

evodionBatch

Autor: Dr. Rudolf Kruse

Datum: 29. November 2005, Version 1.0

© evodion Information Technologies GmbH

Inhaltsverzeichnis

1	Einleitung.....	3
1.1	Nicht alle Geschäftsprozesse sind interaktiv	3
1.2	Was ist Batch-Betrieb?	3
1.3	Beispiele für Batch-Prozesse.....	3
1.4	Oft keine befriedigende Unterstützung für Batch-Prozesse.....	4
2	evodionBatch	5
2.1	evodionBatch	5
2.2	Java-Framework	5
2.3	Trennung von Technik und Fachlichkeit	5
2.4	Laufzeitarchitektur	5
2.5	Testunterstützung, Performance-Messungen.....	6
2.6	JobEditor.....	7
2.7	JobGenerator.....	7
2.8	JobMonitor	8
2.9	Systemvoraussetzungen	8

1 Einleitung

1.1 Nicht alle Geschäftsprozesse sind interaktiv

Auch wenn man bei Enterprise Applications vorwiegend an Dialoganwendungen denkt, müssen unternehmenskritische Systeme immer auch Geschäftsprozesse mit abdecken, die sich nur im Batch-Betrieb angemessen realisieren lassen.

1.2 Was ist Batch-Verarbeitung?

Im Unterschied zum Dialogbetrieb, in dem Anwender einzelne Fälle über graphische Benutzungsoberflächen individuell bearbeiten, werden im Batch-Betrieb Geschäftsprozesse in gleichförmige Weise auf großen Datenbeständen angewendet, ohne dass Benutzereingaben möglich oder nötig wären. Die Datenbestände werden dabei über Eingabedateien oder Datenbankanfragen beschrieben. Batch-Prozesse müssen transaktionsorientiert arbeiten und Reports erzeugen, um ihre Wirkungen zu dokumentieren. Für die Massendatenverarbeitung ist wichtig, dass hohe Durchsätze erzielt werden, um vorgegebene Laufzeitanforderungen einhalten zu können. Genauso wichtig ist, dass der Speicherverbrauch dabei konstant bleibt, also nicht mit der Anzahl der zu verarbeitenden Datensätze ansteigt.

Batch-Verarbeitung wird auch „Offline-Verarbeitung“ genannt und ist im Zusammenhang mit Geschäftsprozessen auch als „Dunkelverarbeitung“ bekannt.

1.3 Beispiele für Batch-Prozesse

Unsere Kunden erledigen mit evodionBatch gleichermaßen fachliche wie technische Aufgaben. Beispiele für fachlich motivierte Batch-Prozesse sind:

- Die Jahreszinskapitalisierung von Kundenkonten.
- Die Erzeugung von Prolongationsangeboten zu Geldanlagen. Der Batch-Prozess ermittelt alle Verträge, deren Festschreibungsfrist innerhalb des Folgemonats ablaufen. Zu jedem Vertrag erstellt und versendet der Batch-Prozess einen Brief an den Kunden mit dem Angebot zur Auszahlung oder Wiederanlage. Zudem erzeugt der Batch-Prozess zu jedem Vertrag im Workflow-Management-System einen Geschäftsprozess für den zuständigen Sachbearbeiter und legt ihn zum Ende der Festschreibungsfrist auf Wiedervorlage.
- Die Erfüllung gesetzlich vorgegebener Meldepflichten, z. B. nach § 24c KWG.
- Der Abgleich von Kundenanschrift-Daten mit den Beständen externer Dienstleister.

Beispiele für technisch motivierte Batch-Prozesse sind:

- Datenabgleiche mit anderen Systemen des Unternehmens.
- Datenmigration bei Ablösung von Altsystemen.

Den verschiedenen Aufgaben entsprechend werden die Batch-Prozesse unterschiedlich oft ausgelöst. Typische Häufigkeiten sind einmalig, täglich, monatlich oder jährlich.

1.4 Oft keine befriedigende Unterstützung für Batch-Prozesse

Die IT-Infrastruktur in den Unternehmen unterstützt interaktive Anwendungen meistens gut, Batch-Anwendungen jedoch kaum. Batch-Prozesse verlangen nach einer soliden Batch-Laufzeitumgebung, werden jedoch oft über unbefriedigende Zweckentfremdungen mit im Dialogsystem untergebracht oder über andere fragwürdige Provisorien realisiert, bei denen vorbei an der Geschäftslogik auf die DB-Daten zugegriffen wird oder die nicht transaktionssicher sind.

2 evodionBatch

2.1 evodionBatch

evodionBatch bietet ein Framework mit einer durchdachten Laufzeitumgebung für Geschäftsprozesse im Batch-Betrieb. Die Laufzeitumgebung bearbeitet die Prozessdatensätze nebenläufig und transaktionsorientiert. Sie skaliert dabei prinzipiell beliebig. Das Framework unterstützt den Entwickler und den Administrator von Batch-Prozessen mit Werkzeugen. Dazu gehören ein JobEditor und ein JobGenerator, mit denen Batch-Prozesse modelliert und generiert werden können. Der JobMonitor ist eine Anwendung, mit deren Hilfe laufende Batch-Prozesse überwacht und konfiguriert werden können.

2.2 Java-Framework

Das Java-Framework evodionBatch ist die ideale Ergänzung zu allen bestehenden oder geplanten Dialogsystemen, die in Java/J2EE entwickelt wurden oder werden. Nur mit einer einheitlichen Entwicklungsplattform können sowohl Dialog- als auch Batch-Prozesse auf der gleichen Geschäftslogik aufbauen.

2.3 Trennung von Technik und Fachlichkeit

Grundsätzlich trennt evodionBatch – wie alle guten Frameworks – konsequent die technischen Aspekte von den fachlichen Inhalten einer Anwendung. Mit dem Batch-Framework kann der Entwickler einen Batch-Prozess entwerfen und implementieren, indem er sich ausschließlich um die fachlichen Belange des zugehörigen Geschäftsprozesses kümmert.

2.4 Laufzeitarchitektur

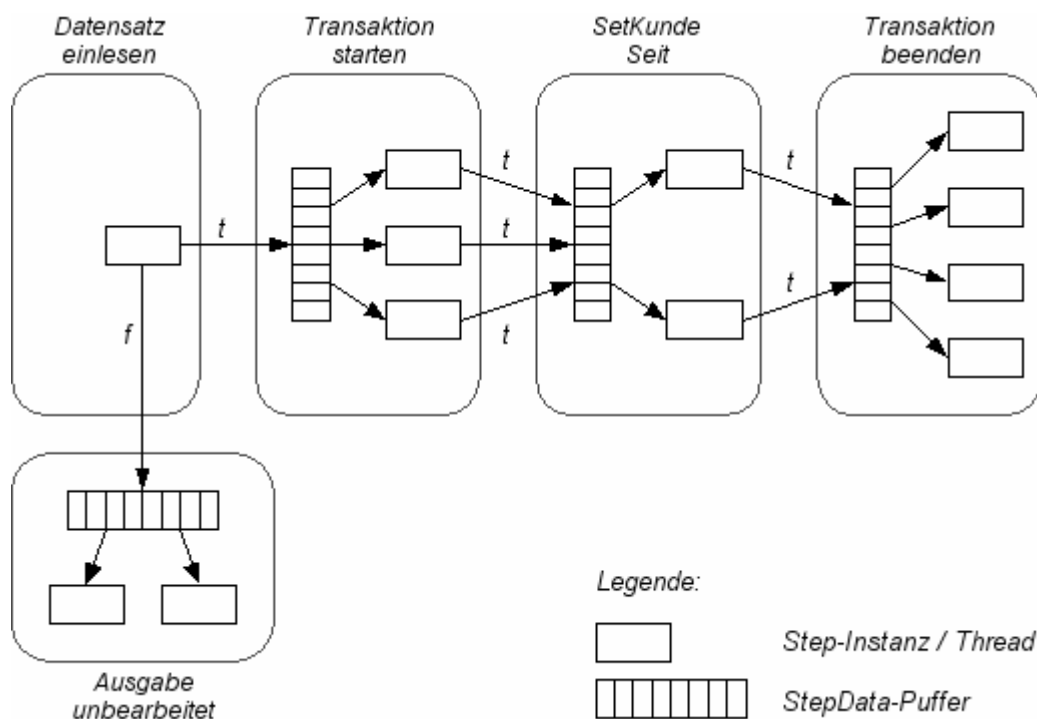
Ein Batch-Prozess ist in evodionBatch aus mehreren aufeinander folgenden Schritten aufgebaut, die ein zu verarbeitender Datensatz durchläuft. Jeder Schritt bearbeitet den jeweils aktuellen Datensatz und erledigt dabei eine bestimmte fachliche Teilaufgabe des Prozesses. Alle Schritte arbeiten gleichzeitig, so dass der erste Schritt bereits den nächsten Satz bearbeiten kann, während der zweite noch mit dem Vorherigen beschäftigt ist. Dadurch nutzt evodionBatch den Pipeline-Effekt aus, der Fließbandverarbeitung so effektiv macht.

Eine zweite Dimension der Parallelisierung kommt dadurch hinzu, dass evodionBatch die einzelnen Schritte in mehreren Instanzen startet. Jede Instanz eines Schritts arbeitet wiederum parallel zu den anderen Instanzen des gleichen Schritts und parallel zu den Instanzen der anderen Schritte. Die Anzahl der Instanzen ist dabei für jeden Schritt individuell einstellbar. Man kann die Laufzeitunterschiede zwischen den einzelnen Schritten ausgleichen und somit die Gesamtlaufzeit eines Batch-Prozesses minimieren, indem man Schritte mit kurzen Laufzeiten mit wenigen Instanzen und Schritte mit langen Laufzeiten mit vielen Instanzen startet.

Die einzelnen Schritte sind zudem über Pufferspeicher miteinander verbunden, die eine konfigurierbare Anzahl von Prozessdatensätzen aufnehmen können. Wenn ein Schritt einen Datensatz bearbeitet hat, kann er ihn sofort in einen Pufferspeicher stellen und muss nicht warten, bis der nächste Schritt bereit ist,

den Satz zu übernehmen. Anschließend kann der Schritt sofort den nächsten Satz aus seinem Eingabepuffer nehmen und muss nicht warten, bis der vorherige Schritt einen Satz abzugeben hat. Sobald ein Batch-Prozess so konfiguriert ist, dass die Pufferspeicher stets gefüllt sind, kann jeder Schritt in jeder Instanz ständig unter Volllast arbeiten.

Insgesamt skaliert mit dieser Architektur und dem hohen Grad an Nebenläufigkeit über zwei Dimensionen evodionBatch grundsätzlich beliebig. Die Skalierbarkeit ist lediglich durch die verfügbaren Ressourcen wie Rechenleistung und Speicherplatz der zugrundeliegenden Plattform begrenzt. Die Laufzeitarchitektur und die Konfigurationsmöglichkeiten erlauben, Batch-Prozesse optimal auf die vorhandene IT-Infrastruktur anzupassen und damit Durchsätze zu erreichen, die sonst nur mit sehr hohen Hardware-Investitionen möglich wären.



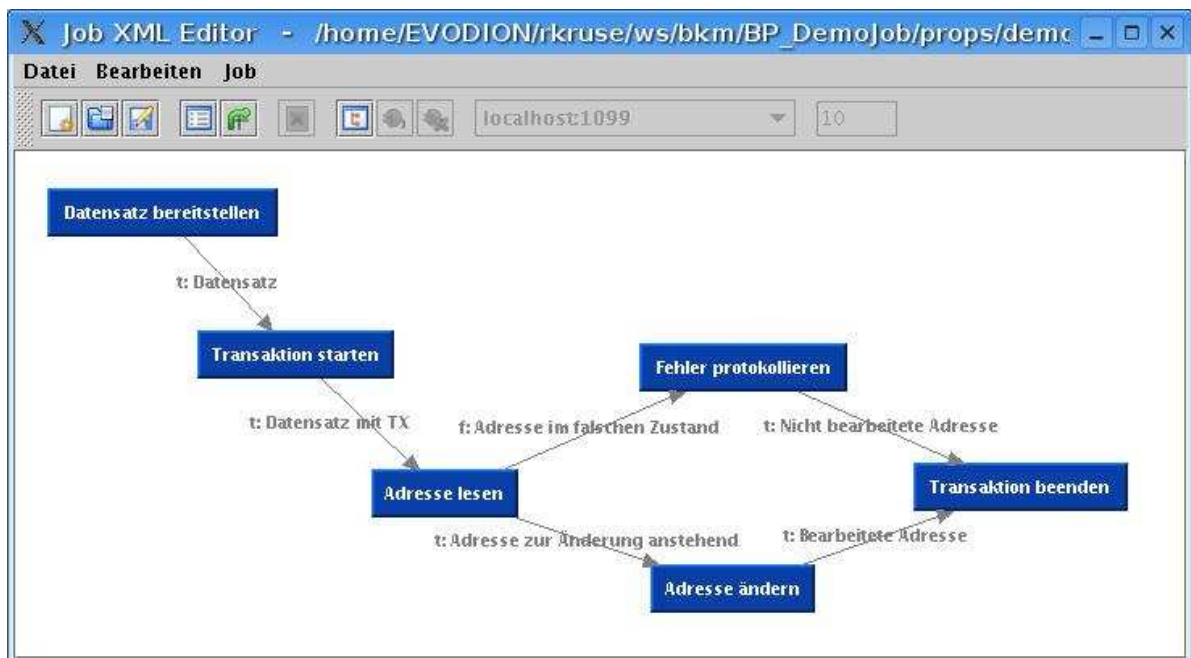
Ein Job mit fünf Schritten, jeder Schritt mit Pufferspeicher und eigener Anzahl Instanzen. Ablaufsteuerung über Verzweigung.

2.5 Testunterstützung, Performance-Messungen

In nebenläufigen Systemen sind Logfiles und Debug-Sitzungen sehr unübersichtlich. Um dies zu verhindern kann das Batch-Framework in einen besonderen Testmodus geschaltet werden, in dem die einzelnen Teilschritte des Prozesses sequentiell abgearbeitet werden. In dieser Betriebsart erstellt das Framework zudem umfangreiche Laufzeitstatistiken, aus denen ersichtlich ist, welche Laufzeiten für die einzelnen Schritte des Prozesses anfallen.

2.6 JobEditor

evodionBatch bietet unter anderem einen JobEditor als Modellierungswerkzeug, mit dem sich Batch-Prozesse graphisch entwerfen lassen. Das erstellte Modell wird von der Laufzeitumgebung als Ablaufplan verwendet und dient vor allem auch als Dokumentation des Batch-Prozesses. Die Prozess-Bausteine sind wiederverwendbar. Besonders hervorzuheben ist, dass man aus bestehenden Bausteinen neue Batch-Prozesse zusammenstellen kann, ohne programmieren zu müssen.



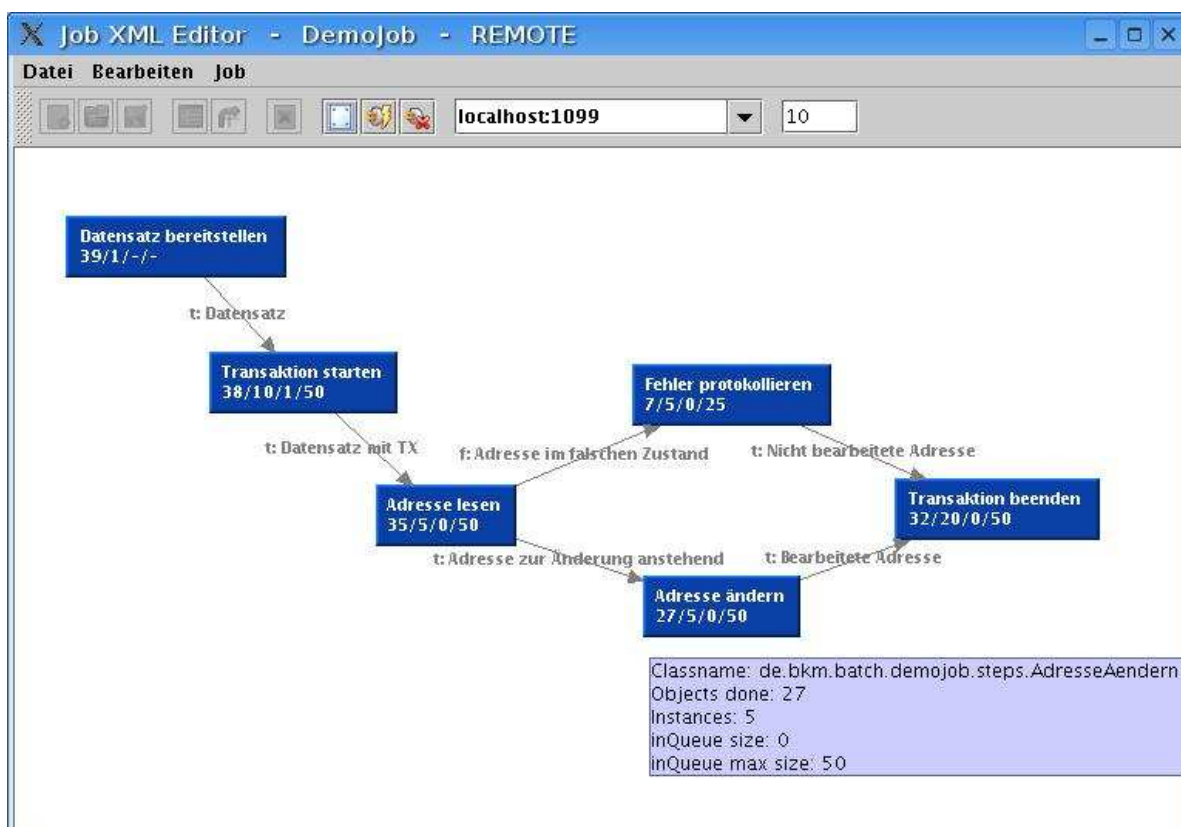
Screenshot: JobEditor mit DemoJob

2.7 JobGenerator

Zur Implementierung eines Batch-Prozesses kann der Entwickler bei evodionBatch auf einen JobGenerator zurückgreifen. Der Generator kann aus dem graphischen Modell, das mit dem JobEditor angefertigt wurde, eine initiale Implementierung erzeugen, die bereits die Rahmenimplementierungen der meisten Klassen und Testklassen sowie alle benötigten Infrastrukturdateien enthält, die das Modul zu einem Eclipse-Projekt mit fertigen Ant-Targets machen.

2.8 JobMonitor

evodionBatch stellt einen JobMonitor zur Verfügung, mit dem die Batch-Prozesse zur Laufzeit überwacht und gesteuert werden können. Diese Anwendung verbindet sich als Client mit einem laufenden Prozess und blendet in dem zugehörigen Modelldiagramm Informationen über die aktuelle Konfiguration und Auslastung des Prozesses ein. Zusätzlich kann man den Batch-Prozess über den JobMonitor im laufenden Betrieb umkonfigurieren und dabei z.B. die Thread-Verplanung oder die Kapazitäten der Pufferspeicher verändern, um die Ressourcenauslastung des Servers zu optimieren.



Screenshot: JobMonitor mit DemoJob

2.9 Systemvoraussetzungen

evodionBatch ist eine reine Java-Lösung und kann deshalb überall dort eingesetzt werden, wo eine Java-VM verfügbar ist. Unsere Kunden setzen evodionBatch beispielsweise auf IBM-Servern mit AIX und Sun-Servern mit Solaris ein. Als Datenbanksysteme werden DB2 und Oracle verwendet. Als Entwicklungs- und Testumgebungen dienen Linux- und Windows-Rechner mit SQL-Server- und Oracle-Datenbanken.