

TGM XX

EXTRACT, TRANSFORM AND LOAD (ETL) WITH EAI

DEZSYS

28.02.2015

Inhalt

1	Aufgabenstellung	2
2	Patterns	2
2.1	File Transfer.....	2
2.2	Message Translator	3
2.3	Content Enricher	3
2.4	Event Message	3
2.5	Selective Consumer	4
3	Apache Camel	4
4	Beispiel	4
5	Docs	5
6	Quellenangaben.....	6

1 Aufgabenstellung

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.

[1] Extract Transform Load (ETL); Apache Camel; Online: <http://camel.apache.org/etl.html>; abgerufen 13.02.2015

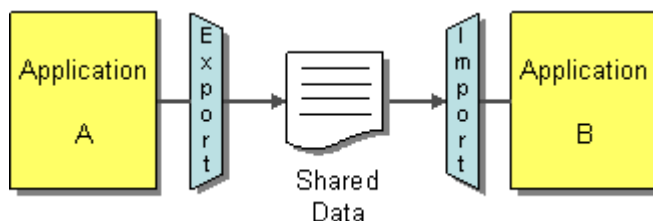
[2] Enterprise Integration Patterns; G.Hohpe, B.Woolf; 2003; Online: <http://www.enterpriseintegrationpatterns.com/toc.html>; abgerufen 13.02.2015

[3] Extract Transform Load (ETL) Example; Apache Camel; Online: <http://camel.apache.org/etl-example.html>; abgerufen 13.02.2015

2 Patterns

Die verwendeten Patterns konnten nicht erkannt werden. Es folgt somit eine Erklärung aller Patterns die im Unterricht in den Präsentationsfolien vorkamen.

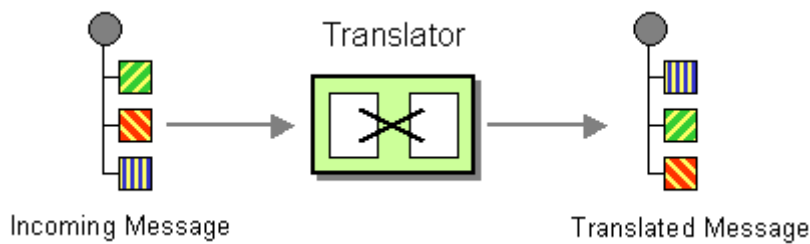
2.1 File Transferⁱ



Der Datenaustausch zwischen Applikationen kann über Dateien geschehen. Die beiden Applikationen müssen jedoch ein Dateiformat/Notation verwenden die beide verstehen und interpretieren können. Daraus haben sich standardisierte Protokolle

entwickelt die zum Fileaustausch zwischen zwei Applikationen verwendet werden können. Eines der bekanntesten und am meisten verwendeten ist XML.

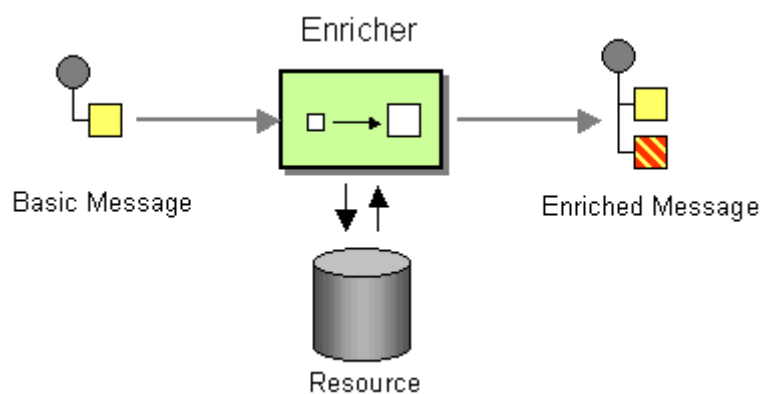
2.2 Message Translatorⁱⁱ



Der Message Translator ist dazu da, um Ausgaben einer Applikation die z.B. ein proprietäres Protokoll verwendet einer anderen Applikation zu übersetzen. Er extrahiert die Daten aus

der Nachricht und wandelt sie so um, dass die andere Applikation diese verstehen kann. Es ist möglich das die Nachricht durch den Translator bloß wieder in ein anderes proprietäres Protokoll umgewandelt wird oder sogar in ein standardisiertes Protokoll.

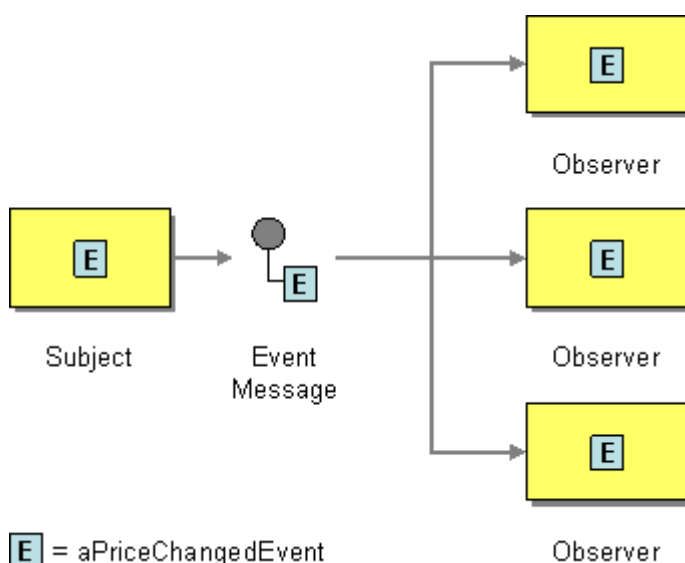
2.3 Content Enricherⁱⁱⁱ



Der Content Enricher reichert Nachrichten mit mehr Informationen an. Da eine Applikation nicht immer den kompletten Datensatz benötigt oder ihn aus Datenschutzgründen nicht erhalten darf, bekommt diese nur die notwendigen Daten. Hat diese Applikation jedoch eine

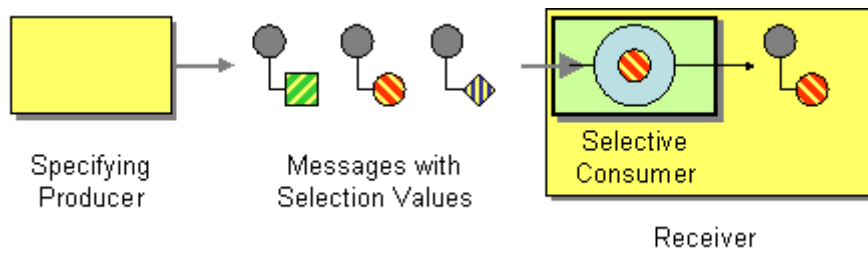
Schnittstelle zu einer anderen Applikation an welche sie Nachrichten versendet so kann es notwendig sein Daten nachzuladen. Die andere Applikation könnte ja mehr Daten benötigen. Der Enricher verarbeitet nun die Nachricht und prüft nach was die zweite Applikation benötigt. Falls Daten fehlen lädt der Enricher sie z.B. aus einer Datenbank nach.

2.4 Event Message^{iv}



Eine Event Message dient dazu andere Applikationen über ein Event zu informieren. Diese ist nötig, da das ständige Pollen von Informationen zu ressourcenaufwändig wäre. Wir kennen dieses Prinzip aus dem ObserverPattern welches bereits in SEW besprochen wurde.

2.5 Selective Consumer^v



Der Selective Consumer erhält Messages von anderen Applikationen. Diese filtert er und verwendet nur die Messages die er

wirklich benötigt. Der Rest wird z.B. weitergeleitet an andere Selective Consumers.

3 Apache Camel^{vi}

Apache Camel ist ein kleines Framework welches alle Enterprise Integration Patterns implementiert. Somit kann man verschiedene Applikationen einfach integrieren durch verwendung der bereits implementierten Patterns. Es kann Java, Spring XML und Scala verwendet werden. Es unterstützt sehr viele Technologien zum Nachrichtenaustausch wie etwa HTTP, FTP, JMS, EJB, RMI und viele mehr. Das Framework kann einfach erweitert werden.

4 Beispiel

Um das Beispiel starten zu können muss Maven1 vorhanden sein. Achtung: Mit Maven2 treten fehler auf. Das richtige Paket ist im APT unter dem Namen „maven“ verfügbar.

Das Beispiel ist wie viele andere in der Camel „installation“ enthalten. Um Camel zu „installieren“ muss dieses nur von der Herstellerseite heruntergeladen^{vii} werden.

Das Paket kann nun mit `unzip` entpackt werden. Die Examples sind nun im Unterordner `apache-camel-x/examples` zu finden. Unser Example ist hier im Ordner `camel-example-etl` zu finden.

Beim ersten Start muss `mvn compile` ausgeführt werden. Bei jedem weiteren Start kann `mvn camel:run` aufgerufen werden.

Die Software liest Customer Entitys aus XML Files aus. Diese sind im Unterordner `src/data` abgelegt. Sobald das XML eingelesen wurde wird überprüft ob dieser Customer bereits als persistentes Objekt in der Datenbank abgelegt wurde.

```
2015-02-28 11:33:04,392 [file://src/data] INFO CustomerTransformer
- Created a new CustomerEntity Customer[userName: hiram firstName:
null surname: null] as no matching persisted entity found.
```

```
2015-02-28 11:33:04,409 [file://src/data] INFO CustomerTransformer
- Created object customer: Customer[userName: hiram firstName: Hiram
surname: Chirino]
```

Kurze Zeit später wird auch der zweite XML Datensatz gefunden und persistiert.

```
2015-02-28 11:33:04,531 [file://src/data] INFO CustomerTransformer
- Created a new CustomerEntity Customer[userName: james firstName:
null surname: null] as no matching persisted entity found.
```

```
2015-02-28 11:33:04,531 [file://src/data] INFO CustomerTransformer
- Created object customer: Customer[userName: james firstName: James
surname: Strachan]
```

Jetzt sind alle Objekte persistiert. Beim genaueren betrachten der Objekte fällt etwas auf. Die Klasse, welche die XML Files einliest hat folgende Attribute:

```
@XmlAttribute
private String user;
@XmlElement
private String firstName;
@XmlElement
private String lastName;
@XmlElement
private String city;
```

Die CustomerEntity enthält jedoch mehr Attribute:

```
@XmlAttribute
private Long id;
private String userName;
private String firstName;
private String surname;
private String street;
private String city;
private String zip;
private String phone;
```

Es kann also angenommen werden das die Datensätze angereichert werden sollen. Es wurde jedoch keinen hinweis auf wirklich erweiterte Datensätze gefunden. Es wird angenommen das die Daten hierfür fehlen.

5 Docs

Beim generieren der Javadocs traten mit Eclipse fehler auf. Eclipse erkannte die JEE Annotations nicht und produzierte somit keine Doc Files. Die Docs wurden dann mit IntelliJ erstellt, welches jedoch auch Fehler ausgab, die Docs jedoch erstellte.

6 Quellenangaben

- ⁱ Gregor Hohpe and Bobby Woolf, File Transfer,
<http://www.enterpriseintegrationpatterns.com/FileTransferIntegration.html>, zuletzt besucht 28.02.2015
- ⁱⁱ Gregor Hohpe and Bobby Woolf, Message Translator,
<http://www.enterpriseintegrationpatterns.com/MessageTranslator.html>, zuletzt besucht 28.02.2015
- ⁱⁱⁱ Gregor Hohpe and Bobby Woolf, Content Enricher,
<http://www.enterpriseintegrationpatterns.com/DataEnricher.html>, zuletzt besucht 28.02.2015
- ^{iv} Gregor Hohpe and Bobby Woolf, Event Message,
<http://www.enterpriseintegrationpatterns.com/EventMessage.html>, zuletzt besucht 28.02.2015
- ^v Gregor Hohpe and Bobby Woolf, Selective Consumer,
<http://www.enterpriseintegrationpatterns.com/MessageSelector.html>, zuletzt besucht 28.02.2015
- ^{vi} Kai Wähner, When to use Apache Camel, <http://www.kai-waehner.de/blog/2011/06/02/when-to-use-apache-camel/>, zuletzt besucht 28.02.2015
- ^{vii} Apache Camel, Downloads, <http://camel.apache.org/download.html>, zuletzt besucht 28.02.2015