Wissenschaftliche Softwareentwicklung

Bausteine einer rechnergestützten Datenauswertung, die den Kriterien der Wissenschaftlichkeit genügt

29. Datenverarbeitung und -Analyse: selbstdokumentierend

Till Biskup
Physikalische Chemie
Universität Rostock
22.01.2026

Zentrale Aspekte

- Nachvollziehbarkeit ist essentiell für Wissenschaftlichkeit. Trotzdem ist sie in den seltensten Fällen real gegeben.
- Jeder Verarbeitungsschritt sollte vollständig mit Parametern und Version der Routine dokumentiert werden.
- Nur durch Automatisierung kann eine lückenlose Dokumentation aller Verarbeitungsschritte gewährleistet werden.
- Nachvollziehbarkeit setzt die Archivierung sowohl der Rohdaten als auch der Verarbeitungsroutinen voraus.
- Eine lückenlose und automatisierte Selbstdokumentation lässt sich nur in einem modularen Gesamtsystem realisieren.

Übersicht

Zielstellung: Nachvollziehbarkeit und Reproduzierbarkeit

Voraussetzungen: vollständiger Parametersatz und Archivierung von Daten und Routinen

Umsetzung in einem Gesamtsystem zur Datenverarbeitung

Offensichtliche Vorteile bei Verwendung des Systems

Zielstellung

Nachvollziehbarkeit und Reproduzierbarkeit

- Ausgangspunkt
  - Daten müssen (vor-)verarbeitet werden, bevor sie dargestellt und ggf. veröffentlicht werden können.
  - Art und Umfang der Verarbeitung und Analyse hängen von den Daten und der jeweiligen Aufgabenstellung ab.
- Anspruch
  - Weg von den Rohdaten zum Ergebnis der Auswertung vollständig transparent, nachvollziehbar und reproduzierbar
  - erfordert lückenlose Dokumentation sämtlicher Verarbeitungs- und Auswertungsschritte

☛ Nachvollziehbarkeit und Reproduzierbarkeit sind wesentliche Aspekte der Wissenschaftlichkeit.

Reproduzierbarkeit und Replizierbarkeit

Ein (auf den ersten Blick) subtiler Unterschied

- Reproduzierbarkeit (*reproducibility*)
  - vollständige Wiederholbarkeit einer beschriebenen Datenverarbeitung und -Analyse
  - Ausgangspunkt sind existierende Daten
  - sollte in jedem Fall möglich sein
- Replizierbarkeit (*replicability*)
  - unabhängige Wiederholung der (Roh-)Datenerhebung, meist in Form von Experimenten und Beobachtungen
  - nicht in jedem Fall durchführbar

☛ Replizierbarkeit hängt von den Gegebenheiten ab.

☛ Reproduzierbarkeit (und damit Nachvollziehbarkeit) liegt in der Verantwortung jedes einzelnen Wissenschaftlers.

Das Problem der wissenschaftlichen Praxis

Nachvollziehbarkeit und Reproduzierbarkeit sind meist nicht gegeben.

These

Die vollständige Nachvollziehbarkeit und Reproduzierbarkeit der Datenauswertung, obwohl essentiell für die Wissenschaftlichkeit, ist in den meisten Fällen nicht gegeben.

(mögliche) Gründe

- mangelndes Bewusstsein für die Wichtigkeit
- fehlende Automatisierung von Prozessabläufen, die gleichzeitig für Dokumentation sorgt
- Mangel an Zeit und Fähigkeiten zur Entwicklung eines Systems zur Sicherstellung der Ansprüche

Eine mögliche Lösung

Ein modulares, einfaches System zur Datenverarbeitung

- modular
  - Anforderungen entwickeln sich.
  - Jeder Datensatz und jede Fragestellung ist anders.
  - Viele Verarbeitungsschritte sind gleich oder ähnlich.
  - Das Gesamtsystem lässt sich nur schrittweise entwickeln.
- einfach
  - Nur ein einfach nutzbares System wird real genutzt.
  - Gute Abstraktionen führen zu intuitiver Bedienbarkeit.
- umfassend
  - reicht von der Datenaufnahme bis zur Publikation
  - umfasst unterschiedliche Plattformen und Medien
  - Vorteil nur durch die konsequente Anwendung

☛ entscheidend: Bewusstsein und persönliche Einstellung

Übersicht

Zielstellung: Nachvollziehbarkeit und Reproduzierbarkeit

Voraussetzungen: vollständiger Parametersatz und Archivierung von Daten und Routinen

Umsetzung in einem Gesamtsystem zur Datenverarbeitung

Offensichtliche Vorteile bei Verwendung des Systems

Voraussetzungen

Vollständiger Parametersatz und Archivierung von Daten und Routinen

- vollständiger Parametersatz jedes Verarbeitungsschrittes
  - soll die (automatische) Wiederholung ermöglichen
  - eigentliche Parameter des Verarbeitungsschrittes, explizit (durch Nutzer angegeben) und implizit (voreingestellt)
  - Information über die Version der verwendeten Software
- Archivierung von Daten und Routinen
  - (Roh-)Daten sollten für Jahrzehnte archiviert werden.
  - Auswertungssoftware sollte genauso lange zugreifbar sein.
  - Die konkret verwendete Version der Software ist essentiell.
  - Code sollte lesbar und ausdrucksstark sein (*Clean Code*).
- konsequente Anwendung des Systems
  - essentiell für lückenlose Dokumentation
  - möglichst weitgehende Automatisierung aller Abläufe
  - Voraussetzungen: Nutzbarkeit und Plattformunabhängigkeit

Was ist ein vollständiger Parametersatz?

Hinweise aus der eigenen langjährigen Praxis

- alle (expliziten) Parameter des Verarbeitungsschritts
  - strukturiert: alle Parameter einer Verarbeitungsroutine
  - objektorientiert: Parameter in einer Instanzvariable (ggf. als assoziatives Datenfeld abgelegt)
- alle *impliziten* Parameter des Verarbeitungsschritts
  - Mitunter werden Parameter auf Standardwerte gesetzt.
  - Voreingestellte Standardwerte können sich ändern.
  - Implizite Parameter sollten explizit dokumentiert werden.
- Datum und Uhrzeit
  - Zeitstempel der Ausführung des Verarbeitungsschritts
  - ggf. in UTC und in einem Standardformat ablegen
  - ermöglicht ggf. Rückschluss auf die (mitunter kritische) Reihenfolge von Verarbeitungsschritten

Was ist ein vollständiger Parametersatz?

Hinweise aus der eigenen langjährigen Praxis

- Durchführender
  - hilfreich für Nachvollziehbarkeit und bei Rückfragen
  - Nutzername oder (idealerweise) der Klarname
  - Datenschutz beachten: Nutzer des Systems *vorher* in Kenntnis setzen, dass und was gespeichert wird
- Version der Routine/des Programms
  - eindeutiger Bezeichner (klares Versionsnummernschema)
  - setzt Nutzung eines Versionsverwaltungssystems voraus
  - Entwicklerversionen *nie* produktiv einsetzen
- Plattform
  - Betriebssystem inkl. Version
  - Version der zugrundeliegenden Programmiersprache
  - Name und Version verwendeter Bibliotheken
  - idealerweise auch die Prozessorarchitektur

Archivierung von Daten und Routinen

Anforderungen an eine entsprechende Infrastruktur

- Daten
  - Voraussetzung: System zur Ablage von (Roh-)Daten
  - eindeutige Bezeichner (IDs) für jeden Datensatz
  - Zugriff auf die Daten nur über ihre IDs
  - Sicherstellung der Unversehrtheit der Daten
- Routinen
  - liegen in einem Versionsverwaltungssystem
  - Zugriff auf die jeweils verwendete Version möglich
  - Entwicklerversionen *nie* produktiv einsetzen
    (keine Eineindeutigkeit der Versionsnummer gewährleistet)
  - zusätzliche Bibliotheken ggf. mit sichern

☛ Entwicklung von Hardware und Betriebssystemen erschwert die Reproduzierbarkeit nach Jahren.

Übersicht

Zielstellung: Nachvollziehbarkeit und Reproduzierbarkeit

Voraussetzungen: vollständiger Parametersatz und Archivierung von Daten und Routinen

Umsetzung in einem Gesamtsystem zur Datenverarbeitung

Offensichtliche Vorteile bei Verwendung des Systems

Umsetzung

Im Kontext eines Gesamtsystems zur Datenverarbeitung

- Datenverarbeitung ist nur ein Teil des Gesamtsystems.
  - weitere Aspekte: Datenaufnahme, Datenablage und Auswertung bzw. Darstellung der Analysen
  - Diese Aspekte wurden und werden separat behandelt.
- Gesamtsysteme werden schnell komplex.
  - lassen sich nicht „schnell mal nebenher“ entwickeln
  - umfassen unterschiedliche Plattformen und Medien
  - Voraussetzungen: Modularität, Flexibilität, Nutzbarkeit
- Datenverarbeitung ist in einer Anwendung umsetzbar.
  - Programmiersprache, -Paradigma und Plattform sind egal.
  - wichtig: Grundkonzepte und konsequente Anwendung

☛ Viele Aspekte lassen sich leicht umsetzen.

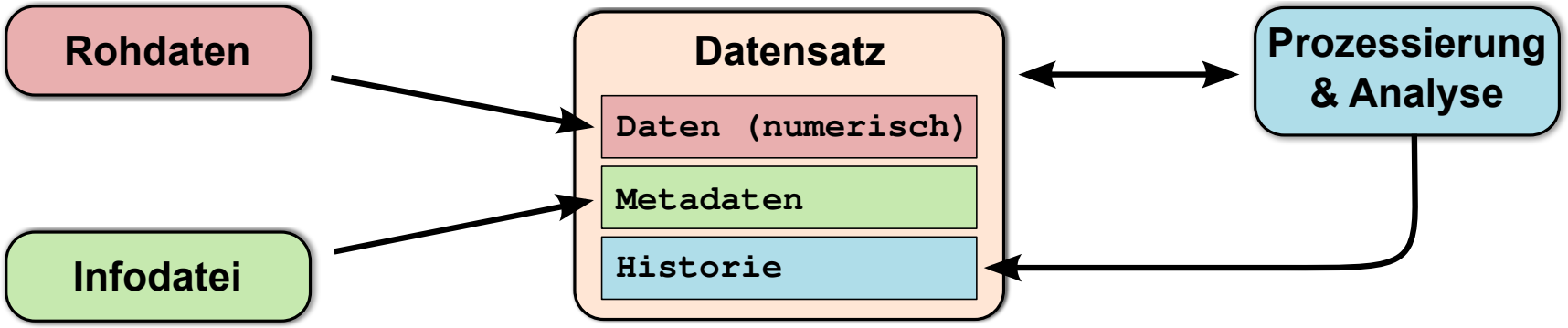
Grundregeln

Was jedes System zur Datenverarbeitung beachten sollte

- so weit wie möglich automatisieren
  - sorgt für Konsistenz
  - hilft bei der lückenlosen Dokumentation
  - erleichtert die Reproduzierbarkeit
  - ermöglicht bequeme Variation einzelner Parameter
- Skripte zur Verarbeitung versionieren und archivieren
  - Voraussetzung: (verteiltes) Versionsverwaltungssystem
  - sicherstellen, dass das gespeicherte Ergebnis konsistent mit dem archivierten Skript ist
- Interaktive Systeme müssen jeden Schritt protokollieren.
  - typisches Beispiel: grafische Nutzerschnittstelle (GUI)
  - Trennung von Schnittstelle (UI) und Verarbeitungsroutinen
  - GUIs lassen sich (bequem) skripten.

Ein datensatzzentriertes System

Überblick über die Komponenten und Ideen



- Ein Datensatz besteht aus drei Komponenten.
  - numerische Daten
  - Metadaten (meist aus einer Infodatei)
  - Historie der Verarbeitungs- und Analyseschritte
- Verarbeitungsroutinen operieren auf Datensätzen.
  - Jeder Schritt wird in der Historie protokolliert.
  - Das Protokoll umfasst einen vollständigen Parametersatz.

Ein datensatzzentriertes System

Ideen zur konkreten Umsetzung

- Verarbeitungsschritte
  - einheitliches Schema für die Parameter (z.B. assoziatives Datenfeld mit festem Namen)
  - von einer generischen Routine ansprechbar
  - können Eintrag für Historie schreiben bzw. die notwendigen Parameter bereitstellen
- Historie
  - geordnete Liste der Verarbeitungsschritte, jeweils mit vollständigem Parametersatz
  - Information über Reversibilität jedes Schrittes
- Datensatz
  - Originaldaten in separatem Feld ebenfalls speichern
  - erlaubt „Undo“-Funktionalität gemeinsam mit Historie

Ein datensatzzentriertes System

Mögliche Umsetzung einer Undo-Funktionalität

Listing 1: Beispiel für die Undo-Funktionalität (Python)

```python
def undo(self):
    if not self.history[self._history_pointer].undoable:
        raise UndoStepUndoableError
    self._history_pointer -= 1
    self.data = self._origdata
    for historyentry in self.history[:self._history_pointer]:
        historyentry.replay(self)
```

- Originale Daten (Rohdaten) sind verfügbar.
- Verarbeitung erfolgt über eine generische Routine.

☛ „Redo“ lässt sich ebenso einfach implementieren.

☛ Verzweigte Historie kann schnell komplex werden . . .

Verarbeitung mehrerer Datensätze

Mehrere denkbare Szenarien

- datensatzzentriertes Szenario
  - Mehrere Datensätze werden miteinander verbunden.
  - Die Historie speichert die Verarbeitungsschritte, auf die Datensätze wird über deren ID verwiesen.
- darstellungszentriertes Szenario
  - Das Ergebnis ist eine Abbildung oder Tabelle.
  - Die dargestellten Daten stellen Charakteristika einer Reihe von Datensätzen gegenüber bzw. in einen Zusammenhang.
  - Protokoll für die Gewinnung der Daten aus Datensätzen
- prozessschrittzentriertes Szenario
  - Verarbeitungsschritt auf andere Datensätze anwendbar
  - einzelne Parameter des Verarbeitungsschritts variierbar

☛ Alle Szenarien sollten vom System unterstützt werden.

Der Wert abstrakter Schnittstellen

Metadaten zu jedem Szenario lassen sich als Textdateien speichern

- Anwendungsfälle lassen sich formal beschreiben.
  - Liste von Datensatz-IDs, auf denen operiert werden soll
  - Liste von (Vor-)Verarbeitungsschritten
  - Liste von Nachbearbeitungsschritten
- universelle Anwendbarkeit
  - Verarbeitung eines einzelnen Datensatzes
  - Erstellung von Repräsentationen (Abbildungen, Tabellen)
  - Erzeugen von Ergebnissen/Ergebnislisten aus Analysen
- Formale Beschreibungen lassen sich einfach persistieren.
  - Reintextformat, idealerweise standardisiert (z.B. YAML)
  - Ablage beim Datensatz oder in Verzeichnisstruktur
  - Historie als Liste von Metadaten-Dateien darstellbar

☛ generische Routinen zum Verarbeiten dieser Metadaten

Der Wert abstrakter Schnittstellen

Beispiel: Metadaten für die (komplexe) Datenverarbeitung

Listing 2: „Rezept“ zur Datenverarbeitung

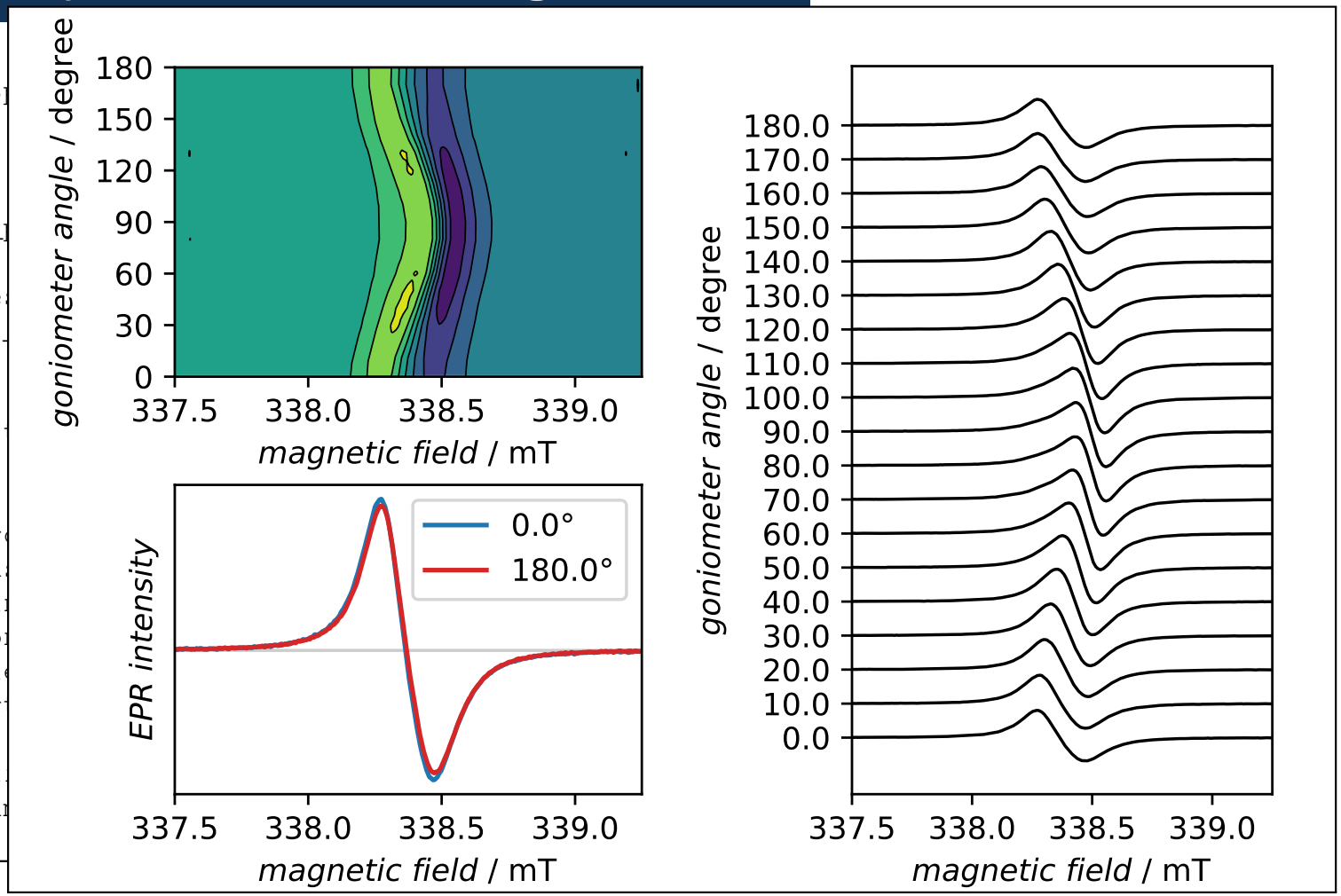
```
format:
  type: ASp
  version:

settings:
  default_p

directorie
  datasets_

datasets:
  - magnet

tasks:
- kind: pr
  type: Ba
- kind: si
  type: Go
  properti
    proper
      axes
        xl
    filenar
```



M. Schröder, T. Biskup. *J. Magn. Reson.* **335**:107140, 2022

Übersicht

Zielstellung: Nachvollziehbarkeit und Reproduzierbarkeit

Voraussetzungen: vollständiger Parametersatz und Archivierung von Daten und Routinen

Umsetzung in einem Gesamtsystem zur Datenverarbeitung

Offensichtliche Vorteile bei Verwendung des Systems

Offensichtliche Vorteile

Motivation zur Verwendung eines Systems zur Datenverarbeitung

- Sicherstellung wissenschaftlicher Qualitätsstandards
  - vollständige Transparenz und Nachvollziehbarkeit
  - in der Regel Reproduzierbarkeit der Verarbeitungsschritte
- automatische lückenlose Dokumentation
  - immense Arbeitsersparnis
  - Information jederzeit verfügbar
  - Möglichkeit der automatischen Berichterstellung (Reports)
- Arbeitserleichterung und -Ersparnis
  - gleiche Verarbeitung unterschiedlicher Datensätze für den direkten Vergleich
  - einfache Variation einzelner Parameter und Analyse des Einflusses auf das Ergebnis

Grundlage für weitere Funktionen

Ideen und Ausblick

- automatisierte Erzeugung von Abbildungen
  - Sichten auf Daten lassen sich abstrakt formulieren.
  - Automatisierung erspart Arbeit und sorgt für Konsistenz.
- Berichte
  - Überblick über alle Informationen zu einem Datensatz
  - sowohl Metadaten als auch Verarbeitungsschritte
  - einfach auf mehrere Datensätze ausweitbar
- Verknüpfung von Repräsentationen und Daten
  - Ziel: Zugriff auf die Daten aus den Repräsentationen
  - analog: eingebettete Excel-Tabelle in Word-Dokument

☛ Wird (teilweise) in der nächsten Vorlesung thematisiert.

☛ modulares, flexibles System: später implementierbar

Zentrale Aspekte

- Nachvollziehbarkeit ist essentiell für Wissenschaftlichkeit. Trotzdem ist sie in den seltensten Fällen real gegeben.
- Jeder Verarbeitungsschritt sollte vollständig mit Parametern und Version der Routine dokumentiert werden.
- Nur durch Automatisierung kann eine lückenlose Dokumentation aller Verarbeitungsschritte gewährleistet werden.
- Nachvollziehbarkeit setzt die Archivierung sowohl der Rohdaten als auch der Verarbeitungsroutinen voraus.
- Eine lückenlose und automatisierte Selbstdokumentation lässt sich nur in einem modularen Gesamtsystem realisieren.