



7437 - EDI und E-Business Standards

Praktikumsaufgabe 08

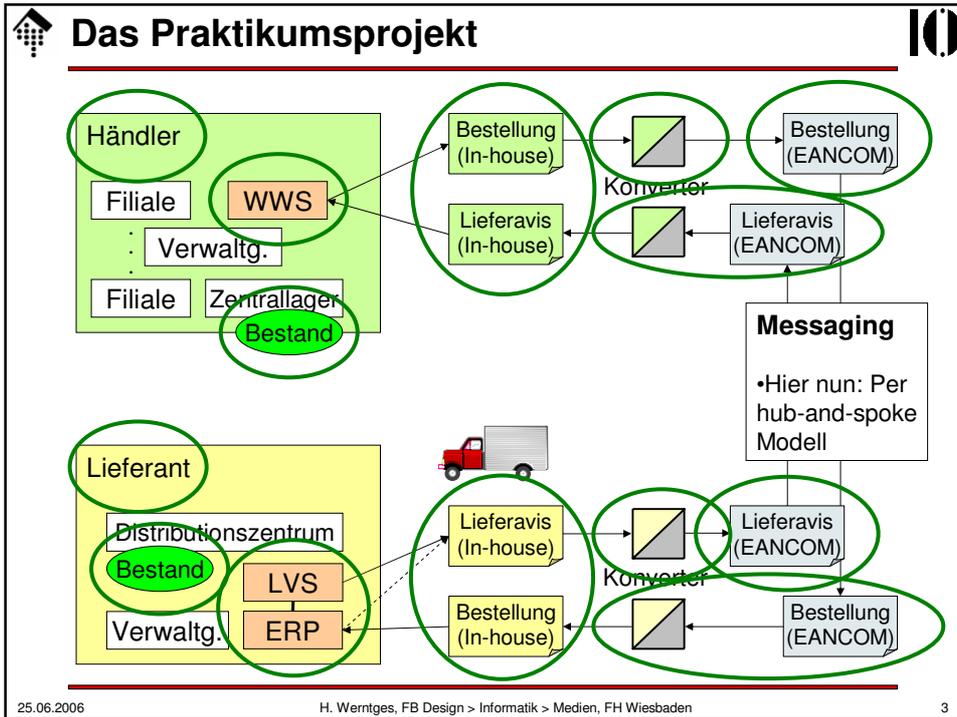
Integration der Abläufe



Das Szenario



- Szenario
 - Die Praktikumssteilnehmer setzen ihre Handelspartner-Rolle (Lieferanten, Händlern) der Konsumgüterbranche aus Praktikum 01/02 fort.
 - Nachdem nun alle Komponenten für den elektronischen Handel bereit stehen, werden diese nun verbunden und automatisiert.
- Ziele der Übung
 - *Messaging*: Simulation per *mailboxing* und *hub-and-spoke* Modell
 - Automatisierungstechniken:
 - Hintergrundprozess, *Logging*, Fehlerbenachrichtigung per *E-Mail*
 - Vorteile und Grenzen der Standardisierung
 - Einheitlichen Datenaustausch mit mehreren Geschäftspartnern trotz unterschiedlicher *inhouse*-Formate erleben
 - Verbleibende Lücken und Interpretationsspielräume in den EANCOM-Daten entdecken.



Die Aufgabe

- Die Aufgabe
 - Schreiben Sie ein (vereinfachtes) „daemon“-Programm, das Ihre Geschäftsabläufe automatisiert
 - Es läuft permanent als Hintergrundprozess
 - Es prüft einmal pro Minute, ob etwas zu tun ist
 - Ggf. leitet es alle notwendigen Schritte ein.
 - Dabei führt es Protokoll und benachrichtigt bei Abbruch per E-Mail
 - Testen Sie die Abläufe gemeinsam mit Ihren Geschäftspartnern!
 - Erzeugen Sie Daten für Ihre bzw. beantworten Sie Daten von Ihren Geschäftspartnern.
 - Programm-Name: **controld.rb**
 - Gewünschter Aufruf: **controld.rb &**

Als Hintergrundprozess betreiben.
Einfacher Ersatz für echte Daemons
bzw. für crontab

25.06.2006 H. Werntges, FB Design > Informatik > Medien, FH Wiesbaden 4



Händler



- Händlersicht
 - Rufen Sie **bestellen.rb** aus Übung 04 jede Minute auf
 - Dieses Programm prüft Ihre Bestände und generiert ggf. Bestellungen in Ihrem *inhouse*-Format
 - Falls Bestellung(en) generiert:
 - Rufen Sie **mapper06.rb** für jede erzeugte, nach Geschäftspartnern vorsortierte Datei auf, um sie nach EANCOM zu wandeln.
 - Kopieren oder verschieben Sie die EANCOM-Dateien in das **out**-Verzeichnis Ihrer Mailbox (s.u.).
 - Prüfen Sie jede Minute das **in**-Verzeichnis Ihrer Mailbox
 - Falls neue (EANCOM-) Daten eingetroffen:
 - Rufen Sie **mapper07.rb** auf und wandeln Sie diese Daten in Ihr *inhouse*-Format für Lieferavise um, löschen Sie sie bei Erfolg.
 - Rufen Sie schließlich **ware_annehmen.rb** aus Übung 04 für die *inhouse*-Daten auf, um den Bestellprozess abzuschließen.



Lieferanten



- Lieferantensicht
 - Prüfen Sie jede Minute das **in**-Verzeichnis Ihrer Mailbox
 - Falls neue (EANCOM-) Daten eingetroffen sind:
 - Rufen Sie **mapper07.rb** auf und wandeln Sie diese Daten in Ihr *inhouse*-Format für Bestellungen um.
 - Entfernen Sie bei Erfolg die verarbeiteten Daten aus der „*inbox*“.
 - Rufen Sie nun für jede *inhouse*-Datei **liefern.rb** aus Übung 04 auf, und erzeugen Sie so entsprechende Lieferavise in Ihrem *inhouse*-Format.
 - Rufen Sie **mapper06.rb** für jede erzeugte (nach Geschäftspartnern vorsortierte) Datei auf, um diese Daten nach EANCOM zu wandeln.
 - Kopieren oder verschieben Sie die EANCOM-Dateien in das **out**-Verzeichnis Ihrer Mailbox (s.u.).



Technische Hinweise



- Logging:

```
#!/usr/bin/env ruby
require 'logger'

# Log levels: DEBUG < INFO < WARN < FATAL < UNKNOWN
log = Logger.new("mein_edi_log.txt")
log.level = Logger::WARN # Optionale Filterung: Log ab WARN
log.info("Programmstart") # Unsichtbar
log.warn("Eine Warnung") # Sichtbar!
```
- Benachrichtigung im Fehlerfall (Unix/Linux)

```
begin
# ... Ihr Code
rescue => e # Alles abfangen
  IO.popen("mail -s 'controld-Abbruch!' werntges", "w") do |io|
    io.puts e, e.backtrace.join("\n")
  end
end
```

Mail subject

Eigne Empfängeradresse
(bzw. user account name)
verwenden!



Technische Hinweise



- Aufruf externer Programme (Beispiele):
 - Mit „Backquotes“:

```
srcname = 'meineQuelldatei.inh'
`ruby mapper06.rb #{srcname} > meinErgebnis.edi`

# Rückgabewert = stdout des aufgerufenen Kommandos,
# Fehlercode (process status) in $?
```
 - Mit Kommando „system“:

```
src = 'meineQuelldatei.inh'
system("ruby mapper06.rb #{src} > meinErgebnis.edi")

# Fehlercode (process status) in $?
# SystemCallError bei Problemen.
```



Technische Hinweise



- Was protokollieren?:
 - Jeden Start von „controld.rb“ Level: **info**
 - Jedes reguläre Ende von „controld.rb“ Level: **info**
 - Jeden Abbruch von „controld.rb“ Level: **fatal**
 - Gleichzeitig Benachrichtigung per E-Mail!
 - Eingeleitete Aktionen Level: **info**
 - Z.B. Namen erzeugter oder angetroffener Dateien
 - Aufrufe von mapper06 & mapper07, von „liefern“ und „ware_annehmen“
 - Fehlercodes in Aufrufen Ihrer Programme, Level: **error**
 - mapper06, mapper07 und alle „Anwendungen“ aus Übung 04



Hub-and-spoke Modell, Mailboxing



- Hintergrund
 - Zum „firmenübergreifenden“ Datenaustausch verwenden wir ein vereinfachtes *Hub-and-Spoke*-Modell und das *mailboxing*-Prinzip
 - Jede „Firma“ besitzt eine Mailbox (d.h. eine Verzeichnisstruktur) auf dem „*hub*“. Der Name der Mailbox ist die Basis-GLN der Firma.
 - Nur UN/EDIFACT-Daten werden ausgetauscht. Ihre Sender- und Empfänger-GLNs im UNB-Segment bilden die Grundlage für die korrekte Zustellung (*mailboxing*).

- Verzeichnisstruktur auf dem *hub*

```
~werntges/lv/edi/hub # Basisverzeichnis des Hubs
hub/<gln> # Basisverzeichnis der Mailbox
           zur Firma mit Basis-GLN <gln>
<gln>/in # Enthält eingehende Dateien
<gln>/out # Zu versendende Dateien hierhin!
<gln>/err # Abgelehnte (z.B. nicht validierende
           oder nicht zustellbare Dateien)
```



Hub-and-spoke Modell, Mailboxing



- Dateinamenskongvention
 - Extension: Stets „.edi“
 - Eingehende Dateien leicht identifizierbar: Suche nach *.edi genügt.
 - Dateien müssen vom Sender eindeutig benannt werden
 - Dateien unterschiedlicher Geschäftspartner müssen kollisionsfrei sein
 - Einfach erreichbar mittels eigener GLN als Präfix
 - Vorschlag für Namenskonvention:
 - <sender-gln>_<empfaenger_gln>_<seriennummer>.edi
 - Für die Seriennummer können Sie den Wert des Interchange Control Reference-Elements aus UNB oder eine eigens vergebene Nummer verwenden, deren Eindeutigkeit Sie mit „numgen“ sicherstellen.
 - Beispiel:
`2965197000005_2165197000009_0001.edi`



Hub-and-spoke Modell, Mailboxing



- Verzeichnisse mit Ruby nach Dateien durchsuchen
 - Beispiel: Suche *.edi in inbox zu GLN 2165197000009

```
mailbox_dir = "/local/werntges/lv/edi/hub/2965197000009"
Dir.foreach("#{mailbox_dir}/in") do |fname|
  next unless fname =~ /\.*\\.edi$/ && File.file?(fname)
  # Verarbeitung der ,Treffer` (angedeutet)
  `ruby mapper07.rb #{fname} >> result.inh`
  File.delete(fname) if $? == 0
end
```



Abgaben



- **Abzugeben**
 - `controld.rb` # Ihr Programm-Code
- **Abgabeordner**
 - Wie üblich, unter `~werntges/lv/edi/abgaben/a/<matnr>`
- **Annahmeschluss**
 - Dies ist eine Doppel-Übung (Aufgabe für 2 Termine).
 - Sie stellt den Abschluss des Praktikums-Projekts dar.
 - Abgabeschluss daher: Freitag nach dem letzten Praktikum (30.6.06).
- **Anmerkungen**
 - Der mittels `controld.rb` erzeugte Gesamtablauf ist zentraler Gegenstand der Projektabnahme!
 - Die Abnahme sollte im Idealfall zusammen mit Ihren Geschäftspartnern geschehen. Im Bedarfsfall stellt der Kursleiter einen Geschäftspartner.