



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

**E**lectronic

**D**ata

**I**nterchange

(Elektronischer Datenaustausch)



# Klassisches EDI - der Kern

Einleitung - die Kernkomponenten  
File Transfer- und Messaging-Standards  
UN/EDIFACT und EANCOM im Detail  
Applikationsschnittstellen  
**Konverter- und Mappingtechniken**



# EDI: Die Kernkomponenten

---

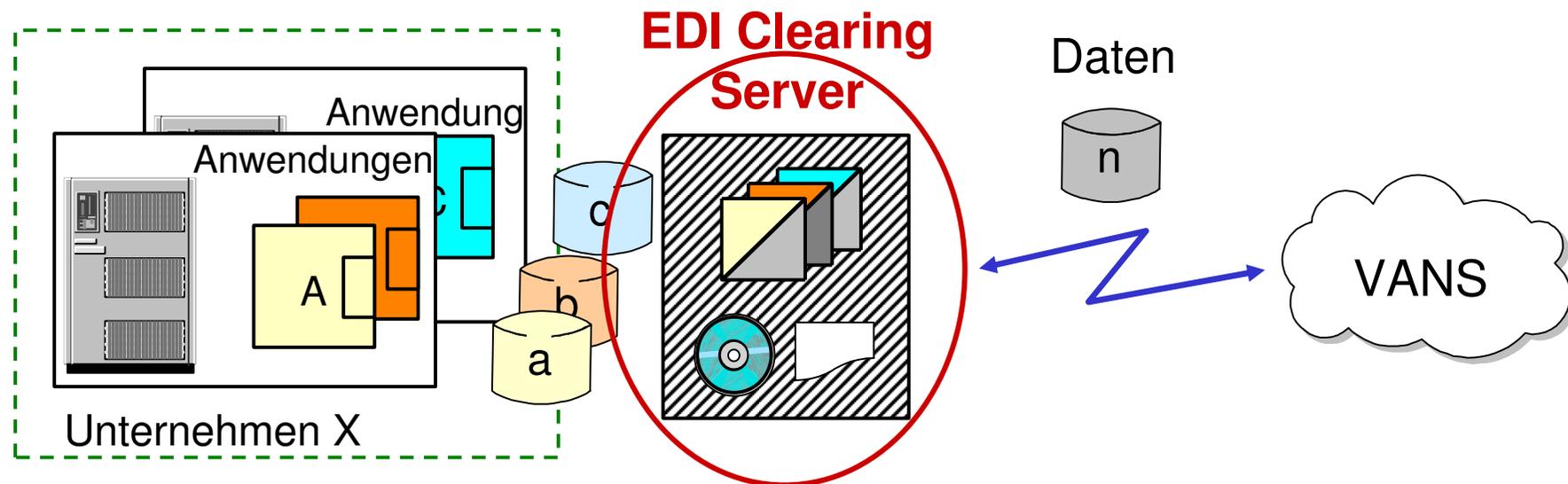


- EDI-Standardaustauschformate
- ✓ Applikationsschnittstellen
- **Mapping**
- Routing
- Messaging / File Transfer
- Extras
  - Archivierung
  - Reporting
  - Alarmierung
  - Tracking & Tracing



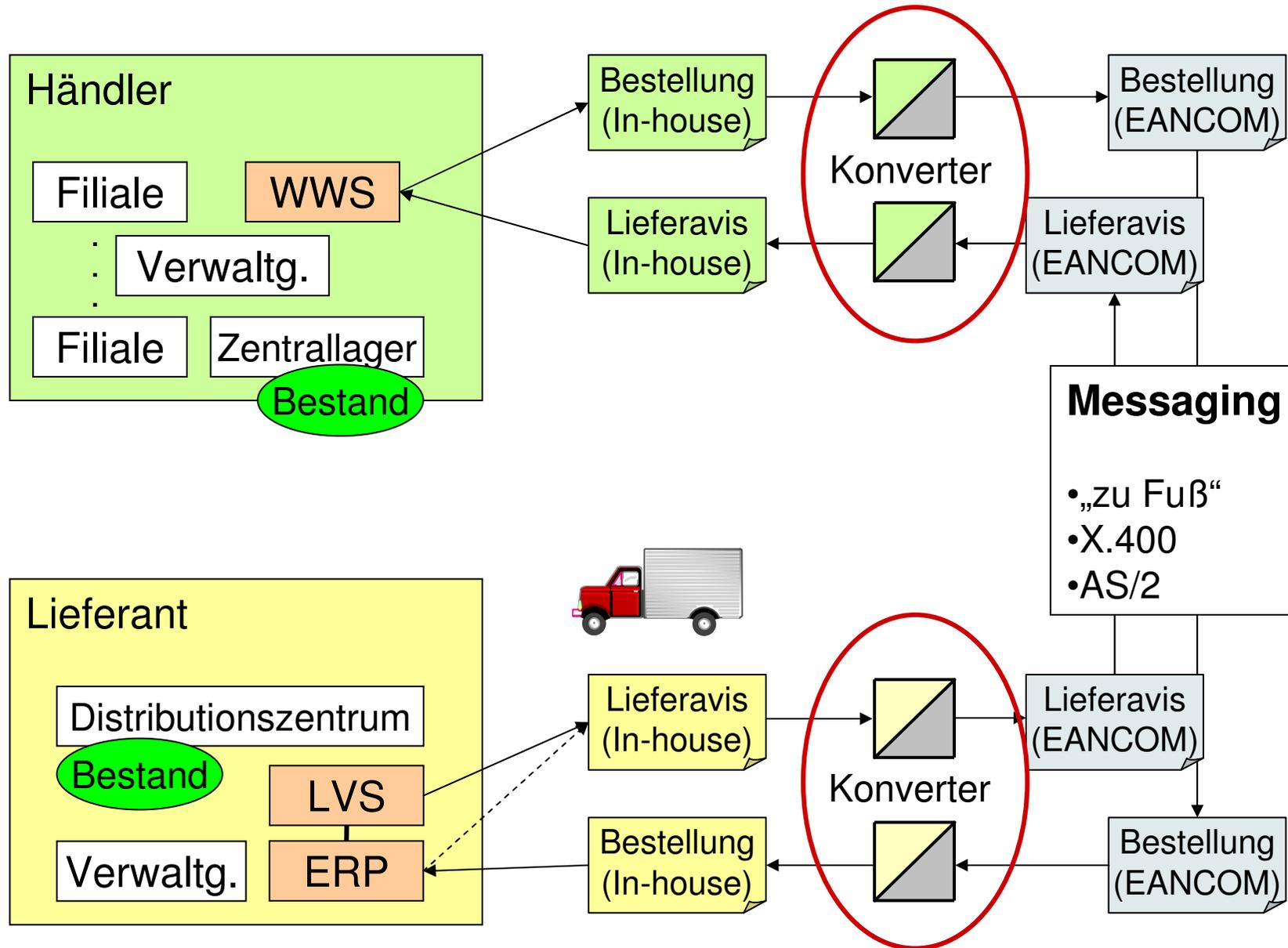
Applikations-  
schnittstellen  
und -formate

## Mapping





# Relevanz für das Praktikumsprojekt





# ***Mapping* und *Mapping-* **Techniken****

Das Grundproblem

Das *Metamap*-Konzept

Andere Ansätze

Spezielle *Mapping*-Aufgaben

Werkzeuge und methodische Ansätze



# Zum Begriff „*Mapping*“

---



- engl. „*map*“ = „Abbildung“ (math.)
- Herstellung / Beschreibung des Zusammenhangs zwischen eingehenden und ausgehenden Daten eines EDI-Konverters, i.d.R. auf Belegebene. Beispiel:
  - Eine SAP IDoc-Struktur „ZINVOIC1“ umwandeln in eine EANCOM-Nachricht „INVOIC“
- Der Begriff suggeriert eine einfache 1:1-Zuordnung zwischen Feldinhalten auf Quell- und Zielstrukturseite, beschreibbar durch eine Zuordnungsvorschrift in Tabellenform
- Tatsächlich können derartige I/O-Beziehungen recht komplex werden! Sie sind dann besser algorithmisch als tabellarisch beschreibbar.



# Mapping: Das Grundproblem

---



- Wunsch und Wirklichkeit
  - Die Anwendung möchte oder kann nur Daten in immer gleicher Weise bereitstellen bzw. akzeptieren.
  - Geschäftspartner benutzen zwar EDI-Standards, aber diese Standards lassen noch eine Fülle von partnerspezifischen Variationen zu.
- Beispiel für Variationen trotz Standardisierung
  - Die Tabelle der Partnerspezifika (234 DE x 11 Firmen!) aus den Anwendungsrichtlinien des AK Handel zu Rechnungsdaten
  - **On-line Demo:** [http://www.edi-ak-handel.de/nach\\_f.htm](http://www.edi-ak-handel.de/nach_f.htm), „Unterstützte Nachrichten je Firma“, „INVOIC“, „Anhang“
  - Bem.: Dies ist bereits „harmonisiert“, d.h. gleiche Anforderungen werden in UN/EDIFACT auch gleich codiert/erwartet. Früher gab es auch noch Abweichungen in der Verwendung vieler Qualifier!
- *Mapping*-Techniken
  - Je nach Vielfalt der Anforderungen ist die passende *Mapping*-Technik (und damit u.U. das dazu passendste Konverterprodukt) zu wählen.



## Algorithmischer Grenzfall: *Mapping* = Software-Entwicklung

- Gemeinsamer, komplexer Programmcode
  - für viele Partnerspezifika
  - mit vielen Fallunterscheidungen / Logikabschnitten
  - redundanzarm, leicht zu warten bei gemeinsamen Veränderungen
- Softwareschalter und optionale Parameter
  - Steuerung im Einzelfall über Schalter, ohne Eingriff in den Code
  - Zusammenfassung zu **Schalter-Sets**
  - Organisation z.B. als *Lookup*-Tabelle:
    - Eine Zeile pro Schaltername, eine Spalte pro Set
- Auswahl des zuständigen Schalter-Sets mittels Absender/Empfängererkennung
  - z.B. aus UNB-Segment oder EDI\_DC

[Quelle: Auch Mapping ist Software-Entwicklung. H.Werntges, edi-change 02/2001, S. 27 - 30]



- Redundanzvermeidung der 2. Ordnung:
  - Funktionale Schaltergruppen.
  - Beispiel Netto-vs. Bruttoabrechnungsverfahren
    - Eine Gruppe repräsentiert „Netto“, eine andere „Brutto“
  - Schalter-Sets
    - Komplette Schaltersammlung
    - Gruppierung: Jeweils ein Set für alle Partnerkennungen mit gleichen Anforderungen
  - Vererbungskonzept:
    - Sets erben Werte von Schaltergruppen und anderen Sets
    - Geerbte Werte können bei Bedarf lokal überschrieben werden
    - Beispiele:
      - Set X wie Set A, aber mit Nettoabrechnungs-Gruppe
      - Set B wie Set A, zusätzlich Schalter S := „M“



- Redundanzvermeidung der 2. Ordnung (Forts.):
  - Ergebnis: Minimierung der Redundanzen der Schaltermatrix
  - Voraussetzungen dazu
    - Vererbungstechnik
    - Default-Sets
    - Erst Fachwissen über die Bedeutung der Schalter gestattet sinnvolle Definition von Schaltergruppen
    - Analyse der Schalterbeziehungen und insbesondere die Definition des Vererbungsgraphen erfordert Erfahrung, Branchenkenntnis und Vorausschau
- **NAD-Mapping** - ein separater Einsatz für Metamaps
  - Partnerfunktionen (z.B. „Besteller“, „Rechnungsempfänger“) aus Sicht des EDI-Partners unterscheiden sich oft von der eigenen Repräsentation in den Stammdaten, also:
  - Partnerspezifische Anpassungen notwendig



# Mapping: Andere Ansätze

---



- **Zweistufiges *Mapping***
  - (1) Quelle → interne Struktur, (2) interne Struktur → Zielformat
  - Variante: Semantische Geschäftsprozess-Integration
    - interne Struktur erzwingt semantisch korrekte Bereitstellung
    - zweiter Teil des *Mapping* ist dann frei von Missverständnissen
- **Tabellarischer Grenzfall: *Mapping* - wörtlich genommen**
  - Viele, aber einfache tabellarische *Maps*
  - Vorteile:
    - Keine Programmierlogik, GUIs einsetzbar,
    - *Mapping* nach Spezifikation durch angelernte Kräfte möglich
  - Nachteile:
    - Schlecht wartbar, hohe Redundanz zwischen verschiedenen *Maps*
    - Aufwändig bei systematischen Änderungen



# Spezielle *Mapping*-Aufgaben

---



- **Plausi-Checks**
  - wenn möglich auch automatische Korrekturverfahren
  - Bsp: Materialstamm; Rechnungslisten; GTIN-Prüfzifferkontrolle
- **Nachrüsten** (konstanter) Stammdaten
  - Bsp: Metro Lieferantenummer
- **Ausnahmetabellen**
  - Bsp: Spezielle Materialien
- **Logs** für Warnungen
  - Bsp. GTIN-Lücken, GLN-Lücken
- Erzeugen von **Meta-Belegen**
  - Bsp. Sammel-Rechnungsliste für REWE
- Während der Entwicklung:
  - ***Unit tests, Test-Suite*** !



- Angepasste Fehlerbehandlung
  - *en bloc*
    - alle Belege ablehnen bei einem Fehler
    - notwendig bei Abhängigkeiten zwischen Belegen,
      - z.B. bei einem Bündel Rechnungen mit Rechnungsliste
  - *split*
    - Nur fehlerhafte Belege abtrennen,
    - Rest konvertieren & versenden
    - Verschafft Zeit zur Fehlerkorrektur
    - Hält nur die Problemfälle zurück



# Grundsatzfrage beim *Mapping*

---



Welche Logik / Funktion gehört

- a) in den Konverter,
- b) in die Anwendung?

## Allgemeine Empfehlung:

Anwendung:

- Gemeinsame Funktionalitäten
- Generelle Logik
- Für Revisionen relevante Dinge
  - Bsp: Preisberechnung, Steuerbetragsermittlung

Konverter:

- Partnerspezifika
- Informationen, die auf den EDI-Standard begrenzt sind



- Grenzfall 1: Mapping wörtlich genommen
  - Felder möglichst 1:1 abbilden, Programmierlogik vermeiden
  - Graphische Mapping-Werkzeuge evtl. gut einsetzbar
  
- Grenzfall 2: Große Vielfalt, komplexe Regeln, algorithmisch beschreibbar
  - Klassische Software-Entwicklung mit geeigneter *Mapping*-Sprache
  
- Technische Umsetzung in Konverterprodukten
  - Interpreter-Ansatz (typisch) vs. kompilierter Code
  - Herstellerabhängig
  - Allgemein: *Tradeoff* suchen zwischen
    - einfach zu erlernenden und bedienenden Werkzeugen, sowie
    - flexiblen, günstig zu wartenden, performanten und skalierbaren Umgebungen
  - Beachte: EDI-Komponenten sind oft langlebig und wartungsintensiv



- Bekanntes (erschwingliches) kommerzielles Werkzeug:
  - Altova MapForce
- On-line Demo
  - Screenshot
    - [http://www.altova.com/de/produkte/mapforce/daten\\_integration.html](http://www.altova.com/de/produkte/mapforce/daten_integration.html)
    - Erkennbar: Graphisches Mapping, tabellarisch + Zusatzfunktionen
  - Video (Flash)
    - <http://www.altova.com/videos.asp?type=0?video=mapforce>
    - Hinweis auf Code-Generatoren
- Diskussion
  - Vor- und Nachteile dieses Ansatzes