



# 7437 - EDI und E-Business Standards

Praktikumsaufgabe 07  
Bestellungen und Lieferavis mit  
UN/EDIFACT und EANCOM,  
Mapping per EDI-Bibliothek  
(eingehende Daten)



## Das Szenario



- Szenario
  - Die Praktikumssteilnehmer setzen ihre Handelspartner-Rolle (Lieferanten, Händlern) der Konsumgüterbranche aus Praktikum 01/02 fort.
  - Im Unterschied zur Übung 06 steht nun die Verarbeitung eingehender UN/EDIFACT-Daten im Mittelpunkt.
- Ziele der Übung
  - Umgang mit eingehenden UN/EDIFACT-Daten
  - Selektion von Segmenten / Segmentgruppen
  - Mappingtechnik: Vertiefung



- Die Aufgabe
  - Mappen Sie Ihre Belege aus Übung 06 (Händler: ein eingehendes Lieferavis, Lieferanten: eine Bestellung)
    - von EANCOM '02 gemäß des Nachrichtenaufbaus Ihres Handelspartners
    - in Ihre Inhouse-Struktur aus Übung 03
  - Programm-Name:  
`mapper07.rb`
  - Gewünschter Aufruf:  
`mapper07.rb srcdata.edi > fixreodata.msg`



- Hinweise zum Ablauf
  - Besorgen Sie zunächst das *Interchange*
    - Das *Interchange* stammt von Ihrem Handelspartner. Es stellt das Ergebnis von Übung 06 dar.
  - Benutzen Sie als Zielstruktur Ihre *Inhouse*-Struktur aus Übung 03



- Dateikopf (Einbindung der Bibliothek):

```
#!/usr/bin/env ruby
require "rubygems"
require_gem "edi4r"
```
- Einlesen eines EDIFACT Interchange:

```
1) ic=EDI::Interchange_E.parse(File.open(fname, 'r'))
2) ic=EDI::Interchange_E.parse($stdin)
mapping = Inbound_maps.new # Controllerklasse
```
- EDIFACT Interchange nachrichtenweise verarbeiten:

```
ic.each do |msg|
  mapping(msg) # schreibt nach stdout
end
```



- Selektion der eigentlichen Konverter-Methode:
  - Die Methode *mapping* wertet aus:
    - `msg.root.header.cS002` (Interchange Sender)
    - `msg.root.header.cS003` (Interchange Recipient)
    - `msg.header.cS009` (Message Identifier)
  - Aus diesen Angaben wird die zuständige Konverter-Methode bzw. -Variante ermittelt. Diese wird dann aufgerufen.
  - Hier kann dieses Verfahren stark vereinfacht werden.
- Kern einer Konvertermethode:

```
msg.each do |seg|
  seg_name = seg.name
  seg_name += ' ' + seg.sg_name unless seg.sg_name.empty?
  case seg_name
  # ... Viele when-Blöcke, s.u.
  end
end
```



- when-Konstrukte einer Konvertermethode:

```
when 'DTM' # Selektiert DTM außerhalb einer SG
  cde = seg.cC507      # Hilfsvariable, optional
  # Std-Code mit Hash auf hauseigene Codes mappen:
  a = myDTMcode[cde.d2005]
  # Datum und ggf. Uhrzeit auslesen
  case cde.d2379
  when '203', '102'
    cde.d2380 =~ /^(\d{8})(\d{4})?/
    mydate = $1
    mytime = $2 unless $2.nil?
  when ... # etc.
  else
    raise "Falsches Format in DTM, DE=C502/2379"
  end
```



- Selektion eines Segments einer Segmentgruppe:

```
when 'NAD SG1' # Selektiert NAD der SG1
  # ...
when 'CTA SG2' # Selektiert CTA in SG2 (in SG1)
  # ...
when 'NAD SG4' # Selektiert NAD der SG4
  # ...
when 'NAD'      # FEHLER: NAD nie außerhalb einer SG!
  # (Kein Abbruch, aber: Trifft nie ein!)
```



- Umgang mit (großen) Segmentgruppen:

```
when 'LIN SG28' # Trigger-Segment?  
  item_mapper(seg) # an spezielle Methode geben  
  # ...  
  
# In item_mapper-Methode:  
tseg["descendant-or-self:*"].each do |seg|  
  # case aufsetzen, wie in Hauptmethode  
end
```

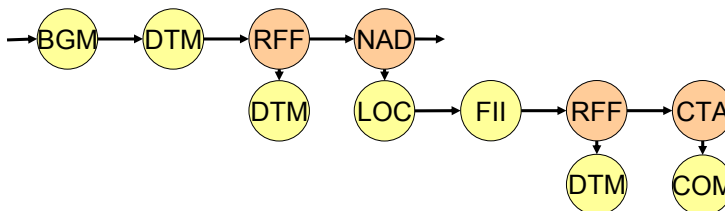
- Bemerkungen

- Selektion einer Teilmenge der Segmente: XPath-artig!
- Verteilung auf mehrere Methoden fördert Modularisierung.



- Umgang mit (großen) Segmentgruppen:

- Selektion einer Teilmenge der Segmente: XPath-artig!
- Verteilung auf mehrere Methoden fördert Modularisierung.
- Zerlegung der "Seitenketten" in Einzelketten, je eine Methode pro Seitenkette.
- Konzept:
  - Es gibt einfache Knoten und T-Knoten (Knoten = Segment hier)
  - Vernetzung über die (realen) T-Knoten
  - Segmentgruppen bleiben ein virtuelles Konstrukt (nur Attribut-Info)





- Hinweise zum Ablauf (Forts.)
  - Alle eingehenden Nutzdaten sind zu mappen,
  - Nicht mehr benötigte Qualifier sowie Inhalte der Service-Segmente müssen Sie nicht in Ihr Inhouse-Format übernehmen.



- Abzugeben
  - `mapper07.rb` # Ihr Programm-Code
  - `srcdata.edi` # Ihre verwendeten Quelldaten
  - `fixreccdata.msg` # Ihre Ergebnisse  
(`srcdata.edi` sollte von Ihrem Handelspartner stammen, `fixreccdata.msg` erzeugen Sie nach eigener Definition)
- Abgabeordner
  - Wie üblich, unter `~werntges/lv/edi/abgaben/a/<matnrn>`
- Annahmeschluss
  - Donnerstag der **ü**bernächsten Übung (bis 12 Uhr).