

3. Projektaufgabe DBS-II

Abgabe : 16.2.2004 Betreuer: Joos-Hendrik Böse
boese@mi.fu-berlin.de

2PC Koordination

Implementieren Sie einen Transaktionsmanager der die Überweisungen zwischen zwei Banken *A* und *B* mit Hilfe des 2PC Protokolls koordiniert.

Jede Bank verfügt dabei über eine eigene Datenbank, in der die Salden der Konten ihrer Kunden verwaltet werden. Die Datenbanken enthalten jeweils folgende Relationen:

- *Konto*(AccountNr, ClientName, CreditLine, Balance)
- *Transfer*(TransferNr, SourceBank, SourceAccount, DestAccount, Amount)

Programmieren Sie den Transaktionsmanager (TM) und einen Client der die lokalen Transaktionen der einzelnen Banken kontrolliert und mit dem TM über eine Socket Verbindung kommuniziert¹. Die Kommunikation zwischen der Bank und der Bank DB soll über JDBC realisiert werden².

Der Transaktionsmanager soll eine Liste mit Transaktionen in folgendem Format aus einer Datei einlesen können:

```
<!DOCTYPE Batch [  
  <!ELEMENT Batch (Transfer*)>  
  <!ELEMENT Transfer (SourceAccount, DestAccount, Amount)>  
  <!ELEMENT SourceAccount (#PCDATA)>  
  <!ELEMENT DestAccount (#PCDATA)>  
  <!ELEMENT Amount (#PCDATA)>  
  <!ATTLIST Transfer  
    id ID #REQUIRED>  
  <!ATTLIST SourceAccount  
    bank CDATA #REQUIRED>  
  <!ATTLIST DestAccount  
    bank CDATA #REQUIRED>  

```

Lokale Transaktionen scheitern, wenn z.B. die Kreditlinie für einen Kunden nicht ausreicht oder ein Konto zu einer bestimmten Kontonummer nicht existiert. Weiterhin soll der Ausfall eines Knotens simuliert werden. Bauen Sie dazu in jeder Phase des 2PC Protokolls im Clientprozess einen zufallsbasierten Abbruch ein, der den Ausfall des Clients simulieren soll. Die Wahrscheinlichkeit für einen Ausfall soll bei 0,03 liegen.

Der Inhalt der Nachrichten, die über Sockets zwischen TM und Client ausgetauscht werden, ist frei zu definieren. Sie können hier entweder die einzelnen Kommandos im XML-Format verschicken oder eine eigene Syntax definieren.

Um aufzuzeigen, was in dem verteilten System passiert, benutzen Sie bitte Log4j (<http://logging.apache.org/log4j/docs/index.html>) und protokollieren Sie jede Aktion der Clients und des TMs. Konfigurieren Sie Log4j so, dass alle Ihre Meldungen an einen zentralen Logserver geschickt werden und dort mit Hilfe eines Loganalyzers wie z.B. LogFactor5 aufbereitet werden.

¹Informationen zur Realisierung von Socketverbindungen in Java finden Sie unter <http://java.sun.com/docs/books/tutorial/networking/sockets/index.html>.

²Vergessen Sie hierbei nicht das Auto-Commit auszuschalten, ansonsten führt die Datenbank nach jedem JDBC Statement ein "Commit" aus, so dass Sie keine Möglichkeit haben per Abort eine Transaktion zurückzusetzen