



7437 – EDI und E-Business Standards, 4661 – E-Business: Standards und Automatisierung

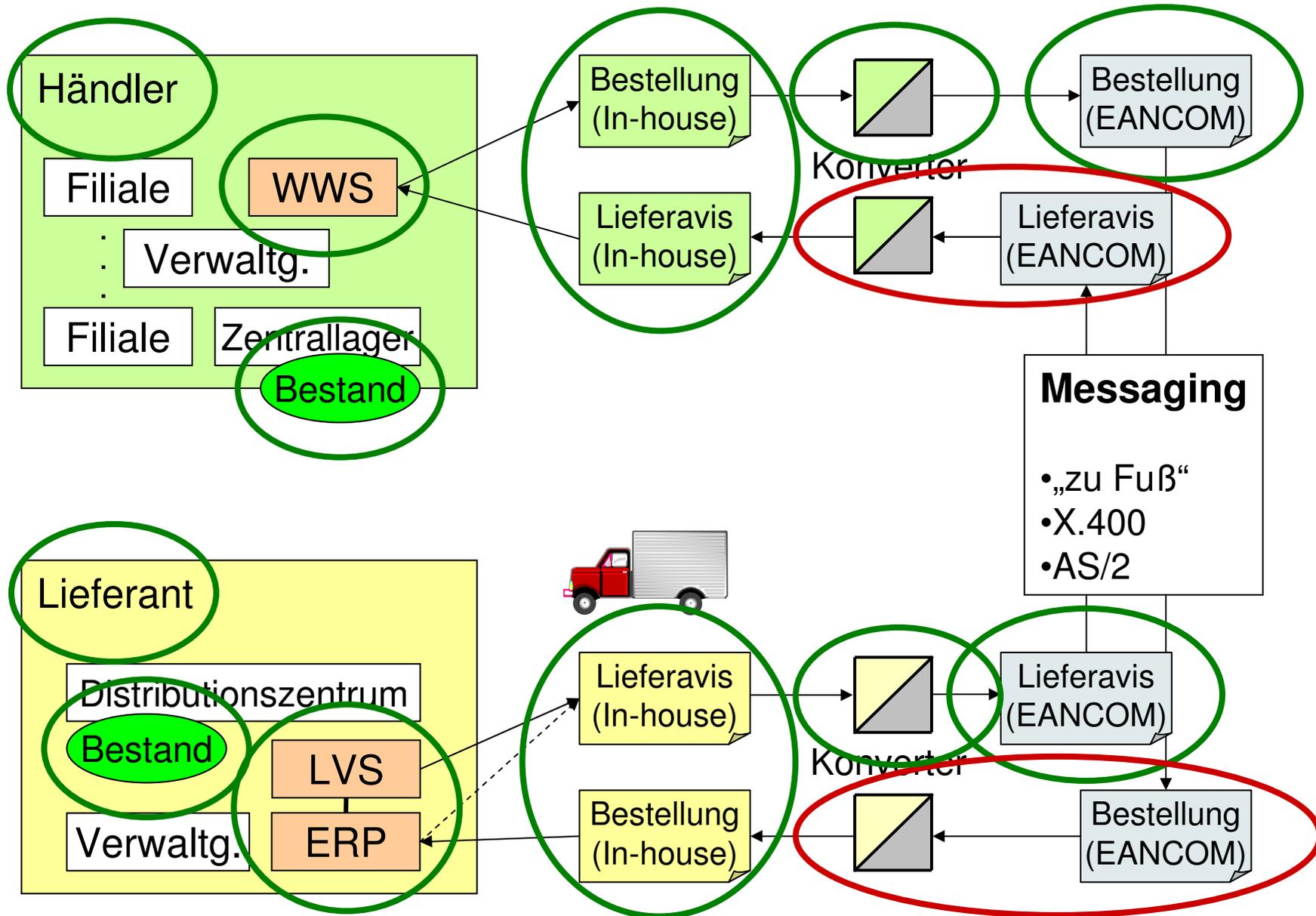
Praktikumsaufgabe 07:
Bestellungen und Lieferavise mit
UN/EDIFACT und EANCOM,
Mapping per EDI-Bibliothek
(eingehende Daten)



- Szenario
 - Die Praktikumsteilnehmer 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



Das Praktikumsprojekt





- Die Aufgabe
 - Falls Sie bisher noch kein eigenes Inhouse-Format für eingehende Nachrichten definiert haben (Vgl. Übung 03), so holen Sie dies zunächst nach.
 - 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 (Beispiel):
mapper07.rb srcdata.edi > fixrecdata.msg



- Hinweise zum Ablauf
 - Besorgen Sie zunächst *Interchanges*
 - Die *Interchange* stammen von Ihren Handelspartnern. Sie stellen das Ergebnis von Übung 06 dar.
 - Mindestens ein *Interchange* sollte vorliegen!
 - Benutzen Sie als Zielstruktur Ihre *Inhouse*-Struktur aus Übung 03



- Dateikopf (Einbindung der Bibliothek):

```
#!/usr/bin/env ruby
require "rubygems"
require_gem "edi4r" # Manchmal nur: require "edi4r"
require "edi4r/edifact"
```

- Einlesen eines EDIFACT Interchange:

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

- EDIFACT Interchange nachrichtenweise verarbeiten:

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



- Selektion der zuständigen (!) 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 für die Kombination aus Sender, Empfänger und Nachrichtentyp 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  
# ...
```

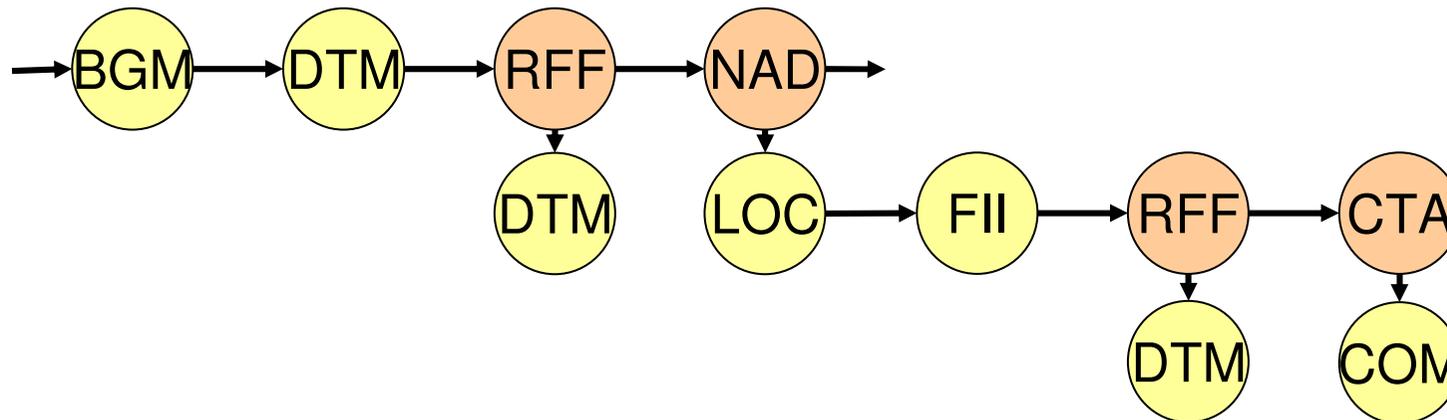
```
# In item_mapper-Methode:
```

```
tseg.descendants_and_self 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 (hier: Knoten = Segment)
 - 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.

- Im „Notfall“:
 - Sollte keiner Ihrer Handelspartner Ihnen ein *Interchange* zur Verfügung stellen, dann gehen Sie wie folgt vor:
 - Erzeugen Sie ein solches Interchange per Editor, vgl. Übung 05
 - Halten Sie sich dabei eng an die Vorgaben von EANCOM



- Abzugeben

- `mapper07.rb` # Ihr Programm-Code
- `srcdata.edi` # Ihre verwendeten Quelldaten
- `fixrecdata.msg` # Ihre Ergebnisse

(`srcdata.edi` sollte von einem Ihrem Handelspartner stammen, `fixrecdata.msg` erzeugen Sie nach eigener Definition; Format ist frei wählbar, auch wenn der Name „fixrec“ suggeriert...)

- Abgabeordner

- Wie üblich, unter `~werntges/lv/edi/abgaben/a/<matrnr>`

- Annahmeschluss

- Normal, also: Dienstag vor der nächsten Übung.
- Bem.: Wegen der Weihnachtspause ist dies erst in 3 Wochen.