

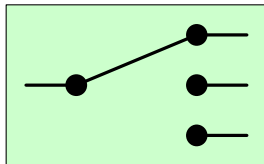
Enterprise Application Integration Patterns Message Routing

Seminar Software Design Patterns

Sommersemester 09

Vortragender: Johannes Schmidt

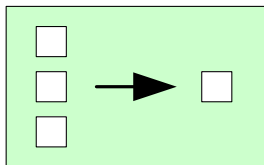
Betreuer: Martin Gebauer



➤ **Einführung**

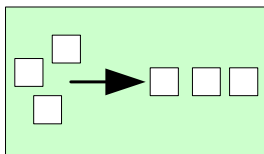
➤ Kontext

➤ Messaging



➤ Content Based Router

➤ Resequencer



➤ Aggregator

➤ Literaturverzeichnis

Enterprise Application Integration Patterns

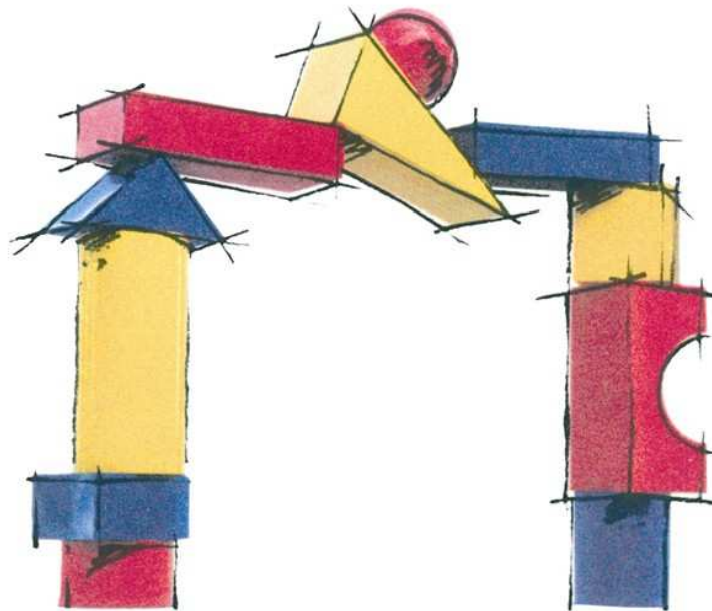


Bild: <http://www.teamsoft-sportzeit.de>

Enterprise Application Integration Patterns

- Firmen beliebiger Größe
- Komplexe interne Strukturen und Abläufe
- Komplexe IT-Landschaft

Enterprise **Application** Integration Patterns

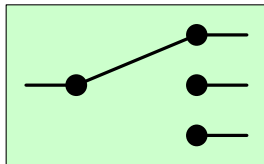
- Software mit besonderen Ansprüchen
 - Zuverlässigkeit
 - Sicherheit
 - Performanz
- Spezialsoftware (CRM, ERP, CMS)
- Unterstützung der Geschäftsprozesse
- Altanwendungen
- Unterschiedliche Hersteller
- Komplexe Daten, hohes Datenaufkommen
- Persistente Daten
- Unterschiedlichste Repräsentation der Daten

Enterprise Application **Integration** Patterns

- Funktionen unterschiedlicher Software bilden Geschäftsprozess ab
- Integration von heterogenen Systemen
 - Firmenfusion
 - Firmenausgliederung
- Oft sehr aufwendig und teuer
- Migration und Konvertierung von Daten
- Anpassung von Geschäftsprozessen
 - schnell
 - flexibel
- Business2Business

Enterprise Application Integration **Patterns**

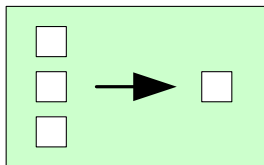
- “*Each pattern describes a **problem** which occurs over and over again in our environment, and then describes the **core of the solution** to that problem, in such a way that you can use this solution a million times over, **without ever doing it the same way twice**” [AIS77]*
- Patterns are for reuse
- Praxiserprobt
- Unterschied zwischen Design Patterns und Integration Patterns
- Graphische Notation nach [HW03]



✓ Einführung

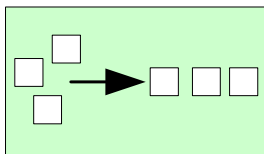
➤ **Kontext**

➤ Messaging



➤ Content Based Router

➤ Resequencer

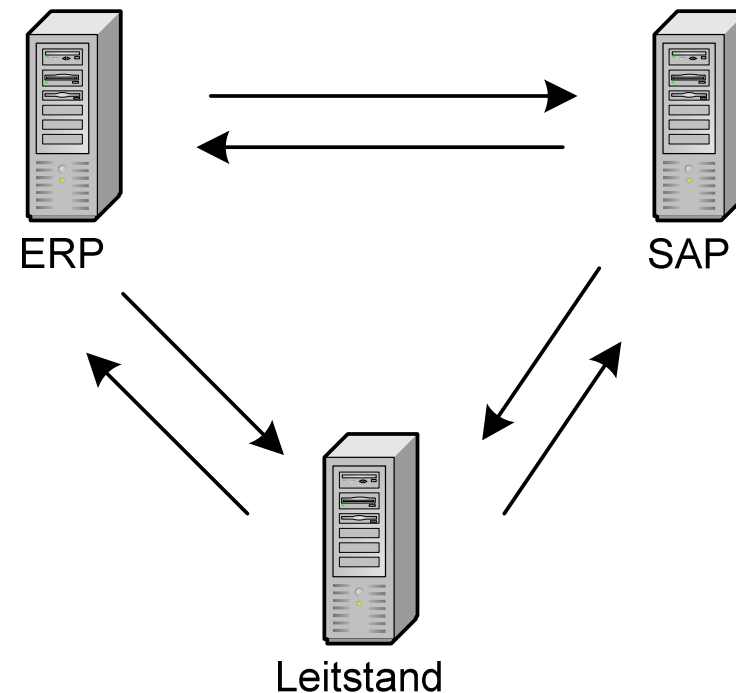


➤ Aggregator

➤ Literaturverzeichnis

Kontext (1)**Integrationsbeispiel**

- Firma in produzierendem Gewerbe
- Kundendaten im SAP
- Grobplanung im ERP
- Feinplanung im Leitstand

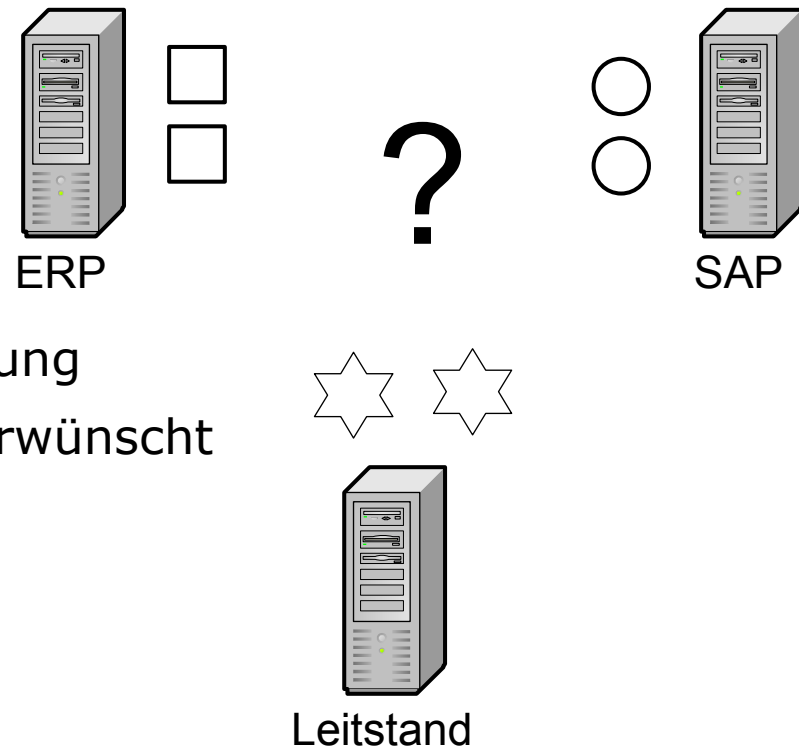
**Warum Integration?**

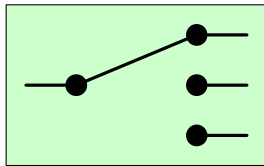
- Kunde soll über fertigen Auftrag informiert werden
- Feinplanung benötigt Daten der Grobplanung

Kontext (2)

Integration ist nicht trivial

- Schlimmstenfalls $O(n^2)$ Verbindung
- Enge Kopplung der System unerwünscht
- Unterschiedliche Technologien
- Unterschiedliche Datenformate
- Hohe Gesamtkomplexität

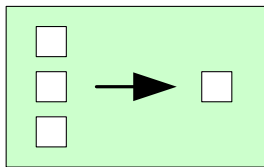




✓ Einführung

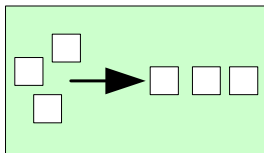
✓ Kontext

➤ **Messaging**



➤ Contend Based Router

➤ Resequencer

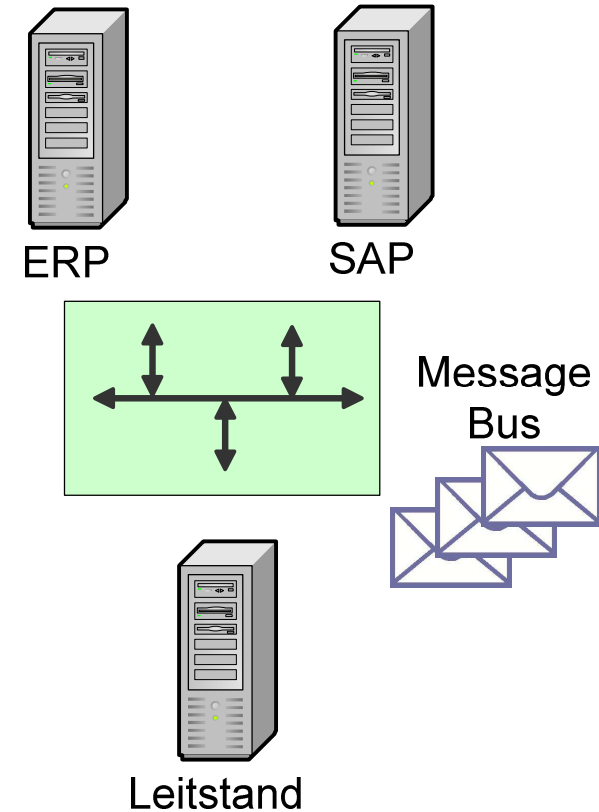


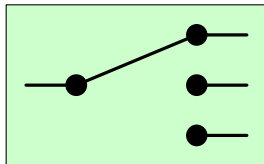
➤ Aggregator

➤ Literaturverzeichnis

Messaging

- Messaging
 - Asynchron
 - Message Channels
 - Message Bus
 - Adapter
- Message
 - Header (Verwaltung)
 - Body (Daten)
 - Unterschiedliche Nachrichtentypen

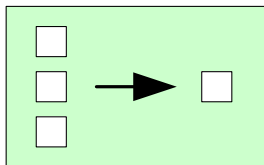




✓ Einführung

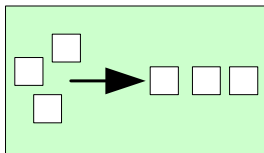
✓ Kontext

✓ Messaging



➤ **Content Based Router**

➤ Resequencer



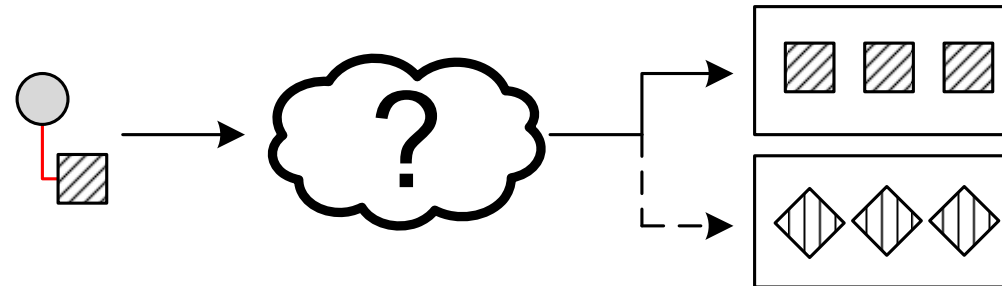
➤ Aggregator

➤ Literaturverzeichnis

Contend Based Router (1)

Problem

- Zielgerichtetes Senden von Nachrichten



Lösung

- Nachricht analysieren und spezifisch weiterleiten
 - Existenz eines bestimmten Feldes
 - Inhalt
 - Messagetyt

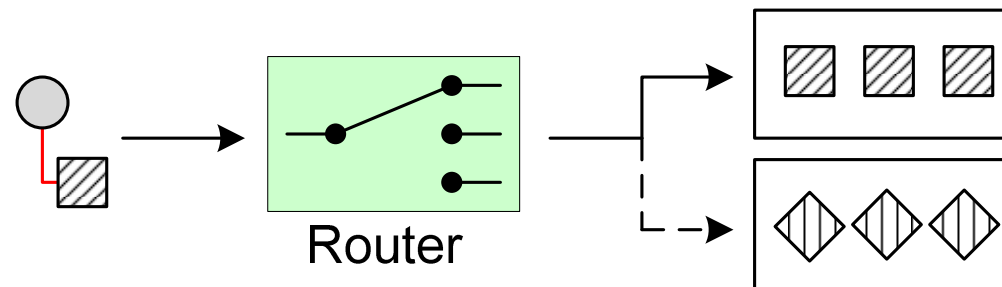


Abbildung nach [HW03, Seite 232]

Eigenschaften

- Ein Eingang, mehrer Ausgänge
- Zustandslos
- Routing-Tabelle
- Keine Änderung der eingehenden Nachrichten

Vorteile

- Routing-Logik im Router (nicht in Anwendungen)
- Hohe Flexibilität durch Rule-Engines
- Weiterleitung nur an spezifisches System
- Lose Kopplung
- Nur wenige zusätzliche Pipes



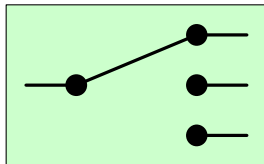
Bild: <http://www.erfolgreiche-firmengruendung.de>

Nachteile

- Single Point of Failure
- Evtl. viele Anpassungen am Router
- Flaschenhals
- Viele Router erhöhen Gesamtkomplexität



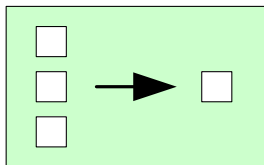
Bild: <http://www.erfolgreiche-firmengruendung.de>



✓ Einführung

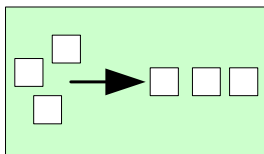
✓ Kontext

✓ Messaging



✓ Contend Based Router

➤ **Resequencer**



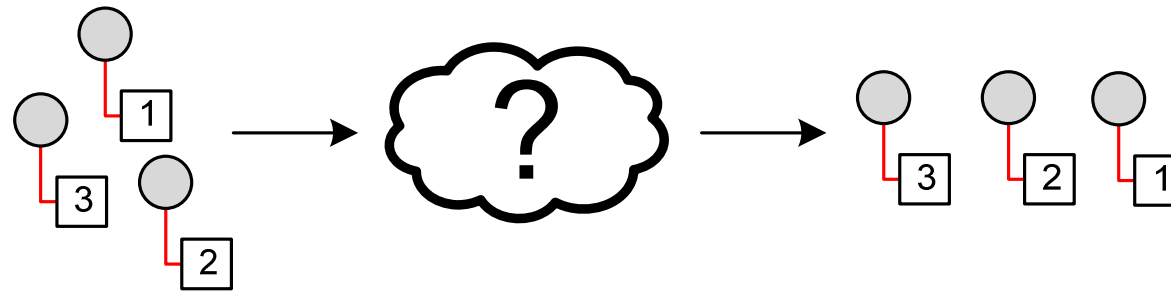
➤ Aggregator

➤ Literaturverzeichnis

Resequencer (1)

Problem

- Nachrichtenreihenfolge wichtig
 - Unterschiedliche Routen der Nachrichten
 - Verzögerungen durch Komponenten möglich



Lösung

- Pufferung der Nachrichten bis Reihenfolge sichergestellt

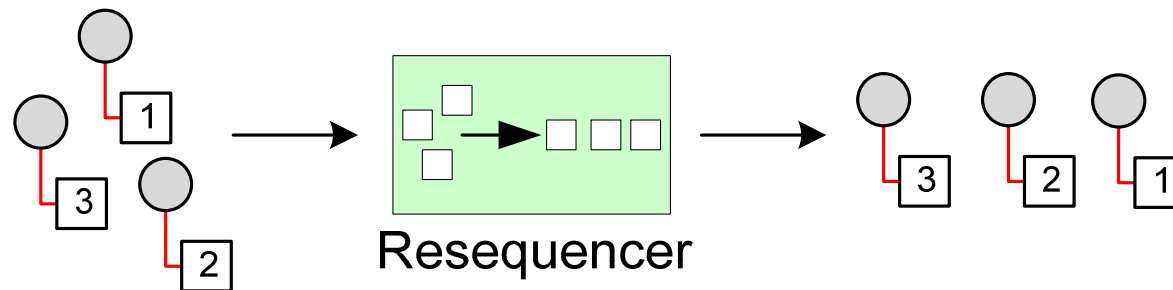
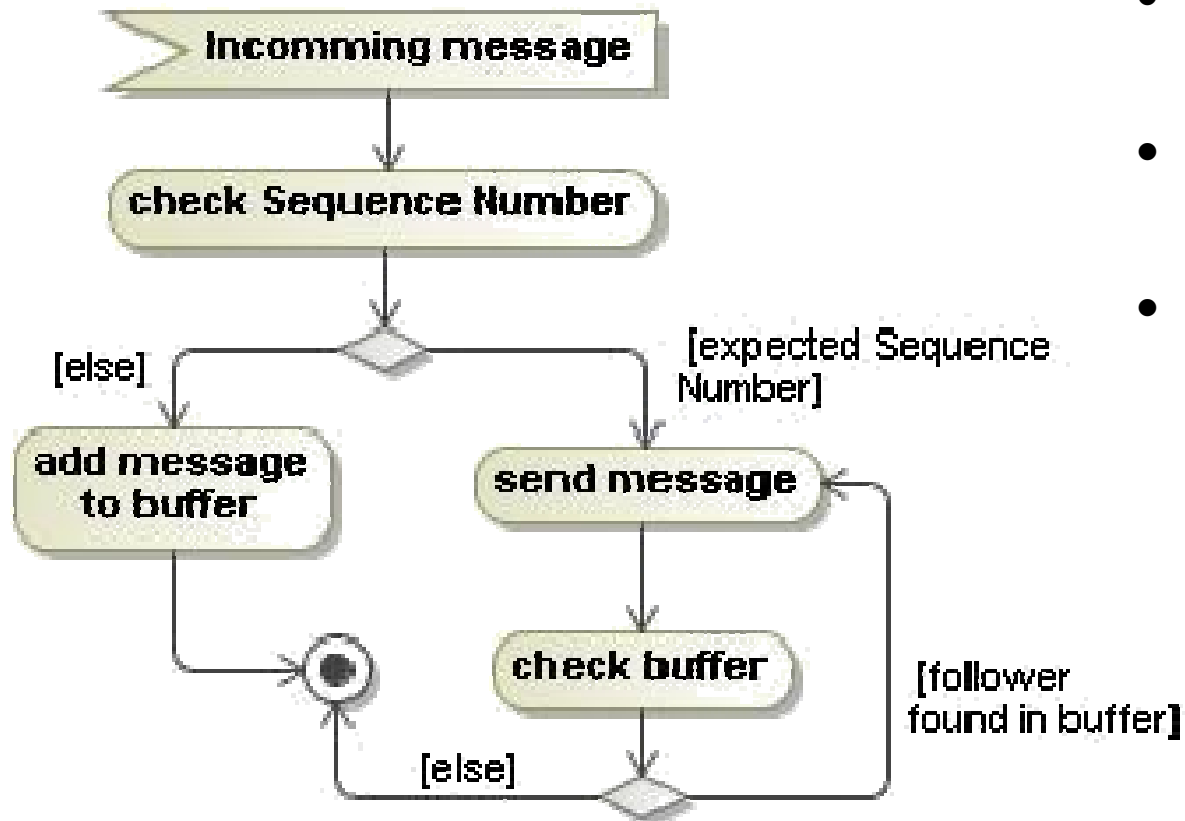


Abbildung nach [HW03, Seite 284]

Eigenschaften

- Ein Eingang, ein Ausgang
- Keine Änderung der eingehenden Nachrichten
- Eindeutige, aufeinanderfolgende Sequence Number im Header
- Zustandsbehaftet



Beispiel

- Empfang Nachricht 3
 - Speicher in Puffer
- Empfang Nachricht 2
 - Speicher in Puffer
- Empfang Nachricht 1
 - Sende Nachricht 1
 - Sende Nachricht 2
 - Sende Nachricht 3

Vorteile

- Reihenfolge der Nachrichten kann garantiert werden
- Einfache Basisimplementierung



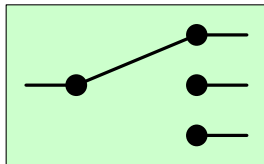
Bild: <http://www.erfolgreiche-firmengruendung.de>

Nachteile

- Erhöhung der Latenzzeit
- Generierung der Sequence Number evtl. Falschenhals
- Buffer-Overrun
- Missing-Message
- Single Point of Failure
- Erhöhung der Laufzeit (notwendig)



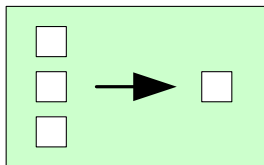
Bild: <http://www.erfolgreiche-firmengruendung.de>



✓ Einführung

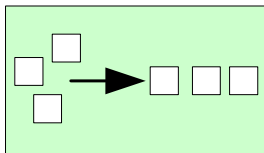
✓ Kontext

✓ Messaging



✓ Content Based Router

✓ Resequencer



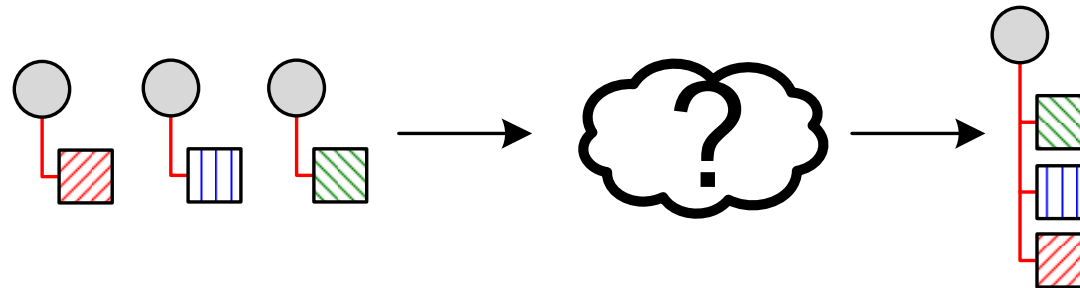
➤ **Aggregator**

➤ Literaturverzeichnis

Aggregator (1)

Problem

- Inhaltlich zusammenhängende Nachrichten zusammengefasst verarbeiten



Lösung

- Pufferung der Nachrichten
- Aggregation
- Einzelnachricht senden

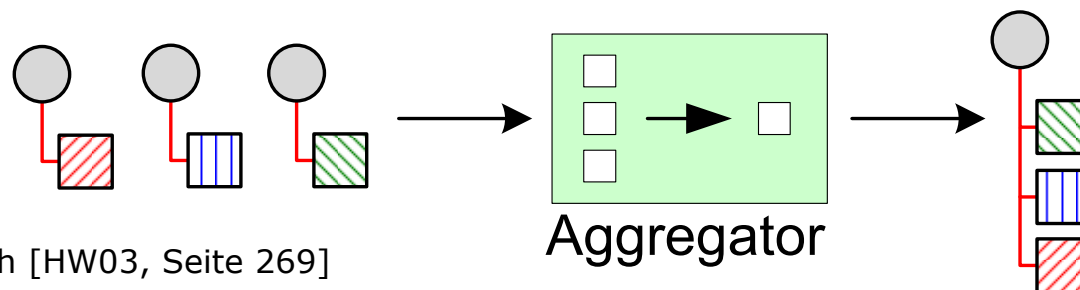


Abbildung nach [HW03, Seite 269]

Eigenschaften

- Ein Eingang, ein Ausgang
- Zustandsbehaftet
- Änderung der Nachrichten

Aggregator (3)

Korrelation

- Message Typ
- Correlation Identier

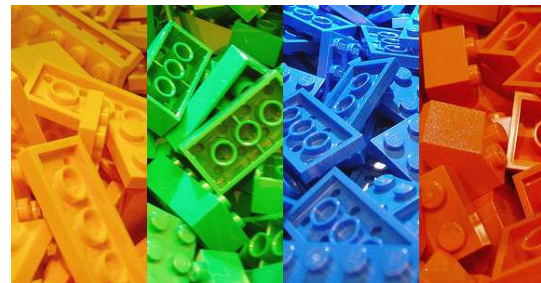


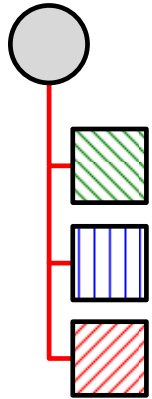
Bild: <http://www.fair-square.de>

Vollständigkeit

- Wait for All
- Timeout
- First Best
- Timeout with Override
- External Event



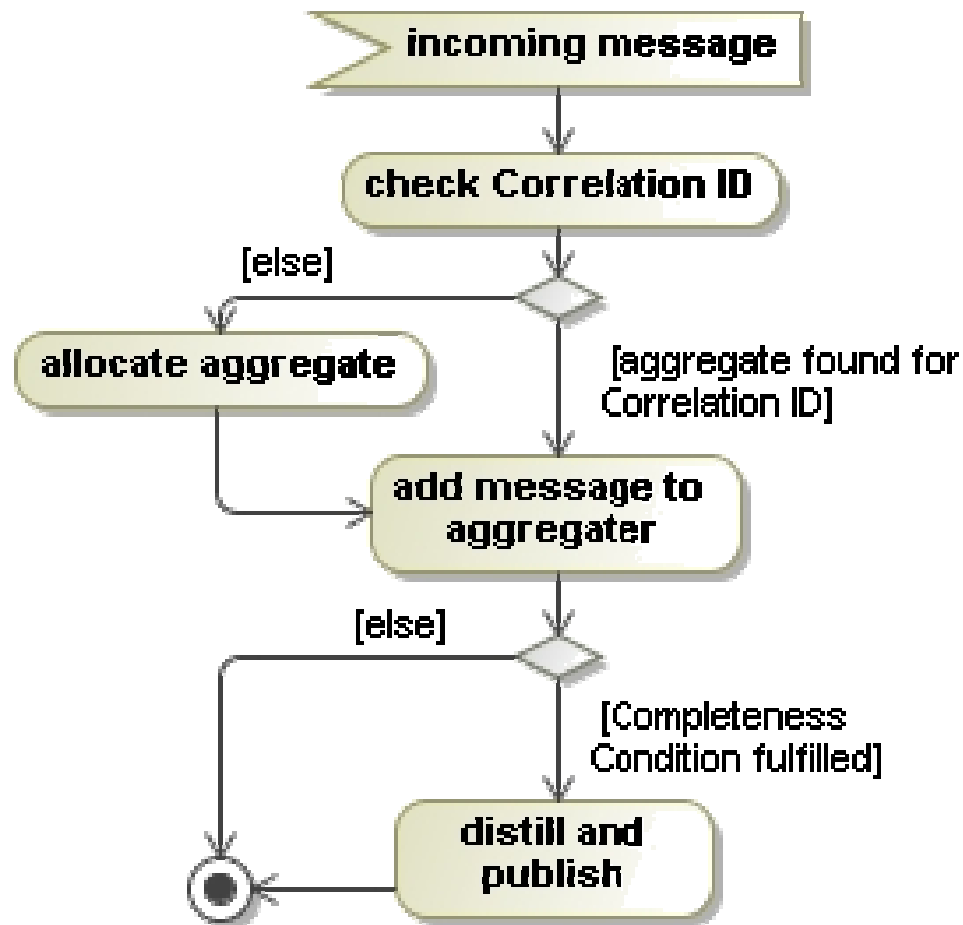
Aggregator (4)



Aggregierung

- „Best“ answer
- Verdichtung
- Verpackung

Aggregator (4)



Beispiel (Correlation ID 4711)

- Empfang Nachricht 1
 - Aggregat anlegen
 - Speicher in Aggregat
- Empfang Nachricht 2
 - Speicher in Aggregat
- Empfang Nachricht 3
 - Speicher in Aggregat
 - Vollständigkeitskriterium erfüllt
 - Sende aggregierte Nachricht
 - SchlieÙe Aggregate

Vorteile

- Zusammenfassen der Nachrichten transparent
- Vorgelagerte Aufbereitung



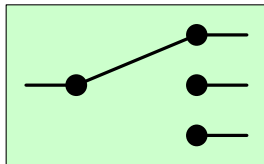
Bild: <http://www.erfolgreiche-firmengruendung.de>

Nachteile

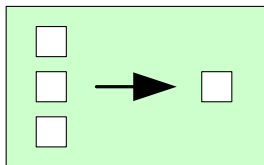
- Hoher Ressourcenbedarf
- Buffer-Overrun
- Management der Komplexität
- Correlation-ID notwendig
- Single Point of Failure
- Erhöhung der Laufzeit



Bild: <http://www.erfolgreiche-firmengruendung.de>



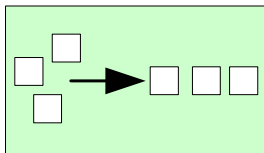
✓ Einführung



✓ Kontext

✓ Messaging

✓ Contend Based Router



✓ Resequencer

✓ Aggregator

➤ **Literaturverzeichnis**

Literaturverzeichnis

- [HW03]** Gregor Hohpe and Bobby Woolf. Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, October 2003.
- [BHS07]** Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt. Pattern-Oriented Software Architecture Volume 4: A Pattern Language for Distributed Computing. Wiley, May 2007.
- [AIS77]** Christopher Alexander, Sara Ishikawa, and Murray Silverstein. A Pattern Language: Towns, Buildings, Construction. Center for Environmental Structure Series. Oxford University Press, New York, August 1977.
- [Pap06a]** Prof. Dr. Christian Pape. Enterprise Application Integration (EAI) - Teil 1. http://www.home.hs-karlsruhe.de/pach0003/eai/01_eai_einfuehrung.pdf (24.04.2009), Hochschule Karlsruhe, 2006. Folien zur Vorlesung.
- [Pap06b]** Prof. Dr. Christian Pape. Enterprise Application Integration (EAI) - Teil 6. http://www.home.hs-karlsruhe.de/pach0003/eai/06_patterns.pdf (24.04.2009), Hochschule Karlsruhe, 2006. Folien zur Vorlesung.
- [Fou]** The Apache Software Foundation. Apache Camel - Enterprise Integration Patterns. <http://camel.apache.org/enterprise-integration-patterns.html> (24.04.2009).

Vielen Dank für die Aufmerksamkeit