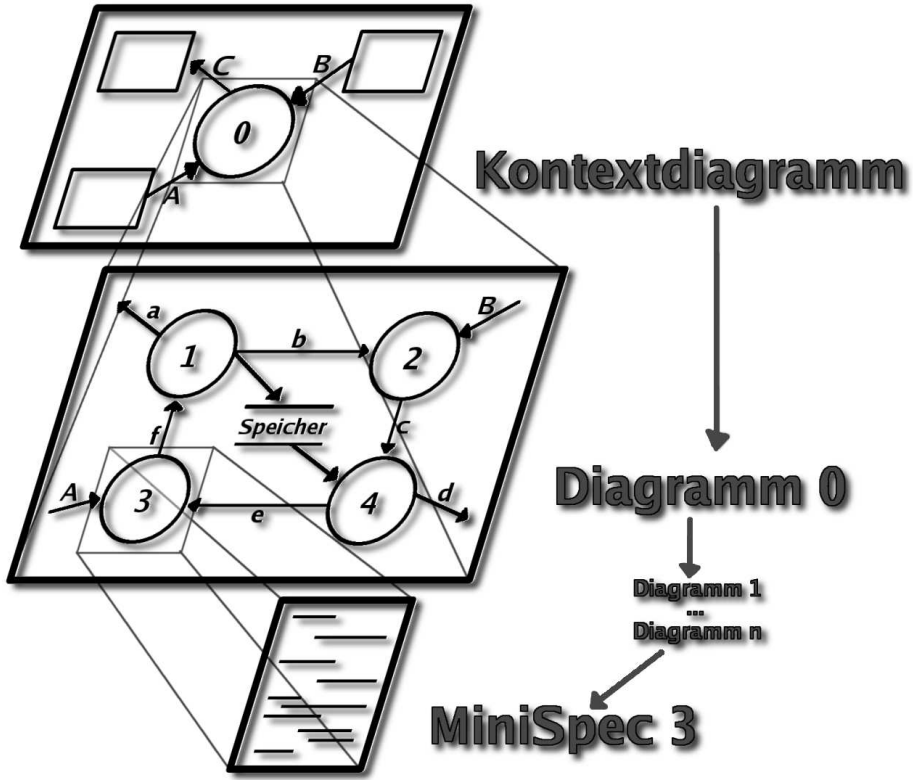
```
* MiniSpec 1.1: koche Wasser *
Stelle Topf mit kaltem Wasser auf Herd
Aktiviere Herd
Zur Couch und döse 200 Sekunden
while true {
     Überprüfe Wassertemperatur mit Finger
     if Schmerzhaft heiß
         Stelle kochendes Wasser vom Herd
         Deaktiviere Herd
         break
     else
         Zur Couch, döse 30 Sekunden
              und kehre dann wieder zurück
}
```

6 Zusammenfassung

Wir lehnen uns zurück und repetieren.

6.1 Drei Konzepte

1. Datenflussdiagramm *(Data Flow Diagram (DFD))*
2. Datenlexikon *(Data Dictionary (DD))*
3. Primitive Prozessspezifikation *(MiniSpec)*



6.2 Hierarchiekonzept

Kontextdiagramm beschreibt die Umwelt des Systemes, es gibt genau einen Prozess.

Diagramm 0 beschreibt die Verfeinerung des Kontextdiagrammes. Der Gesamtsystems-Prozess, *Prozess 0*, wird in Teilprozesse zerlegt.

Prozessverfeinerung Jeder Prozess kann wieder zu einem neuen Diagramm verfeinert werden. Wurde ein Prozess ausreichend verfeinert, wird er abschließend durch eine MiniSpec beschrieben.

7 Ausklang

7.1 Stärken

Wo liegen die Stärken der Strukturierten Analyse?

- leicht erlernbar
- leicht nachvollziehbar
- Kombination bewährter Konzepte
- Qualitätssicherungsmöglichkeiten vorhanden
- ermöglicht *Top-Down*-Einarbeitung in das System
- hierarchische Gliederung sorgt für Übersichtlichkeit
- über den Speicherzusammenhang zu ER-Modellen herstellbar

7.2 Schwächen

Wo liegen die Schwächen der Strukturierten Analyse?

- Schnittstellen können nicht verfeinert werden
- Speicher können nicht verfeinert werden
- nur der Datenfluss wird analysiert
- nur *Top-Down*-Partitionierung
- keine Lokalität von Daten

Vielen Dank für Ihre Aufmerksamkeit

"Ein Arzt, ein Hochbauingenieur und ein Informatiker unterhielten sich darüber, was der älteste Beruf der Welt sei. Der Mediziner führte an, 'Schon in der Bibel heißt es, daß Gott Eva aus Adams Rippe erschaffen hat. Dafür wurde natürlich die Medizin gebraucht, und so darf ich wohl behaupten, daß ich den ältesten Beruf der Welt habe.' Der Hochbauingenieur unterbrach ihn und sagte, 'Aber noch vorher im Buch Genesis heißt es, daß Gott die Ordnung des Himmels und der Erde aus dem Chaos geschaffen hat. Das war der erste und wahrscheinlich der spektakulärste Einsatz des Bauingenieurwesens. Deshalb, lieber Doktor, irren Sie sich: ich habe den ältesten Beruf der Welt.' Der Informatiker lehnte sich in seinem Stuhl zurück, lächelte und sagte dann verbindlich: 'Und wer, meine Herren, glauben Sie, hat das Chaos erschaffen?"'

- Grady Booch

Anhänge

Backus-Naur-Form

Die Notation für die *Backus-Naur-Form (BNF)* setzt sich wie folgt zusammen:

= ist äquivalent

+ Sequenz; *und*

[..|..] Auswahl; *entweder, oder*

{} Wiederholung; *m { } n, von m bis n-mal*
{ ... } 5; *höchstens fünf*
2 { ... } 8; *zwei bis acht*

() Option

.. Kommentar