Modellierung von Datenstrukturen und Datei-Inhalten

Data Dictionary und Syntaxdiagramm

Stand 05.09.2017

Data Dictionary

- Verzeichnis, das Informationen über Daten bzw. Dateiinhalte enthält:
  - Struktur,
  - Eigenschaften
  - Verwendung
- Einsatz:
  - zur Konsistenzüberwachung eines Datenbestandes
  - Übersichten über die Datenstrukturen
  - Überprüfung auf Redundanz- und Widerspruchsfreiheit

Data Dictionary

- Zeitpunkte:
  - Definitionsphase und Analyse
    - entsteht meist in der Definitionsphase
    - Analyse einer **vorhandenen** Datei bzw. Struktur
    - wird in der Entwurfs- und Implementierungsphase weiter verwendet, ergänzt und verfeinert
  - Entwurf
    - Definition einer **noch nicht vorhandenen** Datei bzw. Struktur

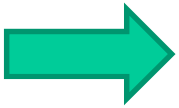
Data Dictionary

Symbole

Symbol	Beschreibung	Beispiel
„=“	„ist äquivalent zu“	A = B
„+“	Sequenz, Verknüpfungen von Datenflüssen bzw. Datenelementen	A = B + C
[|]	Alternativen, Auswahl (entweder ... oder)	A = [B | C]
{ }	Wiederholungen	A = { B }
M{ }N	Wiederholung von M bis N	A = 1{ B }10
( )	optionale Eingaben	A = B + (C)
„*text*“	fest vorgegebener Text	A = "abc"
/* … */	Kommentar	A = X + Y /*Kommentar*/

Data Dictionary

- Wie wird am besten modelliert?



- Top-Down vorgehen!
- **Modularisierung** zur Wiederverwendung, d.h. kleinere oder größere Einheiten bilden und diese dann zusammensetzen

Data Dictionary

Beispiel: Kundendaten

Symbol	Beschreibung	Beispiel
„=“	„ist äquivalent zu“	A = B
„+“	Sequenz, Verknüpfungen	A = B + C
[|]	Alternativen, Auswahl	A = [B | C]
{ }	Wiederholungen	A = { B }

Symbol	Beschreibung	Beispiel
M{ }N	Wiederholung von M bis N	A = 1{ B }10
( )	optionale Eingaben	A = B + (C)
„text“	fest vorgegebener Text	A = "abc"
/* … */	Kommentar	A = X + Y /*Kom.*/

Kundendatei = { Kundeneintrag }

Kundeneintrag = Kunden-Nr. + Name + Adresse + (Geburtsdatum)
+ (Funktion) + Umsatz

Name = Anrede + (Titel) + Vorname + Nachname

Adresse = [Straße + Haus-Nr. | Postfach] + (Länderkennzeichen)
+ PLZ + Ort + (Telefon) + (Fax) + (e-Mail)

Geburtsdatum = . . .

Aufgabe 1 zu Data Dictionary

Durch ein Softwareprodukt sollen viele Grafikelemente importiert und exportiert werden. Dabei soll eine lesbare Textdatei erstellt bzw. gelesen werden. Für die Grafikelemente Punkt, Linie und Kreis sieht diese Datei folgendermaßen aus:

```
/START_DESC;
/== Dies ist ein Kommentar ==!/
#1 = PUNKT( 1.0, 1.0 );
#2 = PUNKT( 2.0, 2.0 );
#3 = LINIE( #1, #2 );
#4 = PUNKT( 2.0, 2.0 );
#5 = KREIS( #1, #2, #4 );
/END_DESC;
```

- Die mit „#“ kombinierten Zahlen stellen Referenzen auf Elemente dar.
- Die Datei beginnt mit einer Startkennung und endet mit einer Ende-Kennung.
- Leerzeichen können beliebig zwischen den Elementen in der Datei stehen, sie müssen bei der Definition nicht berücksichtigt werden
- Der Datentyp FLOAT muss nicht näher beschrieben werden
- Die Geometrieelemente können mit Namen definiert werden (z.B.: “PUNKT“)

Lösungsvorschlag (Grafikelement-Datei als DD)

/* Data Dictionary zu Grafikelemente-Datei */

Grafikelemente-Datei = START + ({ [Grafikelement | Kommentar] }) + ENDE

START = "/START_DESC;„
ENDE = "/END_DESC;"

Kommentar = "/==" + Text + "==!/"

Grafikelement = Adresse + "=" + Element-Definition + Zeilenende

Element-Definition = [Punkt | Linie | Kreis]

Punkt = "PUNKT" + "(" + FLOAT + "," + FLOAT + ")"
Linie = "LINIE" + "(" + Adresse + "," + Adresse + ")"
Kreis = "KREIS" + "(" + Adresse + "," + Adresse + "," + Adresse + ")"

Adresse = "#" + Nummer

Nummer = { Ziffer }

Text = ({ [Buchstabe | Ziffer | Sonderzeichen] })

Buchstabe = ["a" | "b" | ... | "Y" | "Z"]

Ziffer = [0 | 1 | ... | 8 | 9]

Sonderzeichen = ["_" | " " | "!" | ...]

Zeilenende = ";"

Aufgabe 2 zu Data Dictionary

Beschreiben Sie die Struktur einer allgemeinen XML-Datei mit Hilfe von **Data Dictionary** oder **Syntaxdiagramm.**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<verzeichnis>
    <titel>Wikipedia Staedteverzeichnis</titel>
    <eintrag>
        <stichwort>Genf  Att1=XXX  Att2=YYY</stichwort>
        <eintragstext>Genf ist der Sitz von ...</eintragstext>
    </eintrag>
<!-- Kommentar-Text -->
    <eintrag>
        <stichwort>Bonn</stichwort>
        <eintragstext>Bonn ist eine Stadt, die ...</eintragstext>
    </eintrag>
</verzeichnis>
```

Lösungsvorschlag (XML-Datei als Data Dictionary)

/* Data Dictionary einer XML-Datei (vereinfacht) */
/* XML-Kommentare können mehrfach hintereinander auftreten => ({ XML_Kommentar }) */
XML_Datei = XML_Deklaration + XML_Root_Element

XML_Deklaration = ‘ *"<?xml version="1.0" encoding="UTF-8" standalone="yes"?>* ‘
XML_Kommentar = "<!--" + Text + "-->„

XML_Root_Element = XML_Element /* Die Root-Hierarchie kann auch weggelassen werden */
XML_Element = Element_Start + XML_Element_Body + Element_Ende

Element_Start = "<" + Element_Name + (AttributListe) + ">"
Element_Ende = "</" + Element_Name + ">„

AttributListe = { Element_Attribut + " " } /* Elementattribute durch Leerzeichen getrennt */
Element_Attribut = Name + (Attributwert)
Attributwert = „‘“ + Name + „‘“ /* bzw. { [Ziffer | Buchstabe] } ; auch doppelte Anf.-zeichen */

XML-Element_Body = ({ XML_Element | XML_Kommentar })

Element_Name = Buchstabe + (Name)
Name = { [Ziffer | Buchstabe] }

Text = ({ [Buchstabe | Ziffer | Sonderzeichen] })

Buchstabe = ["a" | "b" | ... | "Y" | "Z"]

Ziffer = [0 | 1 | ... | 8 | 9]

Sonderzeichen = ["_" | " " | "!" | ...]

Syntaxdiagramm

- Grafische Darstellung von Datenstrukturen
- Symbole:

Element (wird noch detailliert)

Element (wird nicht weiter spezifiziert)

Verbindung („Fluss“)

- Grundsätzliche Bedingungen:

Genau 1 Element

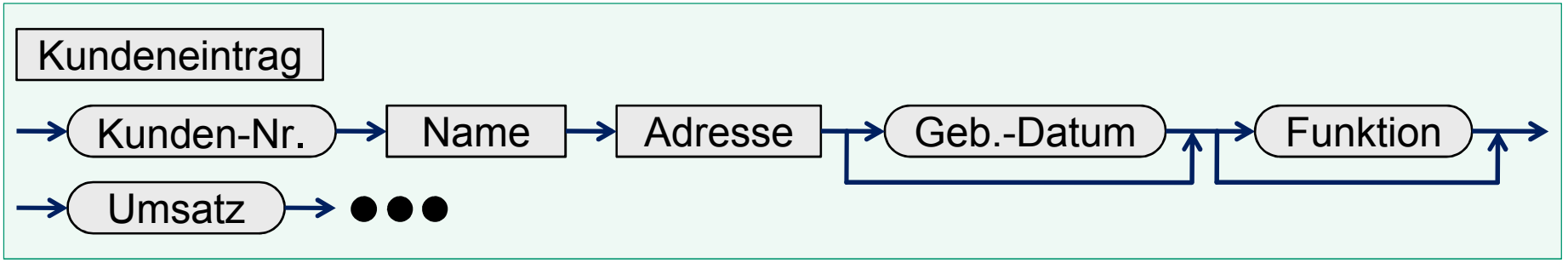
Ein oder viele Elemente

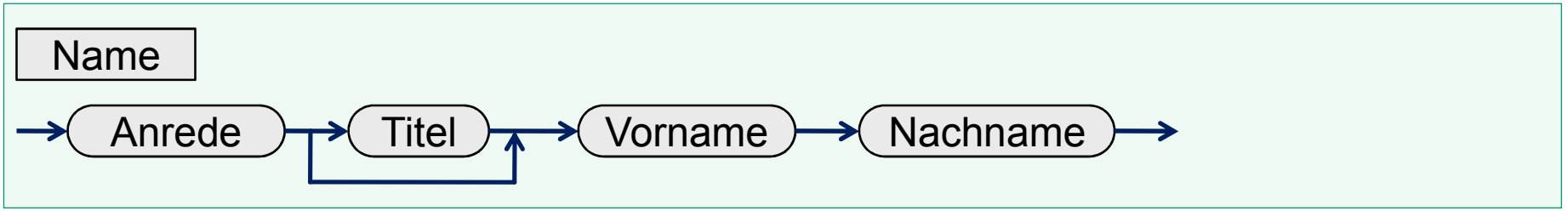
Ein oder kein Element

Keine oder viele Elemente

Syntaxdiagramm

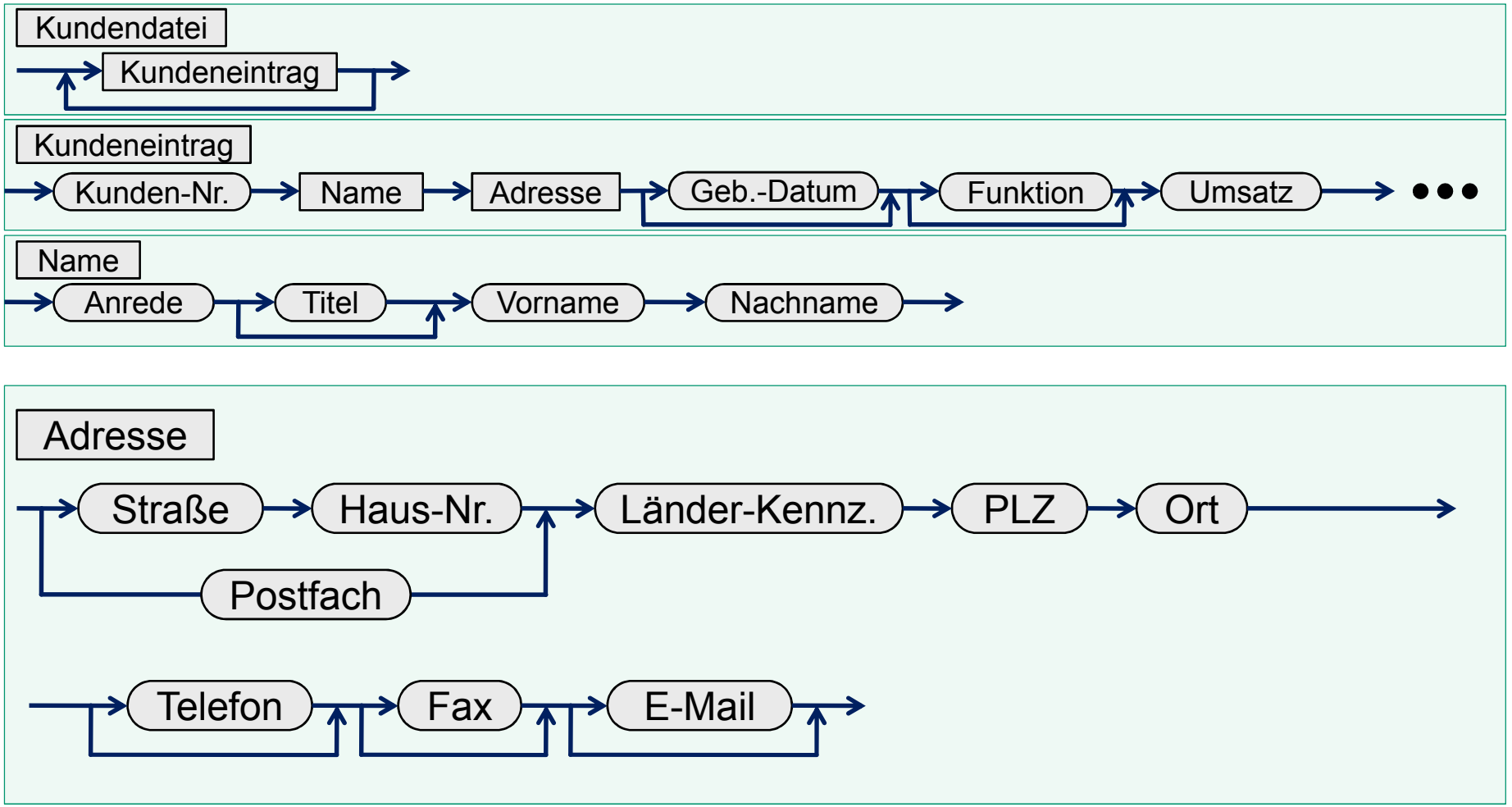
Beispiel: Kundendaten





Syntaxdiagramm

Beispiel: Kundendaten (Forts.)



Aufgabe zu Syntaxdiagramm und Data Dictionary

In einem Eingabefenster einer grafischen Benutzeroberfläche sollen Telefonnummern in unterschiedlichen Formaten eingegeben werden können.

Diese Eingaben sind auf syntaktische Richtigkeit zu überprüfen. Hierfür ist eine Beschreibung der Syntax aller zulässigen Nummernarten erforderlich.

Hinweis: Das Zeichen "#" stellt ein Leerzeichen dar

```
07247#825743
++49#7247#825743
++49/7247/825743
0721/934567
++44/34567334556
```

Aktuelles Beispiel zu Syntax-Diagramm und Data Dictionary

- SQL-Dokumentation zu Oracle 10g:

  http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7002.htm