Überwachung von Aggregationszuständen in verteilten komponentenbasierten Datenproduktionssystemen

1) Anja Schanzenberger, 2) Colin Tully, 3) Dave R. Lawrence

1) GfK Marketing Services, Nordwestring 101, 90319 Nürnberg /
School of Computing Science, University of Middlesex, Bounds Green Road
London N11 2NQ, UK
2) 3) School of Computing Science, University of Middlesex, Bounds Green Road
London N11 2NQ, UK

Email: anja.schanzenberger@gfk.de, C.Tully@mdx.ac.uk, dave7@mdx.ac.uk

Abstract: Einer der signifikantesten Vorteile von Datenbanksystemen besteht in der Möglichkeit enthaltene Daten zu aggregieren und damit die Bildung beliebiger Gesamtbeträge, Statistiken oder auch Durchschnittswerte zu gewährleisten (aus vielen Datensätzen wird als Ergebnis ein Datensatz erstellt). Ebenso wichtig wie das Zusammenführen von Daten ist die Disaggregation. Darunter ist hier, im Gegensatz zur Aggregation, die Splittung von aufaggregierten Daten in ihre Einzelbestandteile zu verstehen (für den Betrachter werden aus einem Datensatz viele Ergebnisdatensätze).
In diesem Bericht werden nicht die mathematischen Möglichkeiten zur Aggregation und Disaggregation untersucht. Vielmehr zielt der Fokus auf die Nachvollziehbarkeit und damit auf die Überwachung dieser Vorgänge ab. Angenommen eine Aggregation wurde vorgenommen, wie können anschließend die eingegangenen Datensätze eindeutig wieder identifiziert und damit die Grundgesamtheit der Aggregation unmissverständlich abgeleitet werden, wenn die zu Grunde liegenden Datenbestände sich stetig ändern? Fragestellungen dieser Art werden in diesem Bericht aufgegriffen und untersucht.
An einem konkreten Beispiel aus der Wirtschaft werden die Probleme mit Aggregationszuständen erläutert. Das reale Beispielunternehmen, die GfK Marketing Services, eignet sich besonders gut für diese Fragestellungen, da ihr Kapital und ihre Geschäftsidee in der Verarbeitung und Veredelung äußerst großer Datenmengen besteht. Dabei spielt ein bald weltweit verteiltes Datenproduktionssystem in den vielen Niederlassungen des Unternehmens eine tragende Rolle. Für genau dieses komponentenbasierte verteilte Datenproduktionssystem wird demnächst ein Planungs- Control- und Monitoring System benötigt, das genaue Auskunft über enthaltene Dateninhalte im Datenproduktionssystem bieten muss. Dies erweist sich als Grundlage der Untersuchungen über die Aggregationszustände in diesem Bericht, welche daran erläutert, diskutiert und evaluiert werden sollen.

1. Einleitung

Die hier bedeutsamen Aggregationszustände entstehen nicht aus Datenmanipulation. Sie entspringen schlicht aus „GROUP-BY“-Anweisungen durch normale Abfragen. Bei derartigen Gruppierungen können wie bekannt sämtliche mathematischen Funktionalitäten einer Datenbank Verwendung finden (z.B. sum(), avg(), etc.) und damit *Aggregationen* durch *Aggregationsvorschriften* gebildet werden (siehe Abbildung 1). Beispielsweise entstehen aus mehreren Lieferperioden eine Auswertungsperiode.

Eine *Disaggregation* stellt das Gegenstück zur Aggregation dar und bedeutet deshalb die Erzeugung von Untermengen aus einem Datensatz. Von einer Auswertungsperiode sind bei der Disaggretation beispielsweise wieder die Lieferperioden abzuleiten. Aggregationszustände existieren jeweils vor beziehungsweise auch nach einer Dis-/Aggregation und sind damit eindeutig bestimmbar.

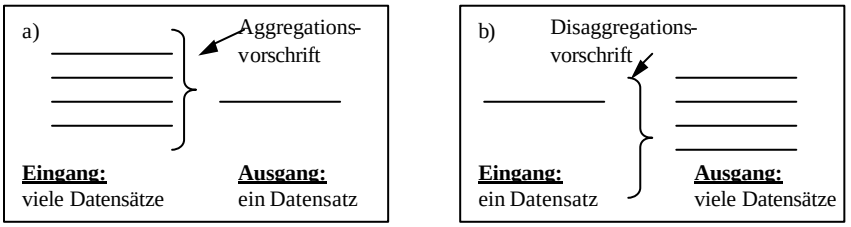

Abbildung 1: a) Aggregation b) Disaggregation

Am Beispiel der Produktionsumgebung und datenintensiven Umgebung der GfK Marketing Services [GfK02] finden die Untersuchungen zur Nachvollziehbarkeit von Aggregationszuständen statt.

Die GfK Gruppe ist eines der weltweit führenden Marktforschungsunternehmen. Deren Informationsdienstleistungen nutzen Industrie, Handel, Dienstleister und Medien für Marketing-Planungen und -Aktionen auf regionaler, nationaler sowie internationaler Ebene. Die GfK Marketing Services ist eines der vier Hauptgeschäftsfelder der GfK Gruppe. Sie hat sich auf Informations- und Beratungsservice im Bereich des Handels spezialisiert und nutzt die kontinuierliche Beobachtung von modernen technischen Gebrauchsgütern, um Berichte und Analysen über Marktsituationen zu gewinnen. Handelspanels, wie sie die GfK Marketing Services bietet, sind heute in der Lage Informationen höchster Präzision zu Verkauf, Verkaufspreis und Distribution zu liefern. Nicht nur deutschlandweit, sondern weltweit werden diese sogenannten Fakten gesammelt und zu Berichten extrapoliert und aufbereitet. Heute existieren weltweit bereits Tochtergesellschaften und Beteiligungen in über 40 Ländern.

IT-technisch gesehen ist die weltweite Verteilung der Niederlassungen eine große Herausforderung (siehe [Li01], [LH00]) für das in Deutschland bereits weitgehend existierende komponentenbasierte Datenproduktionssystem. Es handelt sich bei diesem um ein System, mit dem in unserem Fall Berichtsdaten gesammelt, veredelt und extrapoliert werden, im Gegensatz zu einem normalen Produktionssystem, welches zur Herstellung von güterorientierten Produkten, wie beispielsweise Autos, dient.

Es ist zu großen Teilen basierend auf COM-Technologie [Pa00] realisiert und wird stets weiterentwickelt. Produktionstechnisch werden die Daten von den lokalen Niederlassungen in die zentrale Niederlassung (Deutschland-Nürnberg) übertragen, dort gesammelt und schließlich zu Kundenberichten hochgerechnet. Dabei ist eine gemeinsame organisatorische Aufteilung der Produktionslandschaft, wie sie in Abbildung 2 zu finden ist, sehr hilfreich.

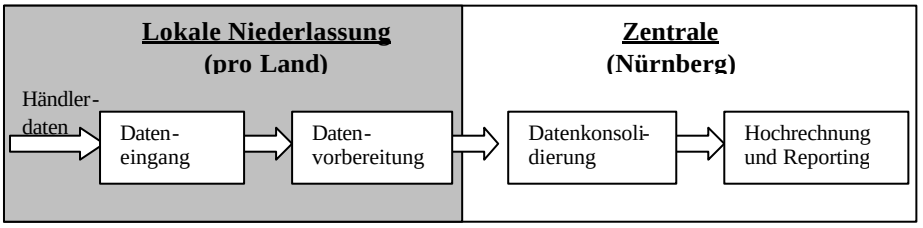

Abbildung 2: Organisatorische Sicht der Daten-Produktionsstrecke

Der Produktionsprozess beginnt mit dem (elektronischen) Einlesen der Daten beim sogenannten Dateneingang. Dieser lokale Bereich jeder Niederlassung ist zusätzlich für eine einheitliche Formatierung der Händlerdaten zuständig. Anschließend erfolgt durch die ebenfalls lokal befindliche Datenvorbereitung eine Zuordnung (Identifikation) von noch nicht (sonst automatisch) identifizierten Händlerartikeln zu einem Gfk-Artikelstamm. Alle identifizierten Händlerdaten fließen danach zur zentralen Datenkonsolidierung. Abschließend können nun die Extrapolation und das Reporting der Daten erfolgen.

Eine Herausforderung in naher Zukunft und Gegenstand der Diskussion in diesem Beitrag ist ein beabsichtigtes Planungs-, Control- und Monitoring System (PCMS) (für eine detaillierte Beschreibung sei der Leser auf [LS02] und [STL02] verwiesen), mit dessen Hilfe nicht nur die einzelnen Prozessschritte koordiniert, sondern auch eine systemtechnische Überwachung der beteiligten Produktionskomponenten erfolgen soll[1]. Als Besonderheit wird in diesem PCMS eine kontinuierliche inhaltliche Prüfung der Daten bzw. der dadurch repräsentierten Informationen (in unserem Fall: Verkaufszahlen) adressiert. Insbesondere die Problematik der permanenten Datentransformation (z.B. Aggregation und Splitten von Daten, aber auch Veränderung der Primärschlüssel) ist ein Grund dafür, dass bisher bestehende kommerzielle Werkzeuge zur Geschäftsprozessmodellierung (gemeint sind Workflow-Managementsysteme) sich nur bedingt für den Einsatz eignen und eine Eigenentwicklung auf einer soliden technischen Basis erzwungen wird.

Ziel und Inhalt dieses Beitrags ist es, Teile des nun grob skizzierten PCMS der GfK Marketing Services in Bezug auf die Nachvollziehbarkeit und Überwachung von Aggregationszuständen in derartigen datenintensiven Umgebungen zu untersuchen und konkrete Ansätze an diesem Beispiel zu diskutieren.

[1] Mit dem Teilbegriff „Control“ ist in diesem Zusammenhang eine Überwachung und nicht finanztechnisches Controlling gemeint.

2. Das Datenproduktionssystem der GfK Marketing Services

Um die Untersuchung von Aggregationszuständen für das geplanten PCMS definieren und motivieren zu können, ist es wichtig, einige Prozessabläufe des Datenproduktionssystems zu erläutern.

Jede Datenlieferung eines Händlers (bestehend aus einer Anzahl von verkauften Artikeln aus einer bestimmten Beobachtungsperiode) an den Dateneingang wird als Job bezeichnet. Jeder Job muss dabei das Produktionssystem durchlaufen. Alle Händlerartikel werden während des Prozesses entsprechenden GfK-Pendants (sog. GfK-Stammartikel) zugeordnet und weiteren Kategorisierungen wie z.B. Produktwarengruppen (Fernseher, Geschirrspüler, etc.) unterzogen. Dieses Ereignis wird als Identifikationsvorgang bezeichnet. Ein unbekannter Artikel wird dabei manuell durch Personal identifiziert, also einem sogenannten GfK-Stammartikel zugewiesen. Treffen nun in dieser oder auch in anderen Datenlieferungen weitere Artikel der Ausprägung des ersten Artikels ein, so repräsentiert der erste Artikel alle weiteren gleichartigen. Dies hat zur Folge, dass alle weiteren gleichartigen Artikel sofort automatisch erkannt und weitergeleitet werden, ohne dass erneut manuelles Eingreifen erforderlich ist.

Logische Objekte am Dateneingang (z.B. Datenlieferungen) stehen zum Ergebnis am Ende der Produktionskette (z.B. Produktwarengruppen) dabei in einer N:M-Beziehung. Ähnlich verhält es sich auf der zeitlichen Dimension, da aus ursprünglichen Datenlieferungen, die gemäß einer individuellen Lieferperiodizität (z.B. wöchentlich, monatlich, 2-monatlich) in das System eingespeist werden, Auswertungsperioden (z.B. wöchentlich, monatlich, ½-jährlich) für die Kundenberichte entstehen.

Besonderheiten:
Wie in [LS02] bereits ausführlicher dargestellt, unterscheidet sich das Datenproduktionssystem der GfK von klassischen Produktionssystemen. Beispielsweise wird eine Stichprobenerhebung von Händlerdaten vorgenommen, um die Marktereignisse wiederzuspiegeln. Ebenso kann es ausreichen, nur gewisse Prozentsätze der eingegangenen Daten auszuwerten, da durch statistische Hochrechnungen bereits aussagekräftige Ergebnisse erzielt werden können. Die für die Auswirkung auf die Aggregationszustände signifikanten Eigenschaften werden im folgenden noch einmal kurz vorgestellt.

- *Statische, jedoch konfigurierbare Produktionsabläufe*
  Bei der Produktion von Daten für die Erzeugung von Kundenberichten ist im wesentlichen ein fest vorgegebenes Prozessschema zu verfolgen, sodass eine flexible Entscheidung durch Evaluierung komplexer Bedingungen durch das PCMS nicht notwendig ist.
- *Daten als Produkt*
  Im Unterschied zur klassischen Produktion von Gütern kann es in dem vorliegenden Szenario vollkommen ausreichen, wenn der Sollzustand des Endprodukts nur zu einem bestimmten Anteil (z.B. 80%) erreicht ist. Die Erlangung von 100% des Sollzustandes würde bedeuten, dass alle Daten die in das Endprodukt „Bericht“ einfließen vollständig durch das gesamte Datenproduktionssystem gelaufen sind.

Jedoch bereits in einem unvollständigeren Zustand sind am Ende der Produktionskette ein Reporting zu realisieren und Trends zu erkennen. Dies beruht auch auf dem statistischen Stichproben-Prinzip.

- *Änderung der logischen Objekte*
  Datentransformation bedeutet, dass eingehende Objekte nicht mehr eindeutig produzierten Objekten zugeordnet sein können. Vielmehr existiert eine N:M-Beziehung zwischen Eingangs- und Ausgangsdaten. Im Gegensatz dazu sind in einer klassischen Produktionskette die Einzelteile im Endprodukt stets identifizierbar.

Die inhaltliche bidirektionale Abbildung der verwendeten Terminologien zwischen Anfang (Dateneingang) und Ende (Reporting) ist eine der wesentlichen Anforderungen an das PCMS. Konkret heißt dies, die Veränderung der Primärschlüssel zieht sich durch das ganze System und muss in jedem Fall nachvollzogen werden können. Die Veränderung von Lieferungen zu Auswertungen lassen somit keine 1:1 Beziehung zu.

3. Das Planungs- Control- und Monitoring System der GfK Marketing Services

Das PCMS setzt auf die Prozessabläufe des Datenproduktionssystems auf. Zum einen ist das Ziel die *Kontrolle* über den eben beschriebenen laufenden Betrieb zu übernehmen. Dabei müssen Auskünfte über den momentanen Zustand der Aufträge ebenso eingerichtet werden, wie schnelle Identifikation und Reaktion in Fehlersituationen. Zum anderen ermöglicht der Einsatz des PCMS eine *Optimierung* der Business Prozesse hin zu einer bedarfsgesteuerten Produktion. „Durch eine bedarfsgerechte Planung und Steuerung auf Anwendungsebene soll eine berichtsorientierte Produktion („*pull-based production*“) im Gegensatz zu der bisherigen lieferungsorientierten Produktion („*pushed-based production*“) stattfinden“ (siehe [LS02]). Eine gleichmäßigere Personalauslastung wird durch eine konsequente Verfolgung und Priorisierung der Aufträge erreicht werden. Daraus ergeben sich schließlich die nun folgenden Basisfunktionalitäten für das PCMS.

Basis-Aspekte des PCMS:

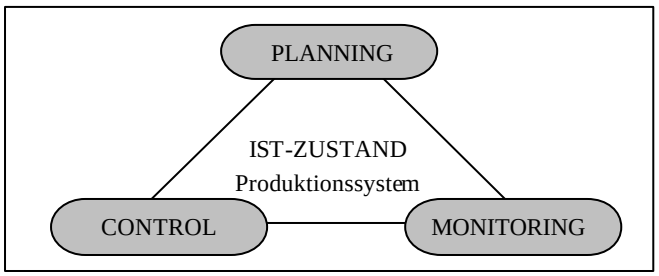

Abbildung 3: Basis-Funktionalitäten des PCMS

Die drei Eckpfeiler des PCMS (siehe Abbildung 3) bestehen im wesentlichen aus einer *Planungskomponente*, die einen *SOLL-Zustand* repräsentieren wird. Durch historischen Vergleich von Kennzahlen (z.B. Verkaufszahlen aus Vorperioden) soll für die gesamte Prozesskette ein TQM (Total Quality Management) stattfinden können. (Gute Planungsansätze können in [MG01] gefunden werden). Der zweite Eckpfeiler ist das *Monitoring.* Durch Monitoring wird der *IST-Zustand* im Produktionssystem abgebildet. Sowohl eine systemtechnische Überwachung der Hardware, als auch eine Überwachung der Jobs auf Anwendungsebene soll einen reibungslosen Prozessverlauf gewährleisten. Darüber müssen damit genaue Aussagen über die jeweiligen Bearbeitungszustände der einzelnen Aufträge erzielbar sein. Die Realisierung wird über ein Logging der beteiligten Komponenten stattfinden. Der dritte und letzte Eckpfeiler ist der *Kontrollteil* des PCMS. Die Möglichkeit steuernd auf den IST-Zustand im Datenproduktionssystem einzugreifen wird einerseits durch eine Vergabe von Job-Prioritäten erreicht. Andererseits werden auf systemtechnischer Ebene lokale Eingriffe zur Hardware und Software Überwachung eingerichtet.

Aspekte zur Überwachung von Aggregationszuständen

Durch die Basis-Funktionalitäten wird den PCMS-Anwendern bereits ein umfassendes und sehr wichtiges Werkzeug zur pünktlichen Erstellung der Kundenberichte an die Hand gegeben. Jedoch wird ein weiterer wichtiger Anspruch an das PCMS gestellt, der genau auf das interessierende Thema der Aggregationszustände zielt.
Zur Verifikation einzelner Kennzahlen in Kundenberichten, aber gerade auch bereits in vorherigen Stadien des Prozesses werden zwei weitere wichtige Bestandteile des PCMS gefordert.

1. Summenprotokolle (Summaries)

Am vorteilhaftesten nach jedem beliebigen Prozessschritt, aber mindestens nach bedeutenden Prozessabschnitten, werden Zusammenfassungen der Verkaufszahlen in Form von Aufaddierung der Werte innerhalb eines Jobs gefordert. Damit können Aussagen über die Richtigkeit und Vollständigkeit des Dateninhaltes getroffen werden. Dies stellt eine weitere wichtige TQM Maßnahme dar, mit der frühzeitig bei Fehlentwicklungen ein Eingriff in den Prozess stattfinden kann (z.B. auch durch Stoppen und völlig neu Starten eines kompletten Jobs). Zusätzlich wird eine historische Speicherung der Summenprotokolle vorgenommen, um Vergleichswerte für sich periodisch wiederholende Jobs zu erhalten.

2. Einzelsatz-Tracking

Zwei elementare Ziele sollen mit dem sogenannten *Einzelsatz-Tracking* verfolgt werden:

a) Nachverfolgung einzelner Händler-Artikel bis zum Kundenbericht.

Zur Evaluierung und Verifikation der gefundenen Zahlen in den Kundenberichten (bzw. in jedem vorherigen Prozessschritt) wird eine Methode benötigt, mit der einzelne Artikel vom Beginn des Prozesses bis zum Ende genau nachverfolgt und überwacht werden können (siehe Abbildung 4).

Auch in Fehlersituationen soll die Nachverfolgung eindeutige Aussagen bereitstellen können, *ob* und in *welcher Weise* ein Artikel an einer Komponente verarbeitet wurde und in welche Datenströme er nach der Komponente mit eingeflossen ist.

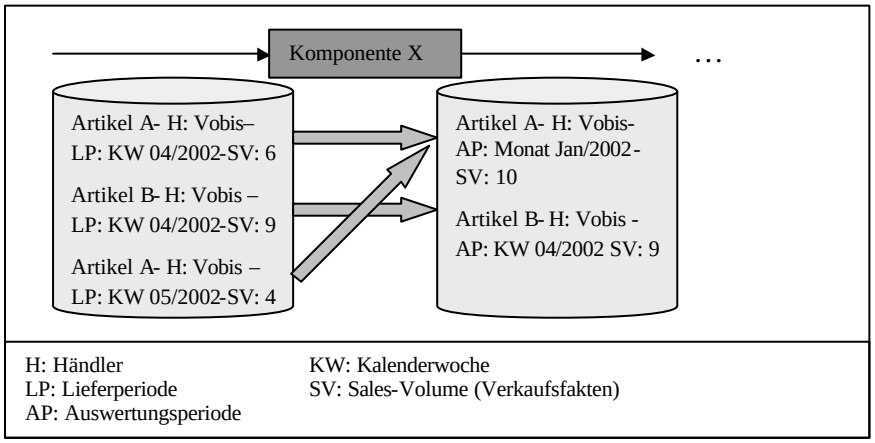

Abbildung 4: Beispiel zur Nachverfolgung der Aggregation von Lieferperioden zu Auswertungsperioden an einer Komponente

b) Simulation von geplanten Produktionszyklen

Weniger brisant, aber sicherlich dennoch ein möglicher Bestandteil des PCMS, könnte eine Simulation konkreter Datensätze durch das Datenproduktionssystem darstellen. Simuliert würde dabei der gesamte Prozessablauf, jedoch ohne das die Ergebnisse der Simulation die tatsächliche Produktion beeinflussen. Die Komponenten führen wie bei der echten Produktion die einzelnen Bearbeitungsschritte durch (Simulationsdurchlauf mit möglichst kleinen Datenmengen und geringer Priorität, damit die echte Produktion zeitlich nicht zu sehr beeinträchtigt wird). Auf Basis der ohnehin benötigten konkreten Systemkennzahlen (z.B. maximale Durchlaufzeit von Jobs an einer Komponente) und der erstellten Erfahrungswerte (z.B. Summenprotokolle) könnte eine Simulation ganzer Jobs und vielleicht auch einzelner Datensätze geschaffen werden, um *„Was-wäre-wenn"-Szenarios* mit in die Betrachtungen einbeziehen zu können. Beispielsweise ETL-Werkzeuge (siehe [BG01]) bieten derartige Funktionalitäten.

4. Strategien zur Nachvollziehbarkeit von Aggregationszuständen

Tubing System

Da, wie in Kapitel 2 bereits vorgestellt, ein relativ statischer Produktionsablauf im Datenproduktionssystem die Basis darstellt, erscheint für die Realisation des Einzelsatz-Trackings das sogenannte *„Tubing System"* sinnvoll.

Als Beispiel möchte ein Benutzer des PCMS nachvollziehen, wie bzw. ob ein bestimmter Artikel in das Endergebnis „Kundenbericht Y“ eingeht. Dafür gibt er die Identifikationsparameter für diesen bestimmten Artikel in das System ein. Dadurch wird nun der Tubing-Mechanismus ausgelöst. Nach der Reihe wird nun jede Komponente im statischen Workflow besucht und der Datenfluss für diesen Artikel nachvollzogen. Das Tubing-System funktioniert für die folgenden beiden Fälle und deckt damit ausreichend alle gewünschten Überwachungssituationen ab:

a) Kompletter Workflow-Durchlauf

Entwicklungen wie und ob ein bestimmter Artikel sich durch das Datenproduktionssystem seinen Weg gebahnt hat, werden rekonstruktabel. Während des Prozesses finden immer wieder Qualitätssicherungsmaßnahmen statt. Fachexperten haben erstmalig die Chance fehlerhafte Daten (z.B. vom Händler falsch geliefert etc.) und widerspruchsvolle Berechnungen (z.B. fehlerhafte Behandlung negativer Verkaufszahlen etc.) zu identifizieren. Stichprobenartig können Durchläufe zur Überwachung der Vollständigkeit von Kundenberichten gestartet werden. Durch das Tubing System können die Ergebnisse der Nachverfolgung an jeder einzelnen „Station“ (Komponente) angezeigt werden und somit ein übersichtliches Gesamtbild der Produktionskette im Bezug auf den gesuchten Artikel dargestellt werden.

b) Erkennen von Fehlersituationen durch Workflow-Testdurchlauf

An jeder Komponente werden Fehler geloggt. Dabei können zwei Fehlersituationen entstehen. Die erste Möglichkeit ist, eine Komponente loggt einen Fehler, kann aber trotzdem weiterarbeiten. Die zweite Situation entsteht, wenn der aufgetretene Fehler so schwerwiegend ist, dass die Komponente ihre Bearbeitung abbrechen muss. In beiden Fällen kann eine Überprüfung einzelner Artikel notwendig werden. Durch das Tubing System kann genau nachverfolgt werden, ob und wieweit ein bestimmter Artikel noch richtig verarbeitet wurde. Dabei wird wie bei der vollständigen Nachverfolgung der Artikel vom Eingang des Systems bis zu der zu überprüfenden Komponente bzw. einer Nachfolgerkomponente durchgeschleust. Aus Performanzgründen ist es vorteilhaft, dass nicht der komplette Workflow durchsucht werden muss.

Eigenschaften der Überwachung

Das Nachvollziehen der Aggregationszustände muss an jeder Komponente einzeln angebracht werden. Dies hängt damit zusammen, dass jede Komponente andere Aggregationsvorschriften besitzt und außerdem die Identifikationsschlüssel am Eingang einer Komponente sich zu denen am Ausgang einer Komponente wie bereits in Kapitel 2 erwähnt unterscheiden können.
Die Möglichkeiten zur Überwachung von Aggregationszuständen werden nun auf die folgenden Eigenschaften hin überprüft:

Eigenschaft	Ausprägung	Beschreibung
Datenmenge	Statisch / nicht statisch	Eine statische Datenmenge bleibt zu jeder Zeit gleich. Eine weitere Ausprägung wäre, Daten dürfen hinzukommen, aber nicht weggenommen werden. Ist die Datenmenge nicht statisch, darf sie sich jederzeit beliebig entwickeln.
Aggregationsvorschrift	bekannt / nicht bekannt	Die Aggregationsvorschrift beschreibt, wie die eingehenden Datenmengen aggregiert werden. Ist sie bekannt, kann sie zur Nachverfolgung herangezogen werden.
Job-Parameter	bekannt / nicht bekannt	Die Job-Parameter beschreiben, welche Datenmenge zu untersuchen ist. Job-Parameter wären in unserem Beispiel: Shop (entspr. z.B. Händler-Filiale), Produktgruppe, Lieferperiode, etc.
Speicheraufwand	minimal bis maximal	Untersuchung, welche der Überwachungsmöglichkeiten mit wie viel zusätzlichem Speicheraufwand auskommen.

Möglichkeiten zur Nachvollziehbarkeit der Aggregationszustände

Der Leser sei an dieser Stelle darauf hingewiesen, dass sich das PCMS selbst noch in der Planungsphase befindet und aufgrund dessen ebenfalls Summenprotokolle, Einzelsatz-Tracking und Tubing-System bisher nicht implementiert sind. Die im folgenden vorgestellten Ansätze basieren deshalb auf rein theoretischen Untersuchungen und wurden noch nicht durch reale Erfahrungen bestätigt.

1. ***statische Datenmengen***

   Wenn historisch gesehen zu jeder Zeit klar ist, wann welche Datenmengen zu betrachten sind, ist diese Methode die best geeignete, um Aggregationszustände vor und nach einer Komponente genau bestimmen zu können. Eine weitere Voraussetzung ist deshalb die Kenntnis über die Job-Parameter. Das Wissen über die Aggregationsvorschrift ist nicht notwendig.

   Beispiel:
   In dem kleinen Beispiel von Abbildung 5 ist die Komponente Cx dafür zuständig, aus vier Lieferperioden eine Auswertungsperiode für einen Artikel eines Händlers zu generieren.
   Angenommen Pool A ist immer pro Tag gültig. Erst nachts, wenn kein Anwender Aktionen ausführt, wird Pool A mit neuem Datenmaterial bestückt. Dann ist pro Tag Pool B immer durch die Job-Parameter reproduzierbar.

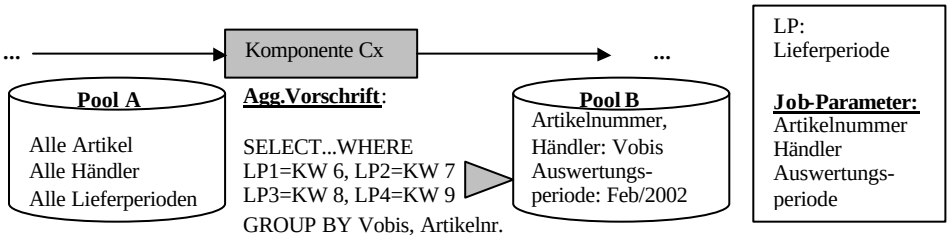

Abbildung 5: Beispiel: statische Datenmengen

Vorteil:

Es fällt kein zusätzlicher Speicheraufwand zur Überwachung an. Eine genaue Aussage kann zu jeder Zeit getroffen werden. Können dazu noch statische Datenmengen historisch gespeichert werden, wird eine schrittweise Nachverfolgung aufgrund der historisch gehaltenen Daten möglich.

Nachteil:

Für die Speicherung der historischen Datenmengen fällt jedoch meist ein zu großer zusätzlicher Speicheraufwand an. Diese Methode ist also nur bei kleinen (historisch gehaltenen) statischen Datenmengen von Bedeutung. Alle Job-Parameter müssen ebenfalls zusätzlich gehalten werden, um eindeutig die zu überwachende Datenmenge zu identifizieren. Außerdem ist bei zunehmenden Speichermengen mit einer Verlangsamung der Abläufe im zu überwachenden wie auch im überwachenden System zu rechnen, die durch den zusätzlich entstehenden Verwaltungsaufwand entsteht.

2. ***Einzelsatz-Logging***

Wird jeder eingehende Datensatz von der Komponente komplett geloggt, wären Eingang und Ausgang der Datenmengen zu jederzeit bekannt. Bei Kenntnis der Job-Parameter können die Aggregationszustände jederzeit nachvollzogen werden.

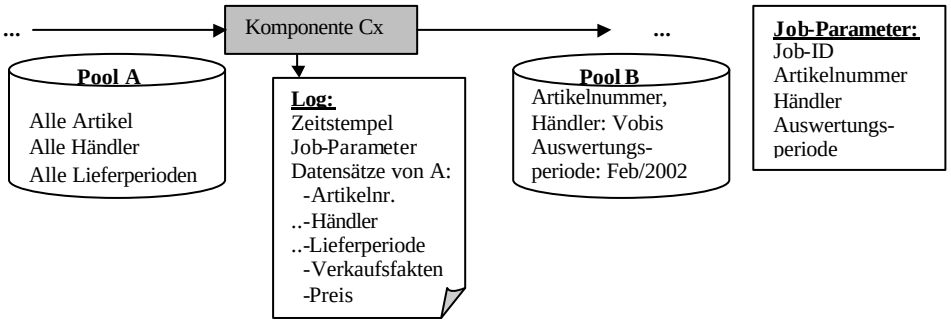

Abbildung 6: Beispiel Einzelsatz-Logging

Beispiel:

Abbildung 6 zeigt, dass alle ausgewählten Datensätze aus Pool A in ein Log

geschrieben werden. Pool B kann mit Hilfe der Job-Parameter jederzeit abgebildet werden, da die Ursprungsmenge A bekannt ist.

Vorteil:
Die Aggregationsvorschrift wird bei dieser Methode nicht benötigt, da Ein- und Ausgang der Datenmengen jederzeit zur Verfügung stehen. Auch die Datenmengen müssen nicht statisch vorgehalten werden.

Nachteil:
Es wird mindestens der doppelte Speicheraufwand für die Datenmengen benötigt. Auch hier ist deshalb eine Verlangsamung der Systeme aufgrund zusätzlichem Verwaltungs- und Speicheraufwand zu erwarten.

3. ***Primärschlüssel-Logging***

Nur die wichtigsten Primärschlüsselattribute der eingehenden Datensätze werden bei dieser Methode geloggt. Auch bei dieser Methode müssen die Job-Parameter bekannt sein.

Beispiel:
Wie bei Einzelsatz-Logging (Abbildung 6) wird geloggt. Jedoch anstatt des ganzen Datensatzes aus Pool A, werden nur dessen Primärschlüsselattribute aufgehoben (in unserem Beispiel: Artikelnummer, Händler, Lieferperiode). Ändern sich im Pool A nur die Verkaufsfakten oder der Preis eines Artikels, kann trotzdem Pool B nachvollzogen werden.

Vorteil:
Die zu speichernde Datenmenge wird im Gegensatz zu Methode 2 zumindest etwas reduziert. Beispielsweise würde in einem Einsatzfall in der GfK die Reduzierung in etwa 1/5 betragen (Verhältnis Primärschlüssel zu Verkaufsfakten).
Aggregationsvorschriften müssen nicht bekannt sein, da die Kombination aus eingehenden und ausgehenden Primärschlüsselattributen vorgehalten wird.

Nachteil:
Der größte Nachteil besteht darin, dass sich die vorhandenen Daten zu den Primärschlüsseln im Sinne von Löschungen nicht ändern dürfen. Neue Attributswerte des selben Datensatzes können hinzukommen. Jedoch dürfen keine Datensätze entfallen, da sonst ein Nachvollziehen der Ergebniswerte nicht mehr folgerichtig stattfinden könnte. Beste Voraussetzungen würden auch hier eher statische Datenmengen bieten, da sich bei diesen die vorhandenen Datensätze nicht ändern.

4. ***Datenschätzung***

Kein zusätzlicher Aufwand zur Nachverfolgung der Artikel ist bei dieser Methode erforderlich. Es werden die aktuellen Datenmengen zum Nachverfolgungszeitpunkt beim Einzelsatz-Tracking verwendet, um eine ungefähre Darstellung des Datenzustandes von dem eigentlich gewünschten echten Abarbeitungszeitpunkt zu erhalten. Das Ergebnis der Nachverfolgung an einer Komponente kann das gleiche sein, wie zum Zeitpunkt der tatsächlichen Abarbeitung. Dieser Umstand ist jedoch nicht zwingend. Ebenso kann sich die zugrunde liegende Datenmenge bereits verändert haben und damit die Nachverfolgung ein anderes (unerwünschtes) Ergebnis liefern.

Beispiel:
Ziel ist, wie in Abbildung 7 gezeigt, Pool B1 eventuell mit Verwendung von Pool A2, Job-Parametern und Aggregationsvorschrift nachzuvollziehen.
Jedoch kann sich das Abfrageergebnis (Pool B2) von Pool B1 unterscheiden, da die Abfragezeitpunkte von B1 (tatsächlicher Abarbeitungszeitpunkt) und B2 (Nachverfolgungszeitpunkt) unterschiedlich sind.

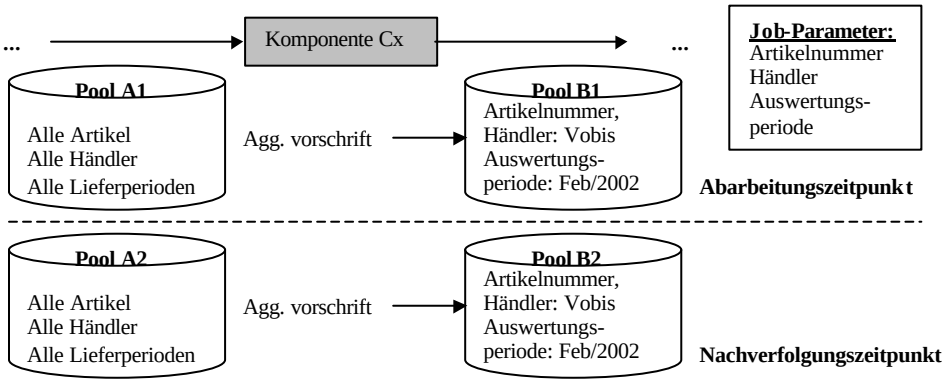

Abbildung 7: Beispiel Datenschätzung

Vorteil:
Es findet kein zusätzliches Logging statt. Deshalb wird kein zusätzlicher Speicherplatz benötigt. Auch die Datenmengen dürfen sich beliebig verändern und müssen nicht statisch gehalten werden.

Nachteil:
Die Aggregationsvorschrift und Job-Parameter müssen bekannt sein. Ersteres erzwingt einen weiteren Aufwand für jede Komponente zur Programmierung der Überwachung, um die unterschiedlichen Aggregationen nachvollziehen zu können. Der wohl größte Nachteil dieser Methode liegt darin, dass nur eine ungenaue Schätzung ermöglicht wird. Die augenblickliche Abbildung der Datenmengen kann jederzeit nachvollzogen werden, jedoch muss dies nicht dem Szenario entsprechen, mit dem z.B. ein Kundenbericht angefertigt wurde. Zusätzlich muss die zur Nachvollziehung benötigte erneute Ausführungszeit der Komponente in Laufzeitbetrachtungen berücksichtigt werden.

5. Resümee

Im Hinblick auf das Beispielszenario kann das folgende Ergebnis abgeleitet werden. Für die immens datenintensive Produktion der GfK MS wird die Methode (1) der statischen Datenmengen nicht zum tragenden Einsatz kommen können. Zu überprüfen gilt noch, ob an bestimmten Schnittstellen (lokales Land zu Zentrale bzw. Konsolidierung zu Reporting) im Produktionsablauf diese Methode trotzdem eingesetzt werden kann, da sie mit geringstem Logging-Aufwand die besten Ergebnisse für die Hauptaufgabe der Nachvollziehbarkeit erzielt.

In Umgebungen mit (evtl. historischen) statischen Datenmengen würde damit eine Empfehlung einer Strategie zur Nachvollziehbarkeit klar zu Gunsten der Methode (1) ausfallen.

Eine Anwendung des Einzelsatz-Loggings (Methode 2) im Überwachungsszenario der GfK MS ist aufgrund der datenintensiven Umgebung nicht zu erwarten. Ein Einsatz würde sich nur dann in einem Szenario rechtfertigen, wenn ein mindestens verdoppelter Speicheraufwand und eine eventuelle Verlangsamung der Systeme tolerierbar sind. Datenbanksysteme mit geringer Mengenlast und viel zusätzlich freiem Speicherpotential stellen eventuell sinnvolle Einsatzumgebungen dar. Außerdem könnte Einzelsatz-Logging sich eignen, wenn eine große Differenz zwischen vorhandenen Daten (große Menge) und Daten zur Auswertung (geringe Menge) verwendet wird. Beispielsweise in Ad-hoc Studien, in denen viele Daten gesammelt aber nur selten und wenige ausgewertet werden, könnte eine Verwendung in Betracht gezogen werden.

Das Problem für den Einsatz der dritten Methode (Primärschlüssel-Logging) ist der extrem wechselhafte Zustand der Daten im Produktionssystem. Zu keiner Zeit sind die Datenmengen in der GfK MS statisch. Zusätzlich können jederzeit Löschungen stattfinden. Ein Einsatz ist damit nur in Umgebungen zu empfehlen, die wenig Manipulationslastig sind und beispielsweise auf Löschungen ganz verzichten können. Auch ist der Aufwand an zusätzlicher Informationsspeicherung vor einem Einsatz zu überdenken. Aus diesen Gründen ist auch Primärschlüssel-Logging nicht optimal in unserem Beispielszenario.

Die Datenschätzung (Methode 4) stellt im Augenblick die wohl systemverträglichste Methode dar, mit der überhaupt (und ohne großen Aufwand) ein Einzelsatz-Tracking in der GfK MS anzudenken ist. Nach der Erstellung eines Prototyps für das Einzelsatz-Logging wird sicherlich auch klar werden, ob der Nachteil der Ungenauigkeit hingenommen werden kann, oder ob trotzdem andere Methoden in Betracht gezogen werden müssen, um eine hinreichende Überwachung des Datenproduktionssystems bereitstellen zu können. Der Einsatz lohnt sich also in Systemumgebungen in denen Ungenauigkeiten im Überwachungsergebnis akzeptierbar sind. Diese Methode besticht außerdem durch den im Vergleich zu den anderen Methoden geringsten Implementierungs- und Speicheraufwand. Anzumerken ist, dass dennoch die Systemlast erhöht wird und deshalb Einsatzfelder zu bevorzugen sind, in denen Zeiten mit geringerer Last (z.B. Nachts) zu vermerken sind.

In diesem Bericht wird deutlich, dass Aggregationen und Disaggregationen selbst wichtige Datenbankbestandteile darstellen, die ausgereifte und unverzichtbare Maßnahmen für datenintensive Businessprozesse darstellen. Wie gezeigt wurde, gestaltet sich jedoch die manchmal unverzichtbare Überwachung der dabei entstehenden Aggregatszustände als schwierig bis unmöglich.
Eine *ideale Methode* wurde dafür noch nicht gefunden, da wie gezeigt am Beispielszenario der GfK MS augenblicklich nur die Methode der ungenauen Daten-schätzung in Betracht gezogen wird. Bei dieser kann es im Einzelfall passieren, das kein befriedigendes Ergebnis gefunden wird. Das bedeutet, ein früher vorliegender Sachverhalt kann nicht detailliert überwacht bzw. nachvollzogen werden. Datensätze können hinzugekommen oder weggefallen sein. Da in den meisten Szenarios keine statischen Datenmengen vorliegen werden, kann die Datenschätzung unzureichend sein.

Eine ***datenbankseitige Unterstützung*** derartiger Überwachungsvorgänge würde einen ***deutlichen Mehrwert*** in Überwachungsfunktionalitäten darstellen, induziert aber auch gleichzeitig einen sehr hohen Anspruch an Verwaltungsinitiativen und Datenhaltung innerhalb von Datenbanken selbst.

Literaturverzeichnis

[BG01] Andreas Bauer, Holger Günzel (Hrsg.), 2001, 'Data Warehouse Systeme, Architektur, Entwicklung, Anwendung', Heidelberg: dpunkt.verlag GmbH.

[GfK02] GfK Marketing Services, 2002, [Online], http://www.gfkms.com, [2002, Sep., 02].

[LH00] Claudia Linnhoff-Popien, Heinz-Gerd Hegering (Eds.), 2000, Trends in Distributed Systems: Towards a Universal Service Market, Third International IFIP/GI Working Conference, USM 2000, Preface, Sept. 2000, Lecture Notes of Computing Science 1890, Springer Verlag, Heidelberg.

[Li01] David S. Linthicum, 2001, 'B2B Application Integration, e-Business-Enable Your Enterprise', Addison-Wesley.

[LS02] Wolfgang Lehner, Anja Schanzenberger, 2002, 'Einsatz von WebServices in datenintensiven Umgebungen', 1. Workshop von Anwendungen auf der Basis der XML Web-Service Technologie, Gesellschaft für Informatik, Arbeitskreis „Modellierung und Spezifikation von Web-Service basierten Anwendungen", FG 2.5.2.

[MG01] Peter Mertens, J. Griese: 'Integrierte Informationsverarbeitung 2'. Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler, 2001

[Pa00] Ted Pattison, 2000, Programming Distributed Applications with COM+ and Visual Basic 6.0, 2nd edn, Microsoft Press, Redmond, Washingtion 98052-6399

[STL02] Anja Schanzenberger, Colin Tully, Dave R. Lawrence, 2002, 'A web service based approach to monitor and control a distributed component execution environment', Working Paper University of Middlesex, London, Available per email: anja.schanzenberger@gfk.de, GfK Marketing Services, Nordwestring 101, 90319 Nürnberg, Germany