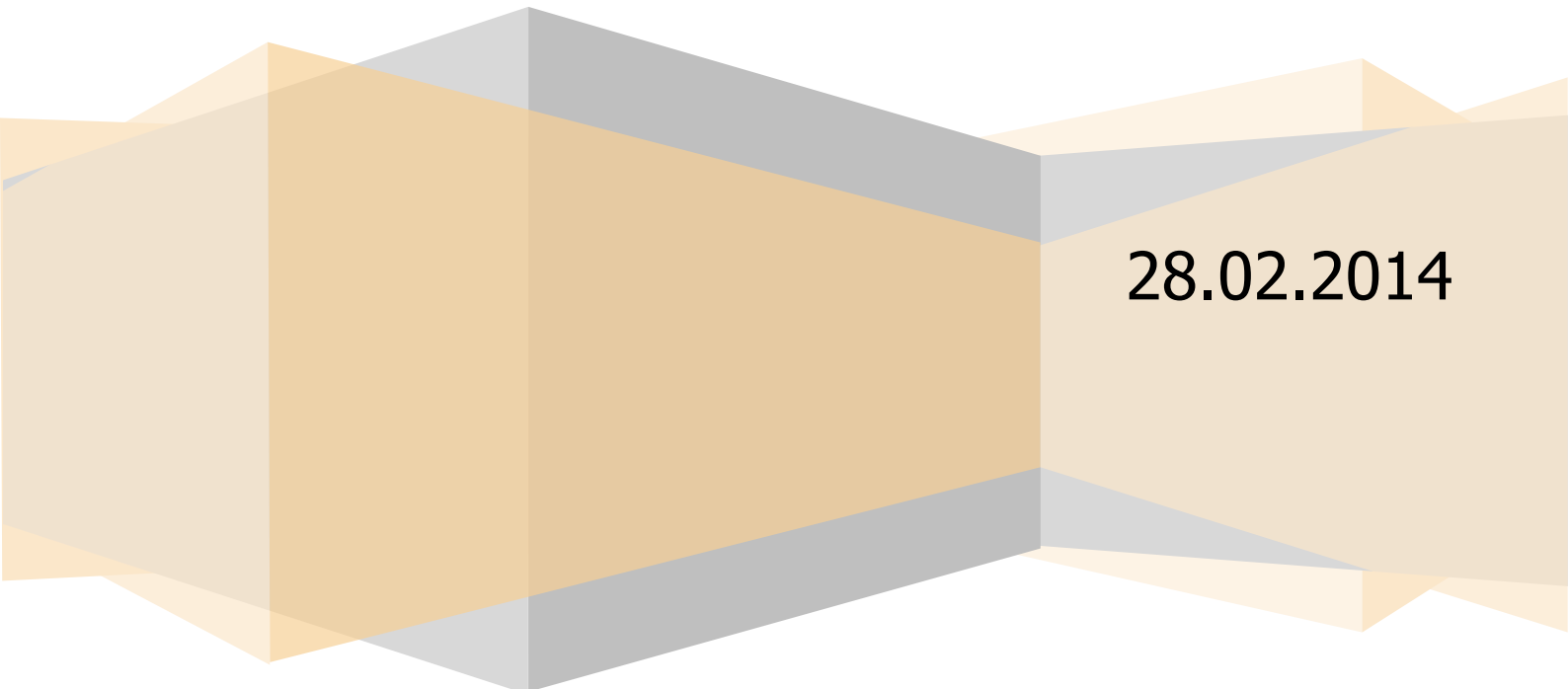


4AHIT

ETL

Protokoll

Belinic Vennesa & Scholz Dominik



28.02.2014

Inhaltsangabe

Git-Link	1
Aufgabenstellung	1
Resources	1
Arbeitsaufteilung	2
Aufwandschätzung	3
Endzeitaufteilung.....	3
Arbeitsdurchführung und Testbericht	4
Technologiebeschreibung.....	7
Quellenangaben	8

Git-Link

<https://github.com/vbelinic-tgm/ETL>

(Dieses Repository ist erst nach dem Schließen der Abgabe sichtbar.)

Aufgabenstellung

Enterprise Application Integration

Gruppenaufgabe (2 Leute)

"The ETL (Extract, Transform, Load) is a mechanism for loading data into systems or databases using some kind of Data Format from a variety of sources; often files then using Pipes and Filters, Message Translator and possible other Enterprise Integration Patterns. So you could query data from various Camel Components such as File, HTTP or JPA, perform multiple patterns such as Splitter or Message Translator then send the messages to some other Component. To show how this all fits together, try the ETL Example." [1]

ETL ist ein wichtiger Prozess bei einem Datawarehouse. Zeigen Sie wie Enterprise Integration Patterns [2] dabei eingesetzt werden können (8 Punkte, nur jene, die in dem Beispiel vorkommen). Verwenden Sie dazu das ETL Example [3]. Dokumentieren Sie die Implementierung sowie alle notwendigen Schritte ausführlich in einem Protokoll (8 Punkte). Fügen Sie den verwendeten Code nach den Metaregeln an und geben Sie alles als ZIP-Archiv (Gesamtes Framework mit Anleitung, wie das System gestartet werden kann) ab.

Resources

- [0.1] Extract Transform Load (ETL); Apache Camel; Online:
<http://camel.apache.org/etl.html>; abgerufen 13.02.2014
- [0.2] Enterprise Integration Patterns; G.Hohpe, B.Woolf; 2003; Online:
<http://www.enterpriseintegrationpatterns.com/toc.html>; abgerufen 13.02.2014
- [0.3] Extract Transform Load (ETL) Example; Apache Camel; Online:
<http://camel.apache.org/etl-example.html>; abgerufen 13.02.2014



Arbeitsaufteilung

Die Arbeit wird so aufgeteilt, dass jedes Gruppenmitglied, soweit wie möglich, von einander unabhängig arbeiten kann. Deshalb werden den jeweiligen Gruppenmitglieder ganze Arbeitspakete zugeteilt.

Vennesa

- Durchführung und Dokumentation der Installation und aller notwendigen Schritte
- Technologiebeschreibung

Dominik

- Ausarbeiten der verwendeten Patterns

	<i>geschätzte Zeit</i>	<i>tatsächliche Zeit</i>
Belinic Vennesa	2 h	1 h 15 min
Scholz Dominik	2 h	1 h

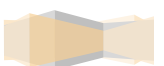


Aufwandschätzung

Arbeitspaket	Aufwand für Arbeitspaket
Ausarbeiten der verwendeten Patterns	2 h
Technologiebeschreibung	1 h
Durchführung und Dokumentation der Installation und aller notwendigen Schritte	1 h
Gesamtsumme	4 h

Endzeitaufteilung

Arbeitspaket	Aufwand für Arbeitspaket
Ausarbeiten der verwendeten Patterns	1 h
Technologiebeschreibung	45 min
Durchführung und Dokumentation der Installation und aller notwendigen Schritte	30 min
Gesamtsumme	2 h 15 min



Arbeitsdurchführung und Testbericht

Ausführung des Beispiels

Als erstes ist es notwendig sich das JAR-File herunterzuladen, am besten von Apache.

```
schueler@debian:~$ wget http://tweedo.com/mirror/apache/camel/apache-camel/2.12.3/apache-camel-2.12.3.tar.gz
```

Danach habe ich das JAR-File entpackt:

```
schueler@debian:~$ tar xfv apache-camel-2.12.3.tar.gz
```

Danach verschiebe ich es in einen anderen Ordner:

```
schueler@debian:~$ mv apache-camel-2.12.3 opt/
```

Nun bevor ich die Examples ausführen kann ist es notwendig Maven zu installieren, da im Tutorial Maven verwendet wird [1].

```
schueler@debian:~/opt$ sudo apt-get install maven
```

Nun wechsele ich in den Ordner vom Example und führe es mit mvn camel:run aus[1]

```
schueler@debian:~/opt/apache-camel-2.12.3/examples/camel-example-etl$ mvn camel:run
[INFO] Scanning for projects...
```

Auschnitte der Ausgabe:

```
[INFO] Building Camel :: Example :: ETL 2.12.3
[INFO] -----
17376 camel TRACE [Camel (camel) thread #1 - jpa://org.apache.camel.example.etl.CustomerEntity] openjpa.jdbc.SQL - <t 693683179, conn 1704664989> executing
prepstmt 1476009298
```

Mit Strg + C kann man das Programm beenden.

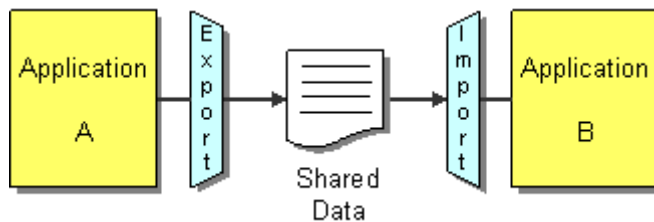


Patterns

Integration Styles: File Transfer

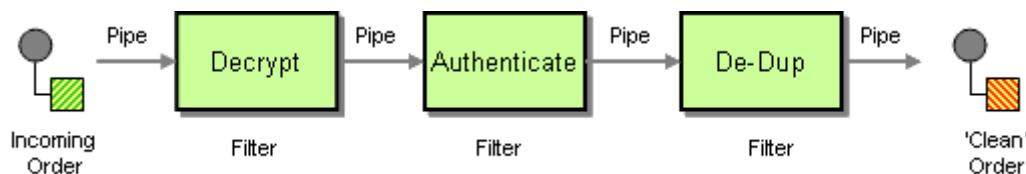
Das Verwendete Integration Style Pattern im Beispiel ist ein File Transfer Pattern. In dem Beispiel werden XML Dateien konsumiert und in POJO und JPA transformiert.

"Jedes Programm erstellt Dateien, welche die Informationen für die anderen Programme enthalten. "Integrators" übernehmen die Verantwortung des Transformierens der Dateien in unterschiedliche Formate. Die Dateien werden in konstanten Intervallen produziert."



Messaging System: Pipes and Filters

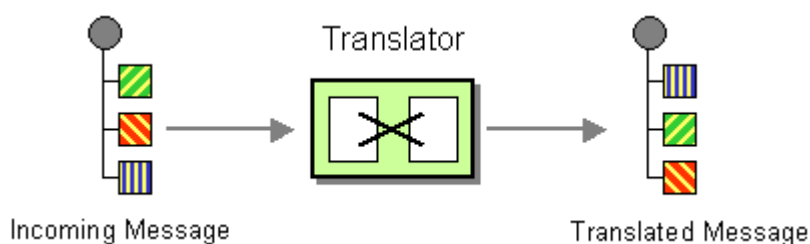
"Mit Apache Camel kann man das verarbeiten auf verschiedene Endpoint Instanzen aufteilen, welche dann verkettet werden können"



Messaging System: Message Translator

In dem Beispiel wird die CustomerTransformer Klasse als Message Translator eingesetzt.

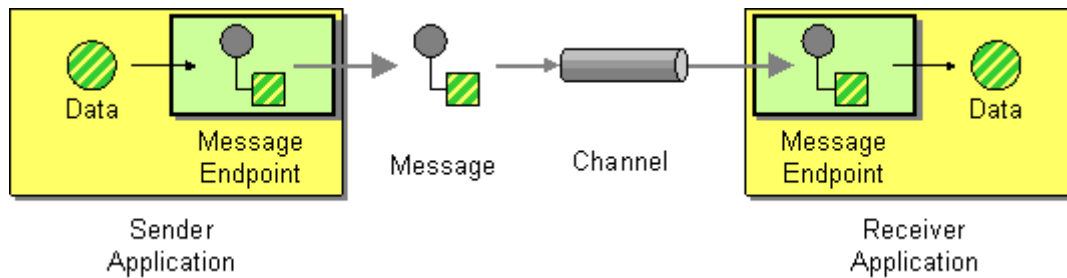
"Benutzt einen speziellen Filter, den Message Translator, um zwischen anderen Filtern oder Anwendungen zu translaten."



Messaging System: Message Endpoint

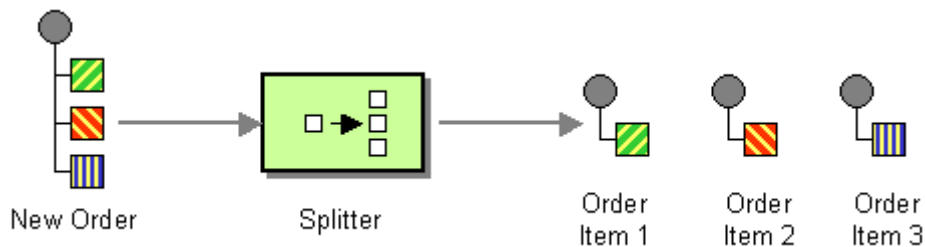
"Verbindet eine Anwendung mit einem Messaging Channel (Kanal) unter der Benutzung eines Message Endpoints. Der Message Endpoint ist ein client des Messagings Systems welches von der Anwendung zum sendn und empfangen von Nachrichten verwendet werden kann."

Im Source Code: EtlRoutes.java



Messaging System: Splitter

"Viele Nachrichten gehen durch eine Integrierte Lösung welche aus mehreren Elementen Entsteht. Der Splitter splittet die Nachricht in eine Reihe einzelner Nachrichten, welche jede einzelne Daten für ein Item entahlsen"



Technologiebeschreibung

EAI

EAI bedeutet Enterprise Application Integration und ist eine IT-Infrastruktur um bestehende System miteinander zu verbinden. Denn nicht alle Systeme übertragen die Daten in gleicher Form, daher muss zwischen den Systemen eine Middleware sein die sich um dies kümmert.

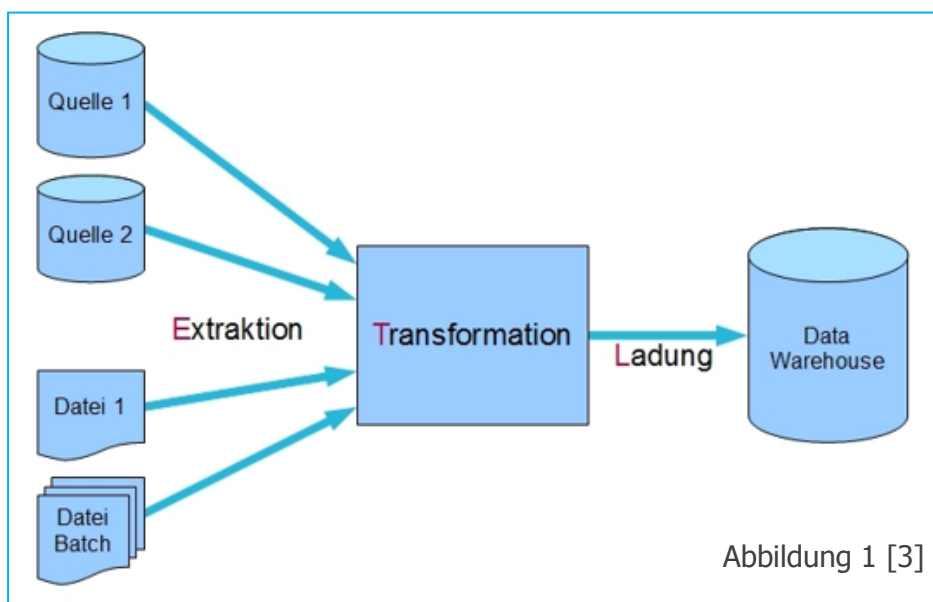
Und weil eine Middleware zwischen den System ist, verringert sich auch die Komplexität, anstatt $N \cdot (N-1) / 2$ möglich Verbindungen hat man nur noch N möglich Verbindungen. [2]

Durch den Einsatz von EIP (Enterprise Integration Patterns), die sich mit der Frage der Softwareintegration beschäftigen, können Probleme bereits im Vorhineine vermieden werden. Denn die meisten EIP bieten die "best practice" in diesem Bereich und bereits fertig funktionierende Beispiele.

ETL

"Der Begriff ETL (Extract Transform Load) bezeichnet die drei klassischen Schritte zu Befüllung eines Data Warehouses:

- 1. Extraktion:** Die Daten werden aus einem oder mehreren Quellsystemen in das ETL-System geladen.
- 2. Transformation:** Die Daten werden im ETL-System umgewandelt. Dabei kann es sich sowohl um technische Umwandlungen wie beispielsweise der Änderung der Datentypen, aber auch um fachliche Umwandlungen wie z.B. von Berechnungen und Aggregationen, handeln.
- 3. Load:** Die Daten werden aus dem ETL-System ins Data Warehouse geladen." [3]



Quellenangaben

- [1] Apache Camel, "Extract Transform Load (ETL) Example", online verfügbar: <http://camel.apache.org/etl-example.html>, zuletzt besucht: 20.02.2014
- [2] Torsten Horn, "Technische Vorteile", zuletzt aktualisiert: 2007, online verfügbar: <http://www.torsten-horn.de/techdocs/eai.htm>, zuletzt besucht: 26.02.2014
- [3] Dipl.-Ing. Klaus Lipinski, "ETL (extract transform load)", online verfügbar: <http://www.itwissen.info/definition/lexikon/ETL-extract-transfer-load.html>, zuletzt besucht: 26.02.2014

