Transaktionale Dateisystemoperationen

Herfried K. Wagner

Transaktionen?

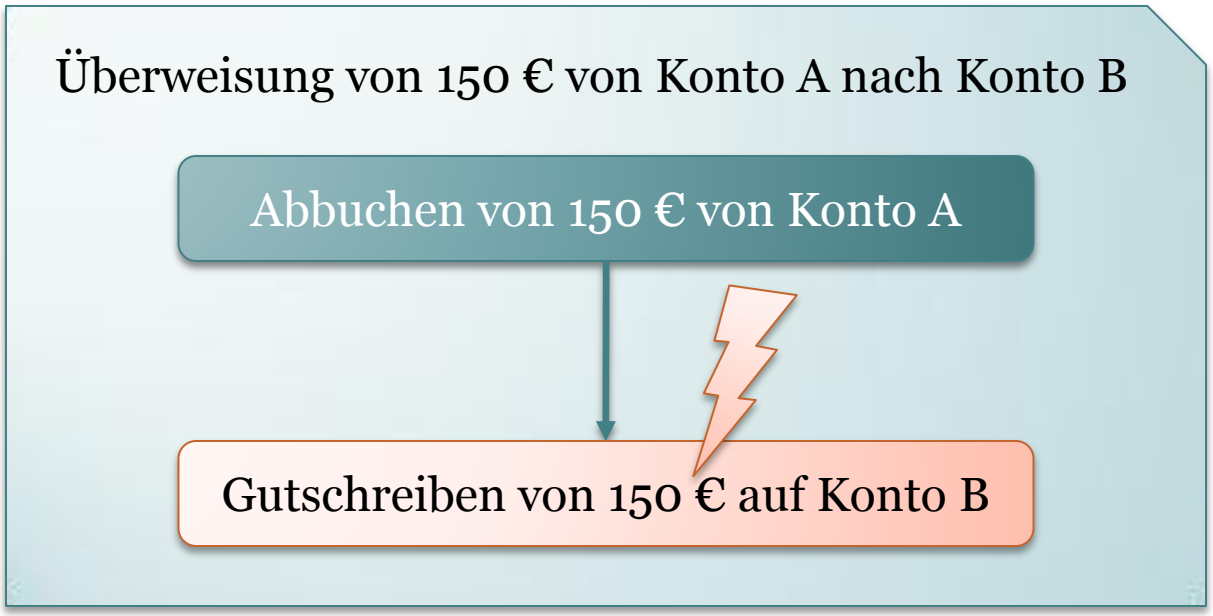
Anwendungsbeispiel

Ausgangspunkt

- Abfolge von z.T. zustandsverändernden Operationen auf Ressourcen
  - Lesen, Einfügen und Verändern in Datenbanken
  - Erstellen, Lesen, Schreiben und Verändern von Dateien
  - Zugriff auf Ressourcen wie die Systemregistrierung

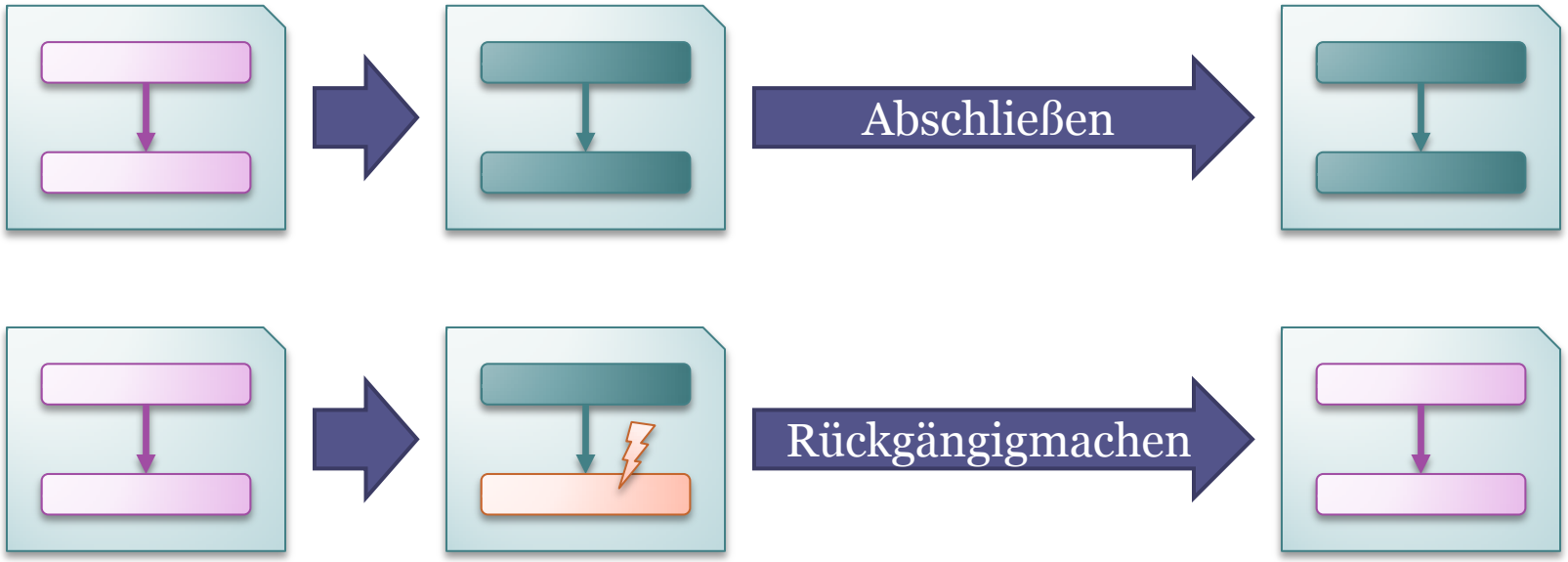
Bittere Erkenntnis

- Operationen können fehlschlagen



Ziel

- Atomare Ausführung der Abfolge von Operationen



Lösung

- Transaktion:
  - Abfolge von zustandsverändernden Operationen auf Ressourcen
  - Eigenschaften: ACID

ACID

Theoretischer Exkurs

Atomarität

- Entweder alle oder keine Operationen der Abfolge werden ausgeführt

Konsistenzerhaltung

- Konsistente Daten sind auch nach Abschluß der Transaktion konsistent

Isolation

- Auswirkungen einer in Ausführung befindlichen Transaktion sind für andere Transaktionen nicht sichtbar

Dauerhaftigkeit

- Die Ergebnisse einer Transaktion sind nach deren Abschluß dauerhaft und überdauern einen Systemausfall

Transaktionen in Windows

Verwendung in .NET-basierenden Programmen

Verfügbarkeit

- Windows Vista (z.T. SP1)
- Windows Server 2008

Transaktionales Dateisystem

- Common Log File System (CLFS)
  - Abschließen (Commit) von Transaktionen
  - Abbrechen (Abort) einschließlich des Rückgängigmachens (Rollback) von Transaktionen
- Transactional NTFS (TxF)

Transaktionen in Win32

- Win32-Kernel-API:
  - Dateisystemoperationen
  - Systemregistrierung
  - Nicht im .NET Framework gekapselt

Transaktionsverwaltung

- Kernel Transaction Manager (KTM)
  - Win32-KTM-API: *CreateTransaction* etc.
  - Stellt Transaktionsobjekte zur Verfügung
  - Nicht im .NET Framework gekapselt
- Distributed Transaction Coordinator (DTC)
  - Implementiert den OLE Transactions-Standard
  - Im .NET Framework gekapselt

Transaktionen in der Praxis

- APIs müssen transaktionale Ausführung unterstützen
- Andere zustandsverändernde Anweisungen (z.B. Änderungen an Zählervariablen) werden beim Rückgängigmachen der Transaktion nicht berücksichtigt

Die .NET-Klassenbibliothek
Transactions

Funktionalität von *Transactions*

- .NET-Kapselung von KTM und Win32-API für Dateisystemzugriff
- API angelehnt an das .NET Framework:
  - KTM: DTC
  - Win32-API für Dateizugriff: *System.IO*
- Mehrere Transaktionen auf eine Ressource parallel in mehreren Threads/Prozessen
- Geschachtelte Transaktionsbereiche

Die .NET-Klassenbibliothek
Transactions

Demo

Transaktionale Dateisystemoperation

Danke für die Aufmerksamkeit!

Herfried K. Wagner
dotnet.mvps.org
hkw@mvps.org

Dateioperationen mit Fortschrittsanzeige

Zugabe

Kopieren/Verschieben mit der Shell

- Funktion *SHFileOperation*
- Vordefinierte Systemdialoge
- Fortschrittsanzeige je Operationsabfolge
- Zeitschätzung
- Nicht transaktional ausführbar

Transaktionales Kopieren/Verschieben

- Win32-Kernel-API-Funktionen mit Rückrüfprozedur:
  - *CopyFileEx*, *CopyFileTransacted*
  - *MoveFileWithProgress*, *MoveFileTransacted*
- Fortschrittsinformation je Datei/Operation
- Wahlweise transaktional
- Eigene Fortschrittsanzeige möglich

Dateioperationen mit Fortschrittsanzeige

Kopieren/Verschieben mit Fortschritt

- Funktionalität in .NET-Klassenbibliothek *Transactions*
- Wahlweise transaktional
- Fortschrittsinformation über Rückrufprozedur

Dateioperationen mit Fortschrittsanzeige

Demo

Dateioperationen mit Fortschrittsanzeige

Danke für die Aufmerksamkeit!

Herfried K. Wagner
dotnet.mvps.org
hkw@mvps.org