

Programmierkonzepte in den Naturwissenschaften

30. Datenaufnahme: Metadaten

PD Dr. Till Biskup
Physikalische Chemie
Universität des Saarlandes
Sommersemester 2021





- ❏ Daten ohne Metadaten sind wertlos.
Beide zusammen bilden eine untrennbare Einheit.
- ❏ Die Information ist bei Datenaufnahme maximal.
Eine sinnvolle Reduktion ist die eigentliche Herausforderung.
- ❏ Metadaten sollen den Auswertungsroutinen ein semantisches Verständnis ermöglichen.
- ❏ Metadaten sollten strukturiert und für Mensch und Maschine lesbar abgelegt werden.
- ❏ Ein Format für Metadaten sollte plattformunabhängig und möglichst einfach nutzbar sein.

Bedeutung: Daten ohne zusätzliche Informationen sind wertlos

Zielstellung: semantisches Verständnis durch Auswertungsroutinen

Kriterien für eine formalisierte Ablage von Metadaten

Ein reales Beispiel: die Infodatei

- ▶ Datenerhebung findet immer in einem Kontext statt.
 - Was zur Spezifität des Kontextes gehört, muss immer im Einzelfall geklärt werden.

- ▶ Metadaten sollten fünf Fragen beantworten:
 - Wer hat
 - was
 - mit wem
 - wann und
 - wie gemacht?

- ☞ Gilt sowohl für die eigentliche Datenaufnahme als auch später für jeden Verarbeitungsschritt der Daten.

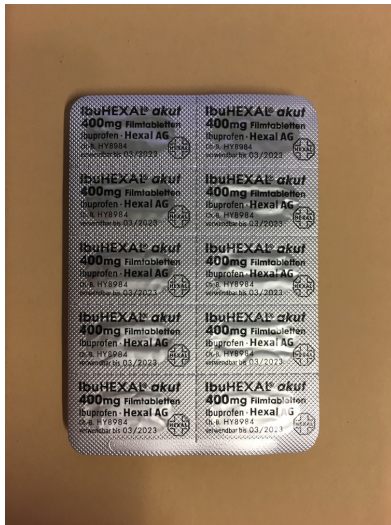
Datensatz

Einheit von (numerischen) Daten und
über die Daten verfügbare Informationen (Metadaten)

- ▶ Daten und Metadaten immer gemeinsam ablegen
 - nicht notwendigerweise in derselben Datei
 - konkrete Implementierung nebensächliches Detail (DIP)
- ▶ Importroutinen lesen die Metadaten mit ein.
 - Nur so wird ein semantisches Verständnis der Daten durch die Auswertungsroutinen ermöglicht.
- 👉 Datensatz entsprechend in Software implementieren (als Teil des „Kerns der Anwendung“, vgl. DDD)

Daten und Metadaten bilden eine Einheit

Ein alltägliches Beispiel: Blister-Verpackung von Tabletten



These

Die Menge verfügbarer Informationen ist während der Datenaufnahme maximal.

- ▶ Verantwortung des Experimentators
 - aus der Summe der verfügbaren Informationen die relevanten auswählen und dokumentieren

- ▶ Welche Informationen sind relevant?
 - häufig eine Frage der Erfahrung
 - auf der Erfahrung anderer aufbauen
 - Heuristiken anwenden

- ▶ allgemeine wichtige Informationen
 - Datum
 - Durchführende(r)
 - Probe
 - Temperatur

- ▶ verwendete Messapparatur
 - bei mehreren ähnlichen Geräten mit gleichem Datenformat
 - Information geht beim Export der Daten oft verloren.

- ▶ Aufbau aus austauschbaren Komponenten
 - Hersteller und Typenbezeichnung für jede Komponente
 - wichtige Parameter für jede Komponente
 - ggf. Metadaten nach Komponenten geordnet ablegen

- 👉 Erhebung der relevanten Metadaten formalisieren

Bedeutung: Daten ohne zusätzliche Informationen sind wertlos

Zielstellung: semantisches Verständnis durch Auswertungsroutinen

Kriterien für eine formalisierte Ablage von Metadaten

Ein reales Beispiel: die Infodatei

Semantik

Wissenschaft von der Bedeutung der Zeichen;
Zeichen können auch Sätze, Satzteile, Wörter oder Wortteile sein.

- ▶ Computer können nicht denken.
 - „Verständnis“ ist in diesem Kontext immer relativ zu sehen.
- ▶ Formalisierung der Informationen
 - Der Mensch ist viel flexibler bei der Mustererkennung...
- ▶ strukturierte Ablage innerhalb der Software
 - als Schlüssel-Wert-Paare
 - assoziatives Datenfeld (zunächst) am Besten geeignet

Geordnete Listen

#	Wert
1	0.0000
2	0.0025
3	0.0050

⋮

n-1	0.0600
n	0.0625

#	Wert
1	'Im'
2	'Anfang'
3	'war'

⋮

n-1	'die'
n	'Tat'

Assoziative Datenfelder

Schlüssel	Wert
Name	'K. Racht'
Alter	42
Adresse	

Schlüssel	Wert
Straße	'Talstraße'
Nummer	21

Hobbies	{'...', '...', '...'}
---------	-----------------------

assoziatives Datenfeld

Datenstruktur, die nichtnumerische Schlüssel verwendet, um die enthaltenen Elemente zu adressieren.

- ▶ Bezeichnung je nach Programmiersprache unterschiedlich
 - *map, dictionary, associative array, hash, struct*
- ▶ Schlüssel in keiner bestimmten Reihenfolge abgelegt
- ☛ Schlüsselnamen sollten eine nachvollziehbare Verbindung zwischen Schlüssel und Feldinhalt liefern.
- ☛ Sorgt für ausdrucksstarken, *lesbaren* Code.

- ▶ einfaches Beispiel: Achsenbeschriftung
 - Informationen (Größe und Einheit) entsprechend ablegen
 - Achsenbeschriftungen weitestgehend automatisierbar
- ▶ einfaches Beispiel: Abbildungsunterschrift
 - alle wichtigen Kenndaten über einen Datensatz bekannt
 - Abbildungsunterschrift automatisch generierbar
- ▶ komplexeres Beispiel: Einheitenkonvertierung
 - Voraussetzungen: Einheit, Umrechnungsvorschrift
 - Umrechnungsroutine bekommt Ausgangs- und Zieleinheit
 - generisch, wenn die Dimension von Einheiten bekannt ist
- ▶ einfache Überprüfung auf Konsistenz
 - Bsp: zwei Datensätze sollen aufeinander addiert werden.
 - Achseninformation erlaubt Überprüfung auf Kompatibilität

Listing 1: Beispiel für automatisch erzeugte Achsenbeschriftungen

```
xlabelText = sprintf('\it %s) / %s',  
    dataset.axes.x.measure, dataset.axes.x.unit);
```

- ▶ ausdrucksstarker Code durch intuitive Benennung
- ▶ Größe kursiv, Einheit aufrecht, Schrägstrich als Trenner

Listing 2: Beispiel für Überprüfung von Achsen auf Konsistenz

```
if dataset1.axes.x.values == dataset2.axes.x.values  
    % Do something with datasets, e.g., add them up  
end
```

- ▶ Es gibt noch viel komplexere Überprüfungsmöglichkeiten ...

Bedeutung: Daten ohne zusätzliche Informationen sind wertlos

Zielstellung: semantisches Verständnis durch Auswertungsroutinen

Kriterien für eine formalisierte Ablage von Metadaten

Ein reales Beispiel: die Infodatei

- ▶ einfach von Nutzern zu schreiben
 - modulares, elektronisches Laborbuch
 - plattformunabhängig während der Messung ausfüllbar
 - ▶ eindeutig parsbar
 - für den Computer (einfach) erkennbare Struktur
 - ▶ robust gegenüber Fehlern des Nutzers
 - möglichst tolerant und flexibel
 - wenige, klar kommunizierte (und nachvollziehbare) Regeln
 - ▶ einfach erweiterbar
 - modularer Aufbau mit „logischer“ Ergänzung
 - Anforderungen entwickeln sich in der Praxis weiter.
- ☞ Das Medium für die Ablage der Metadaten ist sekundär (DIP).

- ▶ **Kompromiss**
 - Parsbarkeit durch den Computer ist nicht verhandelbar.
 - so einfach, intuitiv und nutzerfreundlich wie möglich
- ▶ **Kriterien**
 - möglichst wenig unnötige Schreibarbeit (*kein XML*)
 - offensichtliche, vertraute Struktur für den Nutzer
- ▶ **Fehlertoleranz**
 - zusätzlichen Leerraum (Leerzeichen etc.) ignorieren
 - ggf. Routinen zur Überprüfung schreiben, die den Nutzer frühzeitig und verständlich auf Probleme hinweisen
- ☞ Nur was sich einfach nutzen lässt, wird genutzt werden.
- ☞ Metadaten sind zu wichtig, um nicht notiert zu werden.

Open-Closed-Prinzip

offen für Erweiterungen, verschlossen gegenüber Änderungen

- ▶ Flexibilität bei gleichzeitiger Abwärtskompatibilität
 - Die Grundstruktur sollte sauber überlegt werden.
 - Abwärtskompatibilität beim Import ist zentral.
- ▶ Möglichkeit eines Freitext-Kommentars
 - Es gibt immer zusätzliche, wichtige Beobachtungen.
 - Das Kommentarfeld sollte beliebige Formatierung erlauben.
 - Tauchen immer wieder die gleichen Informationen auf, sollte das Format entsprechend erweitert werden.

Bedeutung: Daten ohne zusätzliche Informationen sind wertlos

Zielstellung: semantisches Verständnis durch Auswertungsroutinen

Kriterien für eine formalisierte Ablage von Metadaten

Ein reales Beispiel: die Infodatei

- ▶ Ausgangspunkt: „Liesmich“-Datei eines Chemotechnikers
 - elektronisches Laborbuch direkt bei den Daten
 - Beobachtungen und Informationen zur Messung

Kriterien

- ▶ maschinenlesbar und vom Menschen les- und schreibbar
 - Fokus: Les- und *Schreibbarkeit* durch den Menschen
 - Voraussetzung: (einfache) Parsbarkeit
- ▶ Reintext (ASCII-7-Bit-Zeichensatz)
 - Grund: mangelnde Unterstützung von Unicode in MATLAB
- ▶ eindeutige Identifizierbarkeit des Dateiformats
 - durch eine Kennung in der ersten Zeile

Listing 3: Beispiel für eine (minimale) Infodatei

```
common Info file - v. 0.1.0 (2014-04-04)
```

GENERAL

```
Filename:          data01
Date start:        2014-04-04
Time start:        11:05:00
Date end:          2014-04-04
Time end:          15:50:00
Operator:          Max Mustermann
Label:             My first measurement
Purpose:           Kill time
```

SAMPLE

```
Name:             xxxCry WT, Peak 2
ID:               42
Description:       Peak 2 aus der xxxCry WT Expression
Solvent:           50 mM Phosphat, 20% Glycerin
Preparation:       Frisch exprimierte Probe aus Uebernachtanzucht
```

TEMPERATURE

```
Temperature:      270 K
```

COMMENT

```
Und hier gibt's ein bisschen Freitextkommentar - aber bitte OHNE Sonderzeichen!
```

- ▶ allgemein
 - Blöcke mit Schlüssel-Wert-Paaren
 - erste Zeile zur Identifizierung von Format und Version
- ▶ Feldnamen
 - werden mit Doppelpunkt vom Feldinhalt getrennt
 - dürfen keine Doppelpunkte im Namen selbst tragen
 - Leerzeichen sind erlaubt
 - müssen mit einem Buchstaben beginnen
- ▶ Leerraum (Leerzeichen, Tabulator, ...)
 - wird ignoriert, wenn er nach Feldnamen und vor und nach Feldinhalten auftritt
 - Feldinhalte *können* vertikal ausgerichtet werden.

- ▶ Zeilenumbrüche in Feldern
 - Folgezeilen müssen mit Leerraum beginnen.
 - ▶ Kommentare auf einer Zeile
 - Alles hinter dem Kommentarzeichen wird ignoriert.
 - Kommentarzeichen selbst durch Backslash verwendbar
 - ▶ Der letzte Block ist für Kommentare vorbehalten.
 - Im Kommentar ist jegliche Formatierung erlaubt.
 - Alles ab der Blocküberschrift bis zum Dateiende wird als Kommentar interpretiert und nicht geparkt.
 - Wiederkehrende Informationen sollten als neue Felder in die Infodatei aufgenommen werden.
- ☞ offenes, flexibles, modular erweiterbares Format

- ▶ Datum und Uhrzeit für Beginn und Ende angeben
 - Messungen über den Tageswechsel bzw. mehr als 24 h
- ▶ Name des/der Messenden angeben
 - bei mehreren Namen durch Komma getrennt
- ▶ Zweck einer Messung angeben
 - hilfreich beim Verstehen einer Messung
bzw. beim Wiederfinden des richtigen Datensatzes
- ▶ Proben mit eindeutigem Bezeichner (ID) versehen
 - im einfachsten Fall eine (fortlaufende) Nummer
- ▶ Werte mit Einheit
 - Größe und Einheit durch Leerzeichen getrennt
 - sinnvollerweise Beschränkung auf SI-Einheiten

Listing 4: Beispiel für eine Metadatenfile im YAML-Format

```
---
format:
  type: measurement metadata
  version: 0.1.0

general:
  start:
    date: 2017-05-27
    time: 08:00:00
  end:
    date: 2017-05-27
    time: 09:05:00
  operator: John Doe
  purpose: kill time

sample:
  name: PC71BM
  id: 42
  solvent: toluene

temperature:
  value: 270 K

comment: >
  Spektrometer hat sich mal wieder unartig verhalten.
```



- 🔑 Daten ohne Metadaten sind wertlos.
Beide zusammen bilden eine untrennbare Einheit.
- 🔑 Die Information ist bei Datenaufnahme maximal.
Eine sinnvolle Reduktion ist die eigentliche Herausforderung.
- 🔑 Metadaten sollen den Auswertungsroutinen
ein semantisches Verständnis ermöglichen.
- 🔑 Metadaten sollten strukturiert und für Mensch
und Maschine lesbar abgelegt werden.
- 🔑 Ein Format für Metadaten sollte plattformunabhängig
und möglichst einfach nutzbar sein.