

ETL-Tools

Einsatz von ETL-Tools im e-Commerce Controlling

Günter Döge

Leiter Data Warehouse

YLine e-Solutions Germany GmbH



Inhalt

- Die Eierlegende Wollmilchsau - Warum ETL-Tools?
- Aufgabenstellung: 24 DWH-Systeme
- Anforderung: Synergien nutzen
- Anforderung: Problembehandlung
- ETL für alle(s)? Beispiel Clickstream-Analyse
- Und, funktioniert`s?

Die Eierlegende Wollmilchsau - Warum ETL-Tools?

- Folgende Überlegungen führten 1999 zu einer Entscheidung für den Einsatz eines ETL-Tools:
 - Alle Aufgaben im Bereich Extraction Transformation und Loading sollten von einem System gelöst werden
 - Datenaustausch über Datenbankgrenzen hinweg sollte möglich sein
 - Eine „Zwischenschicht“ wurde benötigt, da aus Sicherheitsgründen keine direkte Verbindung zwischen Produktiv- und DWH-Datenbanken besteht (also keine DB-Links möglich)

Die Eierlegende Wollmilchsau - Warum ETL-Tools?

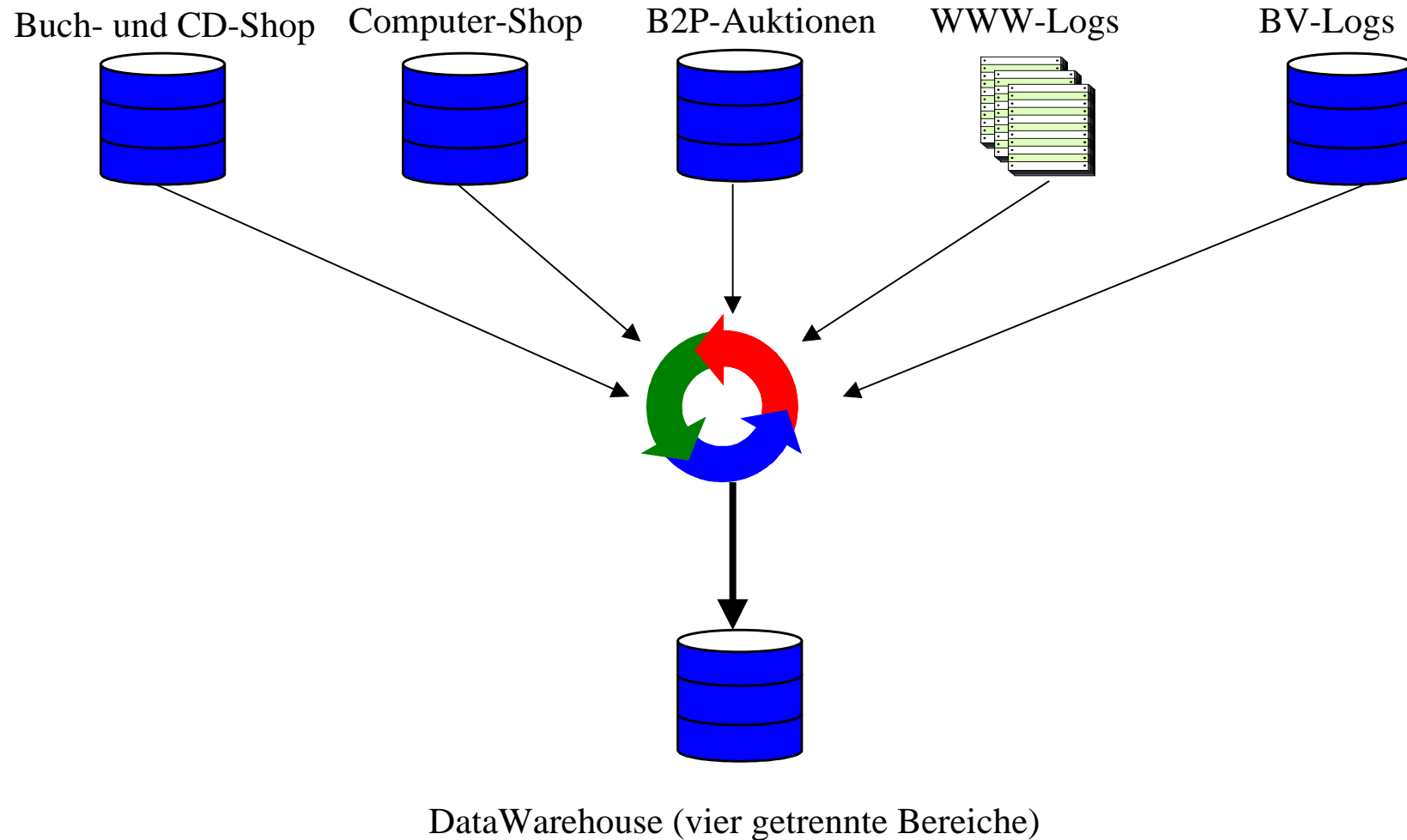
- Folgende Überlegungen führten 1999 zu einer Entscheidung für den Einsatz eines ETL-Tools:
 - Alle Schritte im ETL-Prozess sollten nachvollziehbar und gut dokumentierbar sein
 - Ein Notification- und Error-Handling System wurde benötigt
 - Alle Prozesse sollten wiederverwertbar sein

- ==> Genau das leisten ETL-Tools
(In unserem Fall: Ascential Ardent DataStage 4.0)

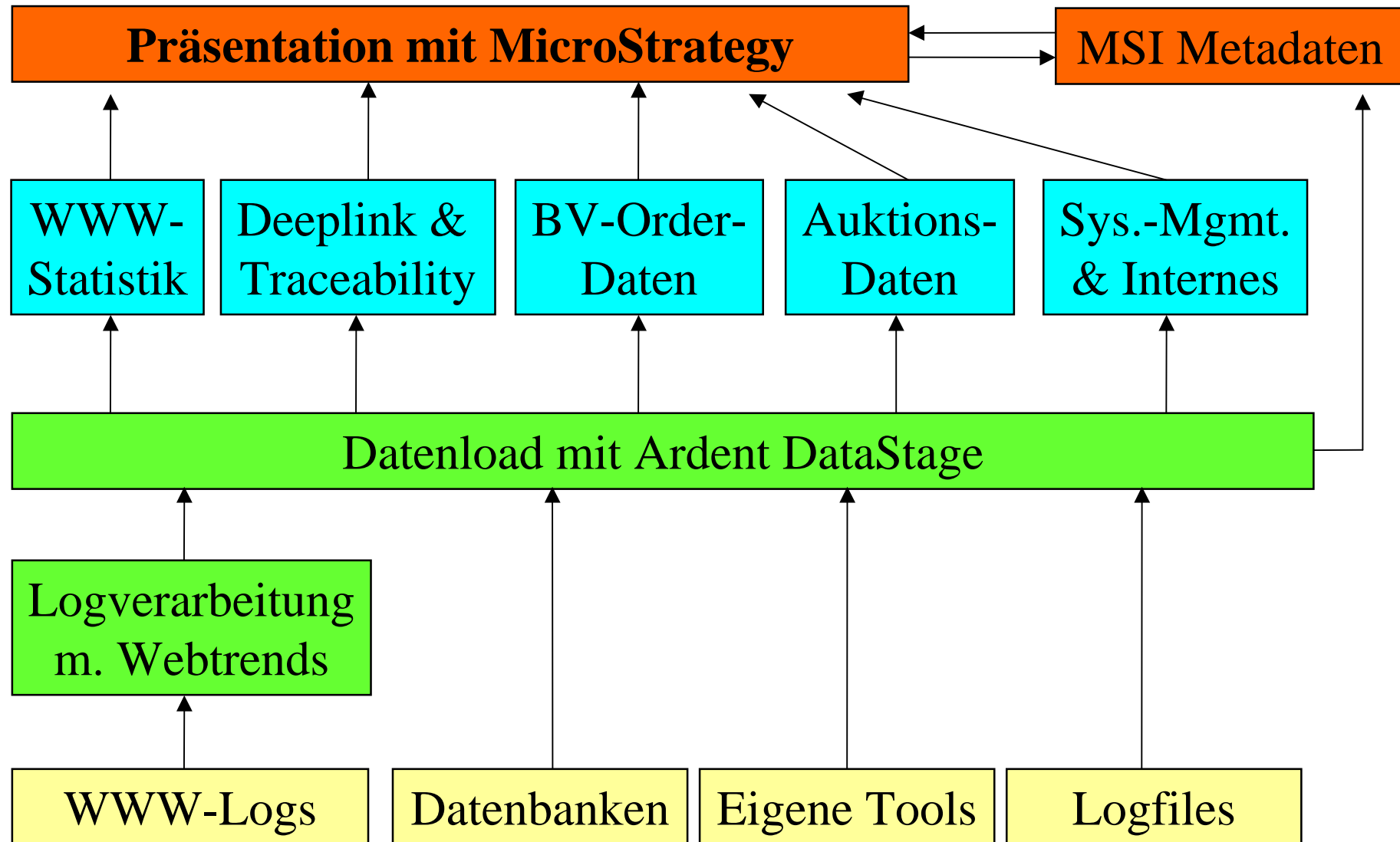
Aufgabenstellung: 24 DWH-Systeme

- Obwohl 1999 noch nicht absehbar war, daß es genau 24 Systeme werden, war klar, daß mehrere unabhängige Systeme zu implementieren waren.
- Diese Systeme nutzen im ETL-Prozess teilweise gleiche oder ähnliche Funktionen, unterscheiden sich aber in anderen Bereichen vollständig.
- Teilweise teilen sich die Produktivsysteme eine Instanz, teilweise haben Sie eigene Datenbanken.

Ausgangssituation 1999



Modulstruktur des DataWarehouse (V 1.0)



Anforderung: Synergien nutzen

- In den Bereichen „Webserverstatistik“ und „Orderdaten“ sind die Prozesse für alle Systeme (außer Auktionen) gleichartig.
- Also sollten universell einsetzbare Prozesse entwickelt werden
- Dafür ist es notwendig, daß sowohl die Connection-Informationen für die Datenbank als auch Teile des verwendeten SQL erst zur Laufzeit über Variablen zugewiesen werden (was nicht alle ETL-Tools können)

Anforderung: Synergien nutzen

- Lösung: Für jedes System existiert ein „Masterbatch“, der alle notwendigen Prozesse starten und mit Informationen (z.B. Datenbankverbindung) versorgt.
- Im Laufe der Nacht starten mehrere dieser Batchjobs und rufen so jeden einzelnen Prozeß mehrfach hintereinander auf (Parallelausführung ist zur Zeit nicht möglich)
- Auf den nächsten Seiten wird die Vorgehensweise am Beispiel der BroadVision Orderdaten gezeigt



- Jobs
 - Auktion
 - Clickstream
 - DayOffset
 - Deeplinks
 - E-Mails
 - MarketingDB
 - MarketingDBPrimus-Online
 - Orderdaten_Sonstige
 - PeopleUnited
 - Sonstige
 - Weblauscher
 - Weblauscher_Old
 - Webserverstatistik_Nachverarbeitung
 - Webserverstatistik_Produktiv
 - zDevelopment GDO
 - zNicht_mehr_benutzte_Jobs

Job name	Category	Status	Started	On date	Last
Batch::OrderLoadAuto	Orderdaten_Sonstige	Finished	01:53	07.11.01	02:0
Batch::OrderLoadEPark	Orderdaten_Sonstige	Finished	01:15	07.11.01	01:2
Batch::OrderLoadMasterbatch	Orderdaten_Sonstige	Finished	01:15	07.11.01	02:1
Batch::OrderLoadMicroshop	Orderdaten_Sonstige	Finished	02:10	07.11.01	02:1
Batch::OrderLoadOffice	Orderdaten_Sonstige	Finished	01:46	07.11.01	01:5
Batch::OrderLoadTemp	Orderdaten_Sonstige	Finished	15:37	30.03.01	15:4
Batch::OrderLoadTov	Orderdaten_Sonstige	Finished	01:22	07.11.01	01:4
Batch::OrderLoadToyCH	Orderdaten_Sonstige	Finished	00:57	05.06.01	01:0
Batch::OrderLoadTronixAT	Orderdaten_Sonstige	Finished	02:00	07.11.01	02:0
Batch::OrderLoadTronixCH	Orderdaten_Sonstige	Has been reset	01:41	10.09.01	01:4
Batch::OrderLoadWerkBox	Orderdaten_Sonstige	Finished	01:40	07.11.01	01:4
LoadBVOOrderFact01	Orderdaten_Sonstige	Finished	02:13	07.11.01	02:1
LoadBVOOrderFact02	Orderdaten_Sonstige	Finished	02:14	07.11.01	02:1
LoadBVOOrderFact03	Orderdaten_Sonstige	Finished	02:14	07.11.01	02:1
LoadBVOOrderFact04a	Orderdaten_Sonstige	Finished	02:15	07.11.01	02:1
LoadBVOOrderFact04aEPark	Orderdaten_Sonstige	Finished	01:21	07.11.01	01:2
LoadBVOOrderFact04b	Orderdaten_Sonstige	Finished	02:15	07.11.01	02:1
LoadBVOOrderFact04c	Orderdaten_Sonstige	Finished	02:15	07.11.01	02:1
LoadBVOOrderFinish01	Orderdaten_Sonstige	Finished	02:15	07.11.01	02:1
LoadBVOOrderFinish02	Orderdaten_Sonstige	Finished	02:15	07.11.01	02:1
LoadBVOOrderPreperationAuto	Orderdaten_Sonstige	Finished	01:53	07.11.01	01:5
LoadBVOOrderPreperationEPark	Orderdaten_Sonstige	Finished	01:15	07.11.01	01:1
LoadBVOOrderPreperationMicroShop	Orderdaten_Sonstige	Finished	02:10	07.11.01	02:1
LoadBVOOrderPreperationOffice	Orderdaten_Sonstige	Finished	01:46	07.11.01	01:4
LoadBVOOrderPreperationPSshop	Orderdaten_Sonstige	Finished	01:40	07.11.01	01:4
LoadBVOOrderPreperationToy	Orderdaten_Sonstige	Finished	01:22	07.11.01	01:2
LoadBVOOrderPreperationTronixAT	Orderdaten_Sonstige	Finished	02:00	07.11.01	02:0
LoadBVOOrderPreperationTronixCH	Orderdaten_Sonstige	Has been reset	09:21	11.09.01	09:2
LoadBVOOrderPreperationWBox	Orderdaten_Sonstige	Compiled			
LoadBVOOrderStep01	Orderdaten_Sonstige	Finished	02:10	07.11.01	02:1
LoadBVOOrderStep02	Orderdaten_Sonstige	Finished	02:10	07.11.01	02:1
LoadBVOOrderStep03	Orderdaten_Sonstige	Finished	02:10	07.11.01	02:1
LoadBVOOrderStep04	Orderdaten_Sonstige	Finished	02:10	07.11.01	02:1
LoadBVOOrderStep05	Orderdaten_Sonstige	Finished	02:11	07.11.01	02:1
LoadBVOOrderStep06	Orderdaten_Sonstige	Finished	02:11	07.11.01	02:1
LoadBVOOrderStep07	Orderdaten_Sonstige	Finished	02:11	07.11.01	02:1
LoadBVOOrderStep07ProdLine	Orderdaten_Sonstige	Finished	01:41	07.11.01	01:4
LoadBVOOrderStep08	Orderdaten_Sonstige	Finished	02:11	07.11.01	02:1
LoadBVOOrderStep09	Orderdaten_Sonstige	Finished	02:11	07.11.01	02:1
LoadBVOOrderStep10	Orderdaten_Sonstige	Finished	02:12	07.11.01	02:1
LoadBVOOrderStep11	Orderdaten_Sonstige	Finished	02:12	07.11.01	02:1
LoadBVOOrderStep12	Orderdaten_Sonstige	Finished	02:12	07.11.01	02:1
LoadBVOOrderStep13	Orderdaten_Sonstige	Finished	02:12	07.11.01	02:1
LoadBVOOrderStep14	Orderdaten_Sonstige	Finished	02:13	07.11.01	02:1

Status of jobs: 75 entries

General Parameters Job control Dependencies

Add Job

```
XSHOPDB = 'autopr'  
XSHOPUSER = 'xxxxxx'  
XSHOPPW = 'xxxxxx'  
XSTATDB = 'DWH_CGN1'  
XSTATUSER = 'xxxxxx'  
XSTATPW = 'xxxxxx'  
XDATERANGE = '365'  
XSTOREID = 104  
XFILIID = 104
```

* Setup LoadBVOrderPreparationToy run it, wait for it to finish, and test for success

```
hJob2 = DSAttachJob("LoadBVOrderPreparationToy", DSJ.ERRFATAL)  
ErrCode = DSSetParam(hJob2, "SHOP_DB", XSHOPDB)  
ErrCode = DSSetParam(hJob2, "SHOP_USER", XSHOPUSER)  
ErrCode = DSSetParam(hJob2, "SHOP_PW", XSHOPPW)  
ErrCode = DSSetParam(hJob2, "STAT_DB", XSTATDB)  
ErrCode = DSSetParam(hJob2, "STAT_USER", XSTATUSER)  
ErrCode = DSSetParam(hJob2, "STAT_PW", XSTATPW)  
ErrCode = DSSetParam(hJob2, "STORE_ID", XSTOREID)  
ErrCode = DSSetParam(hJob2, "DATUMSBEREICH", XDATERANGE)  
ErrCode = DSRunJob(hJob2, DSJ.RUNNORMAL)  
ErrCode = DSWaitForJob(hJob2)  
Status = DSGetJobInfo(hJob2, DSJ.JOBSTATUS)  
If Status = DSJS.RUNFAILED Then  
    * Fatal Error - No Return  
    Call DSLogFatal("Job Failed: LoadBVOrderPreparationToy", "JobControl")  
End
```

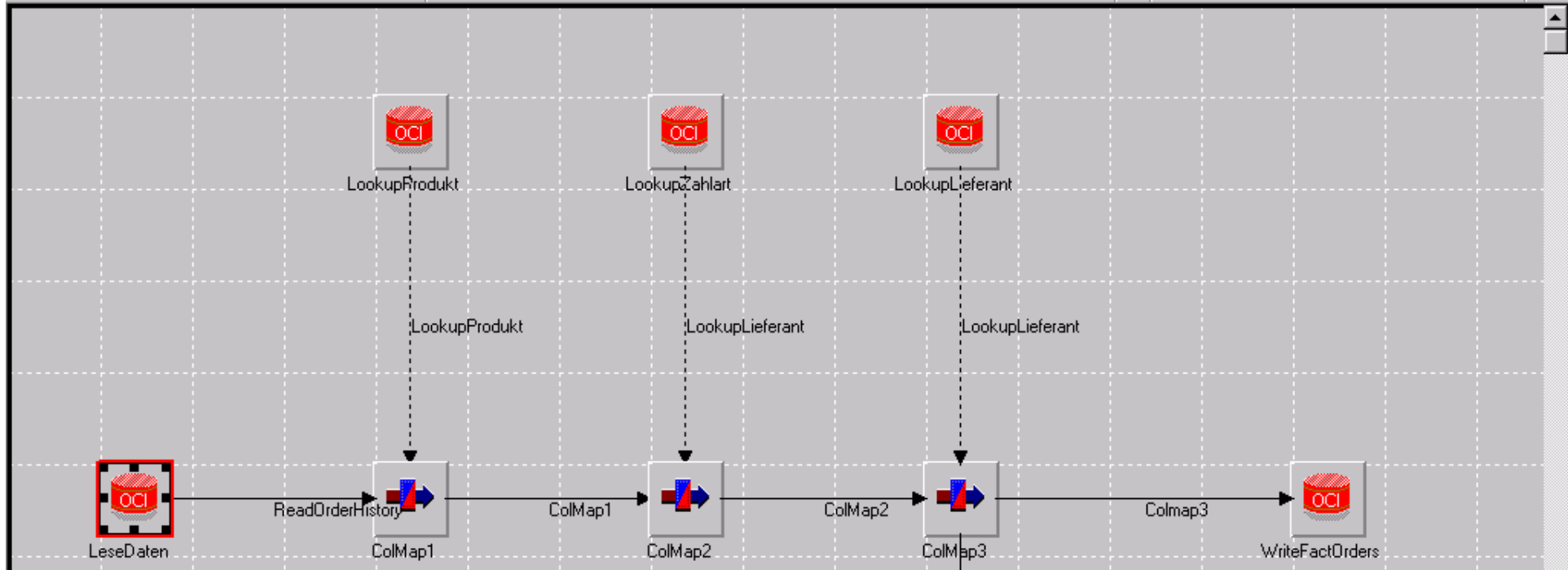
* Setup LoadBVOrderStep01, run it, wait for it to finish, and test for success

```
hJob3 = DSAttachJob("LoadBVOrderStep01", DSJ.ERRFATAL)  
ErrCode = DSSetParam(hJob3, "SHOP_DB", XSHOPDB)  
ErrCode = DSSetParam(hJob3, "SHOP_USER", XSHOPUSER)  
ErrCode = DSSetParam(hJob3, "SHOP_PW", XSHOPPW)  
ErrCode = DSSetParam(hJob3, "STAT_DB", XSTATDB)  
ErrCode = DSSetParam(hJob3, "STAT_USER", XSTATUSER)  
ErrCode = DSSetParam(hJob3, "STAT_PW", XSTATPW)  
ErrCode = DSSetParam(hJob3, "STORE_ID", XSTOREID)  
ErrCode = DSRunJob(hJob3, DSJ.RUNNORMAL)  
ErrCode = DSWaitForJob(hJob3)  
Status = DSGetJobInfo(hJob3, DSJ.JOBSTATUS)  
If Status = DSJS.RUNFAILED Then  
    * Fatal Error - Reset and Abort  
    hJobReset = DSAttachJob("LoadBVOrderStep01", DSJ.ERRFATAL)  
    ErrCode = DSRunJob(hJobReset, DSJ.RUNRESET)
```

OK

Cancel

Help



LeseDaten - ORAOCI8 Stage

Stage Output

Output name: ReadOrderHistory

Columns... View Data...

General Columns Selection SQL

Generated User-defined

```
ord.order_number||pit.item_number dcc_connector,  
nvl(pit.partial_quantity,0) f_menge_geliefert,  
nvl(pit.cancelled_supplier_quantity,0) f_menge_cancel_supplier,  
nvl(pit.cancelled_customer_quantity,0) f_menge_cancel_customer,  
nvl(pit.returned_quantity,0) f_menge_retour,  
decode(ord.fulfill_date, null, null,  
to_date(ord.fulfill_date)-to_date(ord.order_date)) f_days_to_fulfill  
from mr_orders ord,  
mr_priced_items pit  
where ord.oid = pit.oid  
and to_char(order_date,'YYYYMMDD')<to_char(sysdate,'YYYYMMDD')  
and ord.store_id = #STORE_ID#
```

Anforderung: Problembehandlung

- Eine besondere Herausforderung bei der Mehrfachverwendung von Prozessen ist die Fehlerbehandlung.
- Tritt in einem Job ein Problem auf, wird der Jobstatus auf „Aborted“ gesetzt. Jeder nachfolgende Batch, der diesen Job nutzt, bricht nun ebenfalls mit einem Fehler ab.
- Das bedeutet, daß ein Fehler bei einem Batch alle nachfolgenden Batches lahmlegen würde.

Anforderung: Problembehandlung

- Lösung: Der Batch, in dem der Fehler aufgetreten ist, wird zwar abgebrochen, aber der darunterliegende Job wird automatisch zurückgesetzt, so daß nachfolgende Batchjobs ohne Störung arbeiten können.
- Darüber hinaus wird eine Mail mit der Fehlerposition an den Systemadministrator geschickt.
- Beispiel auf der nächsten Seite: Verarbeitung von Webserver-Statistiken.

General Parameters Job control Dependencies



Add Job

```
XSTATDB = 'DWH_CGN1'  
XSTATUSER = 'xxxx'  
XSTATPW = 'xxxx'  
XFILI = 'P-O.POWSH.POWSH'  
XQUELLDIR = '/opt/ardent/webtrends/powershopping.de/'
```

* Setup Batch::WebStataMASTERBATCH2, run it, wait for it to finish, and test for success

```
hJob1 = DSAttachJob("Batch::WebStataMASTERBATCH2", DSJ.ERRFATAL)  
ErrCode = DSSetParam(hJob1, "STAT_DB", XSTATDB)  
ErrCode = DSSetParam(hJob1, "STAT_USER", XSTATUSER)  
ErrCode = DSSetParam(hJob1, "STAT_PW", XSTATPW)  
ErrCode = DSSetParam(hJob1, "FILI_ID", XFILI)  
ErrCode = DSSetParam(hJob1, "QUELLDIR", XQUELLDIR)  
ErrCode = DSRunJob(hJob1, DSJ.RUNNORMAL)  
ErrCode = DSWaitForJob(hJob1)  
Status = DSGetJobInfo(hJob1, DSJ.JOBSTATUS)  
If Status = DSJS.RUNFAILED Then  
  * Fatal Error - Send Mail and Reset  
  hJobReset = DSAttachJob("Batch::WebStataMASTERBATCH2", DSJ.ERRFATAL)  
  ErrCode = DSRunJob(hJobReset, DSJ.RUNRESET)  
  ErrCode = DSWaitForJob(hJobReset)  
  Status = DSGetJobInfo(hJobReset, DSJ.JOBSTATUS)  
  Call DSLogWarn("Job Failed: Batch::WebStataMASTERBATCH2", "JobControl")  
  Call DSExecute("UNIX", "sh /opt/ardent/projektordner/ardentfehler.sh FEHLER_IN_WEBSTAT_POWERSHOPPING", Output, Syst  
End
```

OK

Cancel

Help

ETL für alle(s)? Beispiel Clickstream-Analyse

- Anfang 2000 sollte als zweiter Schritt eine Clickstream-Analyse implementiert werden.
- Für das Jahr 2000 wurde dabei mit einem Volumen von 800 Millionen Pageimpressions, also etwa 4-5 Milliarden Logfile-Zeilen gerechnet.
- Frage: Lohnt es sich, dieses Datenvolumen per ETL in eine Datenbank zu laden und zu verarbeiten?

ETL für alle(s)? Beispiel Clickstream-Analyse

- Einige Überlegungen dazu:
 - 4-5 Milliarden Zeilen, davon 800 Millionen relevante, sind eine Menge Holz!
 - Eine sehr enge Verknüpfung mit den Benutzer- und Orderdaten aus der Datenbank war nicht unbedingt sinnvoll, da 90% der User anonym surfen und vom Rest ohnehin schon viel aus den BroadVision Logs bekannt war
 - Es blieb als Anforderung letztendlich übrig: „Welchen Weg nehmen die Benutzer auf meiner Seite?“

ETL für alle(s)? Beispiel Clickstream-Analyse

- Diese Frage läßt sich aber auch mit weniger Aufwand beantworten!
- Anmerkung: Klassische Logfile-Analyzer liefern hier meistens schlichtweg falsche Aussagen (durch Betrachtung der Referrer-URL)
- Lösung bei uns: Entwickeln eines Perl-Programms zur Logfileanalyse und Übernahme verdichteter Informationen (ca. 1/1000 der ursprünglichen Logfilegröße) per ETL in das DataWarehouse.

Und, funktioniert's?

- JA! Der Einsatz eines ETL-Tools hat sich als eine absolut richtige Entscheidung herausgestellt.
- Die Mehrfachverwendung von einzelnen Prozessen hat zu einer sehr guten Wartbarkeit geführt.
- Durch Fehlerbehandlung und -behebung zur Laufzeit gibt es keine Nachteile durch dieses Vorgehen (vielleicht mit der Ausnahme, daß es eine zeitliche Abhängigkeit zwischen den Prozessen gibt)
- Aber: Das ETL-Tool muß dieses Vorgehen auch unterstützen!

Und, funktioniert's?

- Es sind aber nicht alle Anforderungen erfüllt worden. So ist immer noch ein Teil der Arbeit auf Shell-Scripte oder PL/SQL Programme verteilt (meist weil die Funktionalität im ETL-Tool nicht vorhanden war)
- Außerdem ist sehr viel Logik in den SQL-Statements für die Datenextraktion enthalten, aber relativ wenig im eigentlichen Transformationsprozeß im ETL-Tool, obwohl die Funktionalität auch dort vorhanden gewesen wäre.

Kontakt

Weitere Informationen zum Thema
erhalten Sie beim Referenten.

Ihr Kontakt zu uns:
e-Mail: gdoege@yline-esolutions.de
Internet: www.yline-esolutions.de
oder unter:
Tel.: 0221 / 3091 - 254