

Probeklausur Integrations- und Migrationstechnologien

2014/2015

1. Motivation & Heterogenität

1a) Beschreiben sie das Konzept einer horizontal organisierten IT. Begründen Sie warum dieses Konzept besser geeignet ist um schnell ändernde Unternehmensprozesse abzubilden.

- Keine Abkehr von den Basissystemen
- Aber: Trennung/Entflechtung der Systeme von Prozess- und Oberflächenlogik
- Definition und bereitstellen von grobgranularen Fachdiensten
- Beliebige Verwendung der Dienste in unterschiedlichen Prozessen
- Eine einheitliche GUI welche den Benutzer durchgängig unterstützt
- Prozesse können leichter angepasst werden da die Logik nicht in einer einzelnen Applikation liegt sondern über einen ProcessManager angepasst werden kann. Damit ist eine schnellere Umsetzung möglich.

1b) Was versteht man unter Heterogenität von Softwaresystemen. Begründen Sie warum die Heterogenität Integrationstechnologien notwendig macht? Nennen Sie vier Arten von Heterogenität.

- Unterschiedliche Ausprägung von System-Eigenschaften Merkmalen wie z.B. Datenmodell, Schnittstellentechnologie, Eingesetzte Technologien und Systeme
 - Technische Heterogenität
 - Schnittstellen Heterogenität
 - Datenmodellbasierte Heterogenität
 - Logische Heterogenität (Semantik der Daten)
- Für die Integration der Daten ist es notwendig die Heterogenität zu überwinden so dass die Systeme miteinander kommunizieren können. Diese Transformation kann durch Integrationstechnologien vorgenommen werden

1c) Beschreiben Sie die sog. „technische Heterogenität“. Geben Sie dafür zwei Beispiele an.

Technische Heterogenität

Unterschiede auf der Ebene der technischen Infrastrukturen wie DBMS, Hardwareplattformen, Betriebssysteme und Netzwerkkomponenten

Beispiel:

Lagerverwaltungssystem (LVS) installiert auf MVS-Großrechner
E-Commerce-System (ECS) J2EE-Application auf Linux

2. Grundlagen

2a) Beschreiben Sie die Integrationsart „Shared Function Integration“

Zugriff auf gemeinsam genutzte Funktionen welche durch die Businesslogik exportiert werden.

Voraussetzungen:

- Die benötigten Funktionen müssen in der Quellapplikation verfügbar sein
- Direkter (remote) Zugriff auf die API

2b) Begründen Sie warum die Funktionsintegration der Datenintegration vorzuziehen ist. Beschreiben Sie ein mögliches Problemszenario. Gibt es Argumente, die in Einzelfällen für eine Datenintegration sprechen?

In der Business-Logik ist die eigentlich Logik einer Applikation abgebildet. Diese Logik wird umgangen wenn die Datenintegration benutzt wird, da nur auf die Datenbank zugegriffen wird.

Szenario: Automatisches auslösen von Bestellungen wenn ein Schwellwert unterschritten ist. Bei Datenintegration wird dies nicht ausgeführt da der zugehörige Programmcode nicht benutzt wird

Bei großen Datenmengen oder fehlenden Businessfunktionen kann die Datenintegration notwendig sein.

2c) Nennen und erläutern Sie drei Probleme, die beim Einsatz der Shared Function Integration zu berücksichtigen sind.

- Inefficient with large datasets.
- Programming-language specific.
- Tighter coupling.

2d) Nennen Sie zwei mögliche Kommunikationstechnologien über welche die Shared Function Integration realisiert werden kann. Nennen Sie eine bevorzugte (allgemein) und eine spezialisierte Technologie (hohe Kopplung).

Allgemeine Technologie: _____ SOAP _____

Spezielle/proprietäre Technologie: _____ RMI/CORBA _____

3. Kopplung & Architektur

3a) Beschreiben Sie das Konzept der losen Kopplung und begründen Sie warum die lose Kopplung als ein wichtiges Designprinzip in einer Integrationsumgebung gilt. Nennen Sie vier Arten durch die Kopplung entstehen kann.

- Lose Kopplung versucht Abhängigkeiten zwischen System zu reduzieren mit dem Ziel eine höhere Stabilität gegenüber Fehlern und Veränderung zu erreichen.
- Bei enger Kopplung ist im schlimmsten Fall das gesamte System bei einem Ausfall betroffen
- Vier Arten von Kopplung
 - Kopplung durch Kommunikation
 - Kopplung durch Datentypen
 - Kopplung durch Deployment
 - Kopplung durch Transaktionen

3b) Beschreiben Sie die Kopplungsart „Kopplung durch Kommunikation“. Geben Sie für die lose und enge Kopplung durch Kommunikation jeweils ein Beispiel an. Begründen Sie Ihre Aussage.

Kommunikation zwischen Systemen erfolgt durch austausch von one-way Nachrichten. Dieses Grundprinzip kann verwendet werden um komplexe Kommunikationsszenarien zu realisieren

Hohe Kopplung: **Request/Response (Synchron)**

- Request und Response werden über gleichen Kanal ausgetauscht (z.B. TCP/IP, HTTP)
- System A und B müssen gleichzeitig aktiv sein

Lose Kopplung: **Request/Response (Asynchron)**

- Nachrichten werden indirekt über eine Queue verschickt
- System A und B müssen nicht gleichzeitig aktiv sein
- Korrelation notwendig

3c) Begründen Sie, wann man als Architekt die enge Kopplung (Kommunikation) und wann die lose Kopplung verwenden sollte?

Enge Kopplung ist akzeptabel für System die z.B. im gleichen Unternehmen eingesetzt werden

Lose Kopplung: Für Systeme die in unterschiedlichen Unternehmen angesiedelt sind

Indikator: Entfernung der integrierten Komponenten. Je weiter entfernt umso eher ist lose Kopplung zu verwenden.

3. Kopplung & Architektur (Fortsetzung)

- 3d) Erläutern Sie das Konzept der idealen Kopplungsarchitektur. Begründen Sie warum die Entwicklung einer solchen Architektur wichtig ist für die Flexibilität und Stabilität einer Integrationslösung

Kopplungsarchitektur

- Die Kopplungsarchitektur einer Anwendungslandschaft legt für alle Komponenten fest, ob diese eng oder lose gekoppelt sind.
- Darüber hinaus wird festgelegt welche grundsätzlichen Mechanismen bei der Kopplung eingesetzt werden.

- 3e) Was wird unter der Entfernung von „Komponenten“ verstanden. Wie kann dieses Konzept bei der Definition der idealen Kopplungsarchitektur helfen?

Angemessene Kopplung – Entfernung von Komponenten

- Fachliche Entfernung:
 - Komponenten sind fachlich weit entfernt wenn Sie keine/wenig fachliche Gemeinsamkeiten haben.
 - Insbesondere wenn Sie unterschiedlichen Domänen oder verschiedenen Anwendungslandschaften angehören
- Technische Entfernung:
- Komponenten sind technisch weit entfernt wenn Sie unterschiedlichen Kategorien angehören

Allgemein gilt:

- Der angemessene Grad der Kopplung hängt von der inhaltlichen Entfernung zweier Komponenten ab.
- Inhaltlich weit entfernte Komponenten sollen lose gekoppelt werden.
- Nahe Komponenten können enger gekoppelt werden.

- 3f) Erläutern Sie die Kopplungsdimension „Verfügbarkeit“. Wann herrscht in dieser Dimension hohe Kopplung, wann lose Kopplung?

- Verfügbarkeit: Sind zwei Kommunikationspartner von der Verfügbarkeit der jeweiligen Systeme abhängig.
- Lose Kopplung: A erbringt den Anwendungsservice den es implementiert auch dann, wenn B oder die Kommunikationsverbindung zu B nicht verfügbar ist
- Hohe Kopplung: A kann den Anwendungsservice den es implementiert nicht erbringen wenn B oder die Kommunikationsverbindung zu B nicht verfügbar ist

4. Integration Architecture Blueprint

4a) Erläutern Sie den Unterschied zwischen einem Kommunikationsprotokoll und einem (Kommunikations-)format. Geben Sie jeweils zwei Beispiele an.

- (Kommunikations-)protokoll: Eine Vereinbarung, nach der die Datenübertragung zwischen zwei oder mehreren Parteien abläuft. In seiner einfachsten Form kann ein Protokoll definiert werden als die Regeln, die Syntax, Semantik und Synchronisation der Kommunikation bestimmen.
- (Kommunikations-)format: Ein Format definiert die Syntax und Semantik von Daten innerhalb eines binären Speichers. Es stellt damit eine bidirektionale Abbildung von Information auf einen eindimensionalen binären Speicher dar.

4b) Erläutern Sie das Konzept eines „kanonischen Datenformats“. Verwenden Sie hierzu das nachfolgende Architekturfragment. Nennen Sie Vorteile und begründen Sie wann dieses Konzept genutzt werden sollte. Begründen Sie warum dieser Ansatz eine höhere Flexibilität in Bezug auf Erweiterungen besitzt.



- Idee: Definition eines einheitlichen, allgemeinen Datenformats (kanonisches Format). Eingehende Daten werden in dieses Format transformiert. Ausgehende Daten werden aus dem kanonischen Format erzeugt.
- Es erfolgt eine Entkopplung der Quell und Zielsysteme.
- Der Integrationsaufwand für neue Quell- und Zielsystem ist linear
- Dieses Modell sollte benutzt werden wenn die Anzahl der integrierten Systeme variabel ist und jederzeit neue System hinzukommen können. Mit diesem Ansatz wird der notwendige Aufwand minimiert.
- Flexibel in Bezug auf Erweiterung da neue Systeme hinzugefügt werden können ohne die bestehende Integration zu ändern.

4c) Erläutern Sie die Aufgabe der Komponente „Process“ innerhalb des Architecture Blueprints. Nennen Sie einen möglichen Standard wie diese Komponente realisiert werden kann.

- Orchestrieren von Komponenten- bzw. Serviceaufrufen
- Steuert die Bausteine im Mediation Layer (falls diese nicht autonom reagieren können)
- Unterstützt die Umsetzung komplexer Integrationsprozesse
- Bestandteile
 - Workflow: Realisierung technischer Prozesse mit Hilfe von Prozessabläufen
 - Modellierung mit Hilfe von BPEL, BPMN, ...
 - Ausführung mit Hilfe von Process-Engines
 - Wichtigste Komponente der Process-Layer

5. Enterprise Integration Patterns

5a) Erläutern Sie das Grundprinzip eines Message-Channels und einer Message. Wie können diese Grundelemente für den Bau einer Integrationslösung benutzt werden?

Channels — Messaging applications transmit data through a *Message Channel*, a virtual pipe that connects a sender to a receiver

Die elemente wie z.B. Mapper, Transformation können über Channels verbunden werden. Dadurch kann eine Pipeline-Verarbeitung aufgebaut werden.

5b) Die Implementierung eines Message-Channels bestimmt wesentlich die Qualität der Integrationslösung. Was ist zu beachten, wenn Messages mit sehr großem Inhalt sicher übertragen werden sollen? Wie ist der Message-Channel zu implementieren?

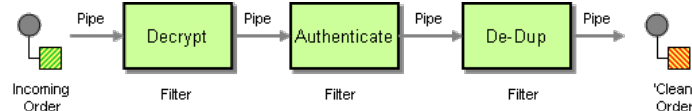
- Nachrichten mit einer großen Payload können den zur Verfügung stehenden Speicher überlasten.
- Solche Nachrichten können z.B. in einen sekundären Speicher ausgelagert werden
- Die Nachrichten werden nur bei Bedarf geladen
- Der Channel ist so zu implementieren dass er die übermittelten Nachrichte persistent speichert. Dadurch wird das System robust und der Speicherverbrauch reduziert.

5c) Was versteht man unter dem sogenannten Message-Routing. Erläutern Sie die Grundidee und geben Sie ein mögliches Einsatzbeispiel

Routing: Übermittlung der Nachrichten an ein oder mehrere Empfänger gesteuert durch ein Regelwerk, welches innerhalb eines Routers implementiert ist. Das Regelwerk definiert welche Empfänger die Nachricht zugestellt bekommt:

Beispiel: Materialien sollen an verschiedene Standorte eines Unternehmens verteilt werden. Anhand der Materialnummer kann entschieden werden ob ein Material relevant für den Standort ist. Falls ja, wird die Nachricht entsprechend geroutet.

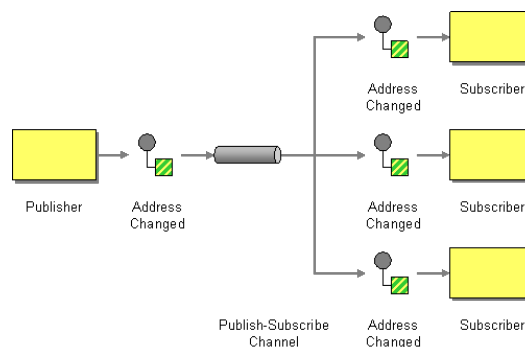
- 5d) Erläutern Sie die Arbeitsweise des nachfolgend dargestellten „Pipe- and Filter“-Ansatzes. Begründen Sie warum durch diesen Ansatz einen höhere Durchsatz und eine höhere Wieder-verwendbarkeit gegeben ist.



Each filter exposes a very simple interface: it receives messages on the inbound pipe, processes the message, and publishes the results to the outbound pipe. The pipe connects one filter to the next, sending output messages from one filter to the next. Because all component use the same external interface they can be *composed* into different solutions by connecting the components to different pipes. We can add new filters, omit existing ones or rearrange them into a new sequence -- all without having to change the filters themselves. The connection between filter and pipe is sometimes called *port*. In the basic form, each filter component has one input port and one output port.

Das System hat einen höheren Durchsatz weil mehrer Nachrichten gleichzeitig verarbeitet werden können.

- 5e) Erläutern Sie die Arbeitsweise des nachfolgend dargestellten „Publish-Subscribe Channel“-Ansatzes. Nennen sie ein Beispiel wann der Einsatz dieses Musters sinnvoll ist.



Send the event on a *Publish-Subscribe Channel*, which delivers a copy of a particular event to each receiver.

A *Publish-Subscribe Channel* works like this: It has one input channel that splits into multiple output channels, one for each subscriber. When an event is published into the channel, the *Publish-Subscribe Channel* delivers a copy of the message to each of the output channels.

Beispiel: Verteilung von Stammdaten. Jedes System welches an Stammdaten interessiert ist abonniert die entsprechenden Daten.

6 Transaktionen

6a) Beschreiben Sie das Grundprinzip einer Kompensations-Transaktion. Warum sollte die Transaktionsart bevorzugt eingesetzt werden? Stellt diese Transaktionsart eine lose oder hohe Kopplung dar?

Bei der Verwendung von Kompensationstransaktionen wird jede Systemaktivität als lokale, einzelne Transaktion ausgeführt. Darüberhinaus wird für jede Aktivität ein sog. „Kompensationsschritt“ aufgezeichnet, welcher in der Lage ist das Ergebnis ungeschehen zu machen.

Ist es nun notwendig eine Transaktion gesamtheitlich rückgängig zu machen werden alle Kompensationsschritte ausgeführt und dadurch die Einzelergebnisse zurückgesetzt.

Insgesamt Sieht es aus als ob der Gesamtvorgang nicht stattgefunden hätte.

Lose Kopplung da die der Aufwand für die Kommunikation der Systeme untereinander sehr gering ist und im Prinzip mit Standardmitteln umgesetzt werden kann.

6b) Erläutern Sie das nachfolgend dargestellte Zustandsübergangsdiagramm für einen Transaktionsteilnehmer (2-Phase-



Commit).

Der Transaktionskoordinator sendet an den Teilnehmer einen Vote-Request. Daraufhin entscheidet der Teilnehmer ob er für einen Commit (Vote-Commit) oder (Vote-Abort) stimmt. Bei einemm Vote-Commit verbleibt der Teilnehmer im Zustand Ready bis er den endgültigen Entscheid von dem Koordinator mitgeteilt bekommt. Der Teilnehmer dar im Zustand Ready seine entscheidung nicht revidieren. Im Falle von Global-Abort wird der Transaktion zurückgerollt. Bei Global-commit werden die Änderungen fest geschrieben. Entscheidet sich der Teilnehmer für einen Vote-Abort darf er sofort abbrechen.