

Software Engineering für große Informationssysteme

TU Wien, Sommersemester 2003

Bernhard Anzeletti
Wolfgang Keller
Rudolf Lewandowski
Klaudius Messner

Übung 2: **Spezifikation von Auszügen eines Provisionssystems**

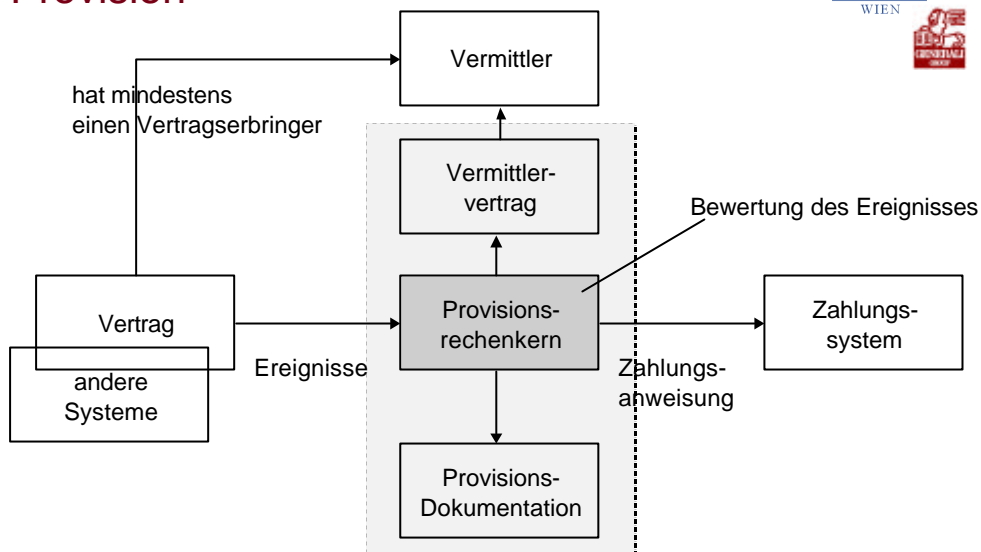
Für die zweite Übung sollten Sie von Ihrer eigenen Architektur ausgehen, die sie im ersten Teil der Übung erarbeitet haben. Sollten Sie aus irgendwelchen Gründen diese nicht verwenden können oder wollen, dann gehen Sie bitte von der nachfolgenden Architekturskizze aus, die einen groben Überblick über die Architektur eines Provisionssystems gibt.

Grundidee eines Provisionssystems

Wie Sie in der Angabe gesehen haben und in den VAA-Unterlagen, muss ein Provisionssystem sehr erweiterbar sein – und damit gut entkoppelt. Provisionsereignisse können nämlich in den verschiedensten Systemen eines Versicherungsunternehmens entstehen.

Die Grundvorstellung ist daher, dass irgendwelche Systeme im Unternehmen potenziell provisionsrelevante Ereignisse erzeugen.

Provision



Vorlesung „Software Engineering für große Informationssysteme“,
© 2003, Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner, All rights reserved, Technische Universität Wien, SS2003

Diese Ereignisse werden von einem Provisionsrechenkern bewertet und in Zahlungsanweisungen umgewandelt. Die Bewertung erfolgt anhand der Vermittlerverträge. Die dabei entstehenden Provisionen müssen natürlich revisionssicher dokumentiert werden (Provisionsdokumentation).

Damit haben wir hier eine Art Pipes & Filters Architektur vor uns. Das Provisionssystem bekommt eine Menge von Ereignissen, bewertet sie und generiert daraus Provisionszahlungen. Diese können von Folgesystemen verarbeitet werden.

Blöcke und Verantwortlichkeiten

Das Provisionssystem besteht also aus mindestens drei größeren Blöcken

- ? Der **Provisionsrechenkern** ist dafür verantwortlich, Provisionsereignisse (im obigen Bild Ereignisse) zu bewerten.
- ? Dafür bedient er sich der Information, die in den **Vermittlerverträgen** abgelegt ist
- ? Welche Provision aus einem Provisionsereignis entstanden ist, wird in der **Provisionsdokumentation** dokumentiert.

Damit sind auch schon die Verantwortlichkeiten und Subsysteme grob festgelegt.

Schnittstellen zu Nachbarsystemen

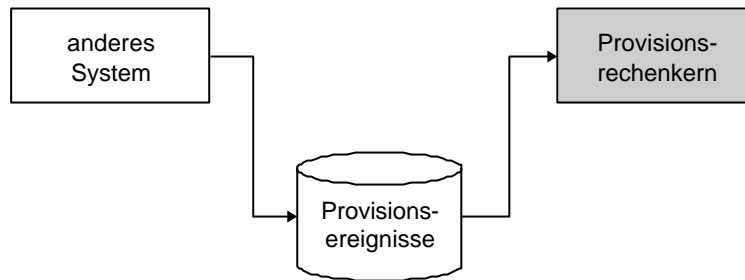
Da wir ein hochperformantes System zur Massendatenverarbeitung benötigen und außerdem im Prinzip eine Pipes and Filters Architektur vor uns haben, müssen wir noch überlegen, über welche Schnittstellen das Provisionssystem mit seinen vor- und nachgelagerten Systemen kommuniziert.

Übung zur Vorlesung „Software Engineering für große Informationssysteme“,
© 2003 Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner,
All rights reserved, Technische Universität Wien, SS2003

Was die Vorsysteme anlangt, geschieht das wie in der folgenden Abbildung dargestellt:



Übergabe an das Provisionssystem



Vorlesung „Software Engineering für große Informationssysteme“,
© 2003, Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner, All rights reserved, Technische Universität Wien, SS2003

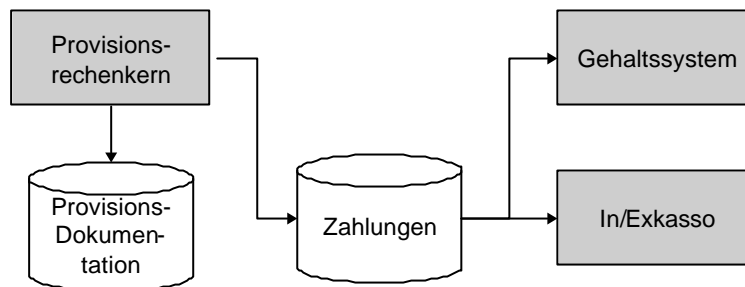
Der Provisionsrechenkern sollte zeitlich und auch was die Schnittstellen anlangt von anderen Systemen gut entkoppelt sein. Damit bietet es sich an, dass andere Systeme potenziell provisionsrelevante Ereignisse in einem Zwischenbestand ablegen. Dieser kann bei Bedarf vom Provisionssystem ausgelesen, optimal umsortiert und in Provisionen umgewandelt werden.

Ein solches Vorgehen hat in der Praxis verschiedene Vorteile:

- ? wenn man feststellt, dass ein Vorsystem fehlerhafte Daten geliefert hat, kann der Lauf schlicht neu gestartet werden und vorher die „falschen“ Sätze storniert werden.
- ? Eine zeitliche Entkoppelung ist dadurch möglich.
- ? Die Ereignisse können vor Verarbeitung so sortiert werden, dass eine optimal performante Verarbeitung möglich ist.

Analog gestaltet man auch die Übergabe aus dem Provisionssystem in die Zahlungssysteme. Es werden Teilzahlungen erzeugt. Diese können von einem Zahlungssystem noch passend gruppiert werden.

Übergabe der Zahlungen



Vorlesung „Software Engineering für große Informationssysteme“,
© 2003, Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner, All rights reserved, Technische Universität Wien, SS2003

Man bekommt dabei jedoch einen Verantwortungskonflikt, der aus der Abbildung nicht ersichtlich ist, nämlich ob die Gruppierung der Zahlungen noch vom Provisionssystem vorgenommen wird oder ob sie vom Exkassosystem vorgenommen wird.

Da ein Exkassosystem typischerweise nichts über Provisionsnoten weiß, wird es Sinn machen, wenn man die Gruppierung der Zahlungen noch dem Provisionssystem überlässt, das dann zu jeder Gesamtzahlung auch eine geeignete Provisionsnote mit der Erklärung der Einzelpositionen enthalten kann

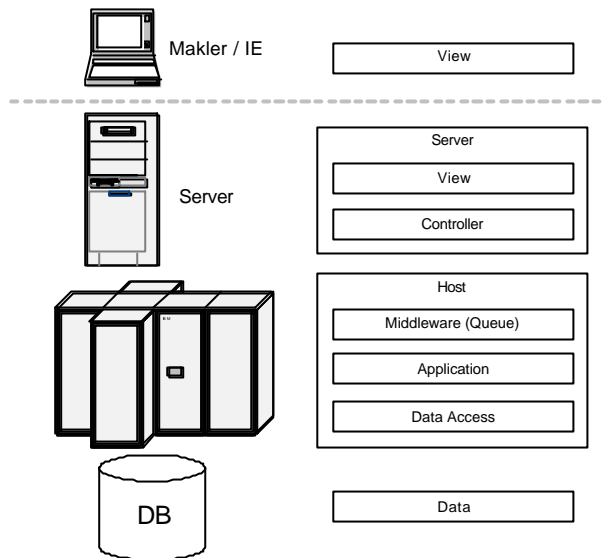
Systemarchitektur

Die Systemarchitektur sieht einen Host vor, der die Anwendungslogik und die Speicherung von Daten ermöglicht. Die Verwendung einer Middleware ermöglicht ein *Queuing* aller Ereignisse. Zudem ermöglicht die Middleware auf anderen Plattformen realisierten Fremdsystemen mit der Provisionsanwendung Daten auszutauschen. Zudem können dadurch auch Abfragen vom Userinterface übernommen werden.

Ein Server erlaubt die Ansicht der Provisionsnoten für Makler. Dadurch wird der Host nicht mit der Erstellung von Userinterfaces bzw. mit der Auswertung der Benutzerkommandos belastet.

Ein Makler hat eine HW-Infrastruktur, die nicht von der Versicherung gewartet wird. Die verwendete Browsertechnologie ermöglicht die Anzeige von Provisionsnoten auf verschiedensten Plattformen ohne Ausbreitung von zusätzlicher Software.

Systemarchitektur



Vorlesung „Software Engineering für große Informationssysteme“,
© 2003, Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner, All rights reserved, Technische Universität Wien, SS2003

2

Für die Administration der Vermittlerverträge wird die selbe Oberflächentechnologie für den Innendienst genutzt. Aus Sicherheitsgründen wird die Innendienstanwendung streng von der Makleranwendung getrennt.

Ist das objektorientiert?

Diese Frage wird sich sicherlich mancher stellen. Das komplette Design ist nicht objektorientiert – es ist eben ein Pipes & Filter Design, wobei es immer die Frage ist, wie man die Pipes implementiert.

Andere Dinge im Design, wie zum Beispiel die „provisionsrelevanten Ereignisse“ oder die Vermittlerverträge kann man gut als Objekte modellieren.

Aufgabe für die Übung 2

Es wird in dieser Übung Ihre Aufgabe sein, Teile des Provisionssystemes in Form von Objekttypen zu spezifizieren.

- ? Gehen Sie dabei von Ihrer eigenen Architektur aus, die sie im ersten Teil der Übung erarbeitet haben. Sollten Sie aus irgendwelchen Gründen diese nicht verwenden können oder wollen, dann gehen Sie bitte von der obigen Architekturskizze aus.
- ? Berücksichtigen Sie ferner jene Requirements, die unten noch angegeben werden.
- ? Spezifizieren Sie Objekttypen mindestens für
 - o Provisionsereignisse
 - o Provisionspositionen

Übung zur Vorlesung „Software Engineering für große Informationssysteme“,
© 2003 Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner,
All rights reserved, Technische Universität Wien, SS2003

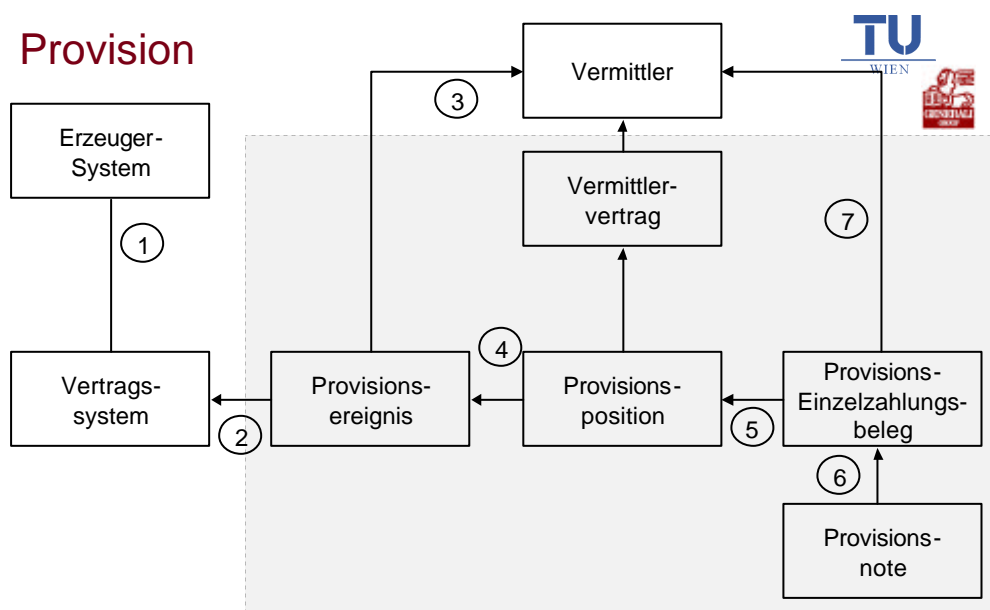
- Provisionseinzelzahlungsbelege
 - Provisionsnoten
- ? und orientieren Sie sich an der Darstellung weiterer Subobjekte, die unten folgt.
- ? Überlegen Sie sich, wie man einen Vermittlervertrag am besten aufbauen könnte. Denken Sie dabei speziell über das Thema „performante Verarbeitung“ nach. Nehmen Sie an, Sie müssen in einem Batch 4 Millionen Provisionsereignisse (Folgeprovision) für 15.000 Vermittler verarbeiten.
- ? **Zusatzaufgabe:** Wenn Sie noch Zeit haben, überlegen Sie sich, wie man die Arbeit der Leute, die Vermittlerverträge abschließen müssen, durch die Einführung eines Provisionsschemas (Rahmenvertrages) erleichtern kann, das pro Vermittler abgewandelt werden kann. Was ist dabei im Bezug auf das Thema Performance zu beachten?

Arbeiten Sie wieder in denselben Gruppen wie beim ersten Teil der Übung. Verwenden Sie zum Erfassen der Attribute CRC Cards und machen Sie sich irgendwo ein Blatt mit einem groben Datenmodell (nur Entities, keine Attribute).

Nach der Übung sollte die Arbeit wieder dokumentiert (Word oder pdf) und abgegeben werden. Ebenso erweitern Sie bitte die Präsentation aus dem ersten Teil mit den neuen Ergebnissen.

Weitere Informationen

In der folgenden Abbildung erhalten Sie weitere Informationen (fachlichen Input) den Sie noch verwerten sollten.



Vorlesung „Software Engineering für große Informationssysteme“,
© 2003, Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner, All rights reserved, Technische Universität Wien, SS2003

Übung zur Vorlesung „Software Engineering für große Informationssysteme“,
© 2003 Bernhard Anzeletti, Wolfgang Keller, Rudolf Lewandowski, Klaudius Messner,
All rights reserved, Technische Universität Wien, SS2003

- ? Die Beziehung (1) ist nicht weiter wichtig. Gehen Sie lediglich davon aus, dass es beliebig viele Systeme geben kann, die Provisionsereignisse erzeugen können.
- ? Die Beziehung (2) ist wichtig. Bei jedem Provisionsereignis (Ereignis, das potenziell eine Provisionszahlung auslösen kann) sollte festgehalten sein, aufgrund welchen Vorganges in welchem Erzeugersystem das Ereignis erzeugt wurde. Ferner muss (Beziehung (3)) das Ereignis einem oder mehreren Vermittlern zuzuordnen sein – aus Gründen der Einfachheit nehmen wir an, lediglich einem.
- ? Der Rechenkern erzeugt unter Umständen aus einem Provisionsereignis eine Provisionsposition – hier muss die Beziehung wieder verfolgbar sein (Beziehung (4)) und es muss weiter nachvollziehbar sein, aufgrund welcher Provisionsvereinbarung das passiert ist (Beziehung nicht nummeriert).
- ? Man wird aus einem Lauf des Rechenkernes die Ergebnisse pro Vermittler zu Einzelzahlungsbelegen zusammenfassen (Beziehung (5))
- ? Man wird dann weiter ab und zu alle Einzelzahlungsbelege zu Provisionsnoten zusammenfassen (Beziehung (6)) und erst diese an das Zahlungssystem weitergeben.

Weitere Requirements

- ? Die Provisionsnote sollte so gestaltet sein, dass der Vermittler daran sieht, für welche Verträge bei welchem Kunden er welche Provisionen erhalten hat. Ihre Einzelzahlungsbelege interessieren den Vermittler dabei wenig
- ? Provisionen müssen stornierbar sein. Es kann immer sein, dass sich jemand geirrt hat, dass Ereignisse falsch waren etc. Bedenken Sie, dass Stornierungen nicht durch Löschung behandelt werden dürfen, sondern nur durch kompensierende Buchungen. Auf der Provisionsnote sollen Stornos aufgrund interner Falschverarbeitung nicht sichtbar sein.
- ? Bei Storno eines neuen Vertrages innerhalb einer parametrisierbaren Garantieperiode muss der Vermittler einen parametrierbaren Teil der Provision zurückzahlen.