Back to the roots

Vom ETL Tool zurück in die Zukunft mit PL/SQL

Edgar Kaemper (AA-AS/EIS3-EU)
Robert Bosch GmbH
Plochingen

Christoph Hisserich
Trivadis GmbH
Stuttgart

Schlüsselworte
Data Warehouse, Datenbank, Datenbewirtschaftung & ETL, DWH & Datenintegration

Einleitung

Sie müssen Ihr ETL Tool ablösen, da der Hersteller den Support für Ihre Datenbank einstellt? Sie wollen ihre ETL Tool ablösen, weil Bugs und fehlende Funktionalität die ETL Entwicklung blockieren? Sie müssen für eine Weiterentwicklung (fast) jeden ETL Prozess ändern und suchen nach Erfahrungen aus solchen Projekten?

In einem bestehenden Data Warehouse haben die Autoren des Vortrags von Full Load auf Delta Load umgestellt. Jeder ETL-Flow musste geändert werden. Das war die Gelegenheit, um das ETL Tool durch PL/SQL Prozeduren zu ersetzen. Die Motivation für das Ersetzen des ETL Tools waren fehlende Funktionalitäten im Tool und Bugs, die eine Automatisierung verhindert haben.

Welche Architektur haben wir für die neue ETL Schicht auf Basis von PL/SQL gewählt? Warum konnten 85% des Codes auf Basis von Metadaten generiert werden? Und vor allem welche Erfahrungen haben wir in diesem Projekt gemacht?

Die Autoren berichten in Ihrem Vortrag über Erfahrungen mit der Technik, der Performance, der Parallelisierung und den Besonderheiten in der XMLType und BLOB Verarbeitung. Zusätzlich werden auch Erfahrungen aus der Zusammenarbeit in internationalen SCRUM-Teams aufgezeigt.

Umfeld

Der Bosch Geschäftsbereich Automotive Aftermarket (AA) bietet Handel und Werkstätten weltweit die komplette Diagnose- und Werkstatttechnik sowie ein umfassendes Kfz- und Nfz-Ersatzteilsortiment - vom Neuteil über instandgesetzte Austauschteile bis hin zur Reparaturlösung. Das Produktportfolio von AA besteht aus Erzeugnissen der Bosch Erstausrüstung sowie aus eigenentwickelten und -gefertigten Aftermarket-spezifischen Produkten und Dienstleistungen. Über 18.000 Mitarbeiter in 150 Ländern sowie ein weltweiter Logistikverbund stellen sicher, dass mehr als 650.000 verschiedene Ersatzteile schnell und termingerecht zum Kunden kommen.

AA bietet unter der Bezeichnung "Automotive Service Solutions" Prüf- und Werkstatttechnik, Software für Diagnose, Service-Training sowie technische Informationen und Serviceleistungen.

Der Geschäftsbereich ist auch verantwortlich für die Werkstattkonzepte Bosch Service, eine der größten unabhängigen Werkstattketten weltweit mit rund 16.500 Betrieben, und AutoCrew mit über 800 Betrieben.

Die wachsende Anzahl und die steigende Komplexität der im Fahrzeug installierten Systeme und Komponenten bedeutet, dass Service-Werkstätten einen Zugang zu breitem Wissen haben müssen. Informationssysteme in der Werkstatt (z.B. ESI[tronic]) müssen praktisch jedes Fahrzeugmodell erkennen und umfassende Informationen für die Werkstätten liefern.

Ausgangslage

Mit dem Fahrzeug-Diagnose und Werkstattinformationssystem ESI[tronic] werden für Werkstätten u.A. folgende Informationen und Funktionen bereit gestellt:

- Steuergeräte-Diagnose mit neuesten Daten für Pkw-, Transporter- und Lkw-Systeme
- Fehlersuche mit geführten Suchanleitungen
- Daten für Inspektion und Service
- Komfortschaltpläne, um Fehler im System schnell lokalisieren zu können
- Schnellzugang zu bekannten Fehlern mit den Technischen Service Informationen

Für die dargestellte Datenarchitektur ist von Bedeutung, dass neben strukturierten Daten auch Dokumente (z.B. Fehlersuchanleitungen, Ein- und Ausbaubeschreibungen, …) und Medieninhalte (z.B. interaktive Schaltpläne, Bilder zur Einbaulage von Fahrzeugkomponenten, ...) von großer Bedeutung sind und große Teile des Datenvolumens auf diese Daten entfallen.

Architektur des CDW

Ziel der Architektur des Central Diagnostic Warehouses (CDW) bei Bosch ist es, alle diagnoserelevanten Daten in einer Datenbank und in einem Datenmodell zu konsolidieren und online und offline Applikationen zur Verfügung zu stellen. Neben den Applikationen sollen die Daten auch für Online Services, Datenexporte und Reporting/Analysen verwendet werden können. Die CDW Architektur ist in die für DWH typischen Layer Staging, Cleansing und Core DWH aufgeteilt. Besonderheiten sind:

- **Internal Loadstore**: Eine Archivierung der Quelldatenlieferungen als Snapshots, jede Anlieferung ist in der Regel ein Full Load und wird komplett zu Analysezwecken im Internal Loadstore gesichert.

- **CDW release**: Die Bildung der Datenhistorie erfolgt nicht beim Übergang von Cleansing in den CDW core Bereich sondern innerhalb des Core nach einer Freigabe der Daten.

- **Feedback**: Aus den Anwendungen bei den Kunden kommen Daten über die Nutzung und Fehlermeldungen zurück.

- **ODS**: Im Operational Datastore werden Daten gespeichert, die auf Grund Ihrer Beschaffenheit (noch) nicht korrekt in das CDW Datenmodell integriert werden können, für einzelne Analysen jedoch Zusatznutzen liefern.

- **Document Generator:** Das CDW enthält neben den relationalen Daten auch Dokumente (z.B. Fehlersuchanleitungen), die aus den Quellinformationen in mehreren Sprachen generiert werden.

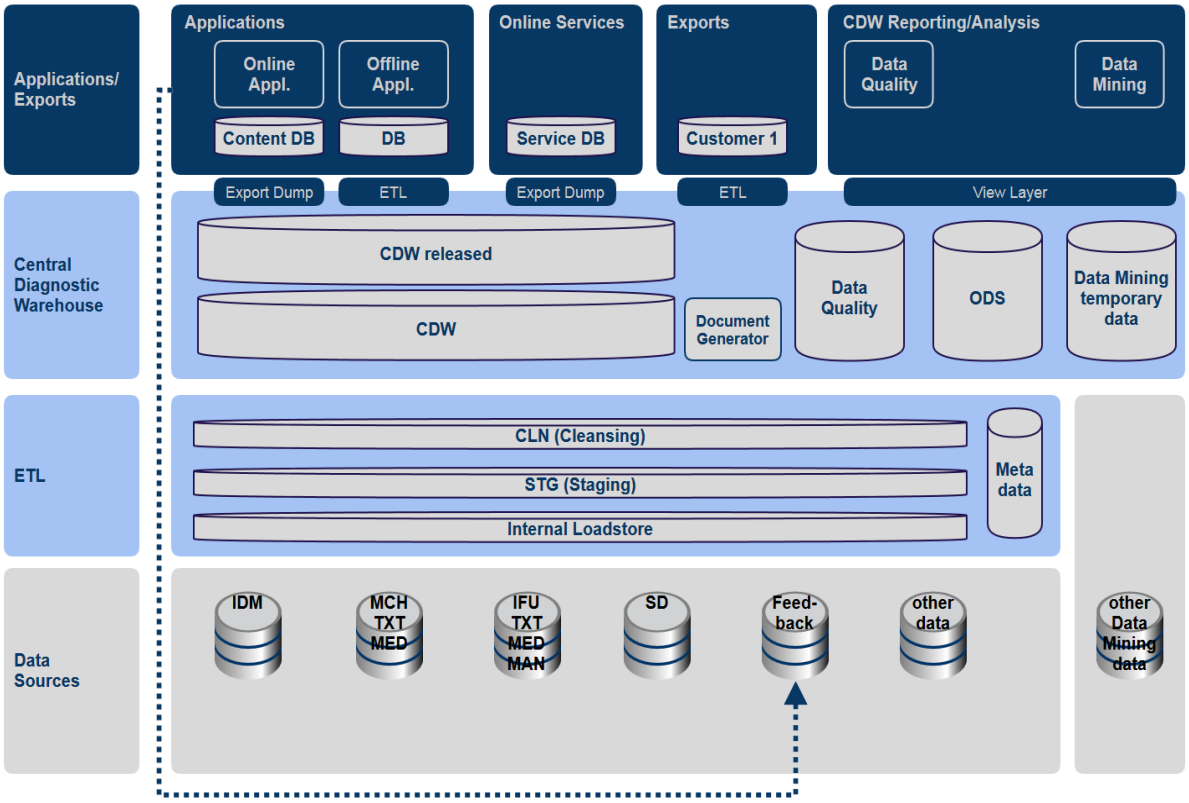

Abb. 1: Central Diagnostic Warehouse (CDW) Data Architecture

ETL Struktur

Die bestehenden ETL Prozesse wurden aus einem Vorprojekt übernommen, dessen Anforderungen ähnlich, aber zum Teil umfangreicher waren. In diesem Vorprojekt wurde für die ETL Schicht ein Tool ausgewählt, eingeführt und die ETL Schicht damit aufgebaut. Damit haben wir nicht auf einer bereits existierenden ETL Struktur aufgesetzt, eine gewachsene Plattform mit Funktionalitäten, die zum Teil nicht mehr benötigt wurden oder unbrauchbar waren (Umstrukturierung im Quellsystem etc.).

Die gesamte ETL Plattform war auf Full Load ausgerichtet, d.h. bei jedem Ladelauf wurde das Core zunächst komplett gelöscht und dann neu beladen. Dieser Ansatz entstand vor dem Hintergrund, dass nur alle 4 Monate neue Daten geliefert werden mussten.

Des Weiteren wurden in manchen Fällen die Layers (Staging, Cleansing, Core) unregelmäßig verwendet, z.B. kamen in Prozessen, die das Core befüllen, Lookups zum Einsatz, die auf Staging Tabellen zugriffen.

Durch die neue Anforderung in Zukunft häufiger Daten bereitstellen zu können, wurde eine Umstellung auf Delta Load notwendig.

Dies erforderte eine Anpassung aller bestehenden ETL Prozesse, sodass es der geeignete Zeitpunkt war, den Einsatz des jetzigen ETL Tools zu überdenken.

Das ETL Tool hatte einige Schwächen in essentiellen Teilen der ETL-Strecke:

- Verarbeitung von BLOB / XMLType Daten ist nur bei direkter Kopie (Quelle zu Ziel) möglich, sobald Schritte dazwischen erforderlich sind (Dekomprimieren von ZIP BLOBs, Änderungen an XMLType Inhalten), musste auf PL/SQL Prozeduren zurückgegriffen werden, die von BODS ausgeführt wurden.

- Des Weiteren konnte das Tool nicht effizient mit der Parallelisierung der Oracle Datenbank arbeiten. Es gibt zwar die Möglichkeit die Beladung zum und vom ETL Server zu parallelisieren, in den meisten Fällen ist dies jedoch nicht so performant gewesen, wie eine parallel Operation, die direkt auf der Datenbank ausgeführt wurde (push down).

- Einer der Vorteile von ETL-Tools soll die Flexibilität der Quellsysteme sein. Jedoch hat sich gezeigt, dass die Ablösung einer Microsoft SQL Server Datenbank durch identische Strukturen in einem Oracle Datenbankschema nicht ohne großen Aufwand funktionierte. Alle Quellobjekte mussten neu importiert und in jedem ETL Prozesse manuell ersetzt werden. Die angebotene Lineage/Impact Analyse war hier auch nicht hilfreich, da die generierte Graphik höchst komplex und unlesbar war, auch wenn man sich nur Teile hat anzeigen lassen.

Diese Ungereimtheiten und die Anforderung der Umstellung auf Delta Load, haben die Entscheidung der Ablösung des ETL Tools und die Umstellung auf reines PL/SQL leicht gemacht.

Umsetzung der Migration vom ETL Tool zu PL/SQL

Da die bestehende Architektur der CDW eine solide Basis ist, wurden alle Layer beibehalten und auch das Datenmodell im Core strukturell unverändert gelassen. Zwischen den Layers wurde alles von Grund auf neu entwickelt. So kommen zwischen zwei Schichten immer eine View und eine Prozedur als Loader zum Einsatz. In die View wird die komplette Transformations- und Geschäftslogik ausgelagert, was die Prozedur auf ihre eigentliche Arbeit reduziert. Analysen der Daten werden vereinfacht, da auf die Views direkt zugegriffen werden kann, anstatt händisch den transformierenden Code aus den Prozeduren zu kopieren und eventuelle Variablen etc. zu ersetzen.

Die Nutzung der Cleansing Area wird nun konsequent durchgezogen, sodass jede Tabelle im Core eine zugehörige Cleansing Tabelle samt Ladeprozessen besitzt.

Da die allgemeine Lade-Logik für fast alle der ca. 100 Core Tabellen identisch ist, kam das Tool biGenius von Trivadis zum Einsatz. Dieses ist in der Lage Data Warehouses von Grund auf neu zu generieren, aber auch bestehende Strukturen einzulesen und zu berücksichtigen. So konnte in nur 5 PT die Trivadis Standard-Architektur an die CDW-Architektur angepasst und die Templates (für jedes Datenbankobjekt gibt es ein Template, dass für die Generierung mit den Metadaten gefüllt wird und den PL/SQL Code erzeugt) mit den entsprechenden Anforderungen modifiziert werden. Danach war bereits eine Beladung des CDW möglich, alle Geschäftslogiken, die in den ETL Prozessen implementiert wurden, fehlten aber noch. Diese wurden manuell erfasst und in die generierten Objekte ein gepflegt, sodass nach weiteren 35 PT die Funktionalität des mit ETL Tool beladenen CDW wiederhergestellt war. Von den 1.200 generierten ETL Objekten wurden 172 in diesem manuellen Prozess angepasst, das entspricht einer Quote von ca. 85%.

	Vorher	Nachher
Ladeverfahren:	Full Load	Delta Load
Architektur:	Gewachsene Strukturen	Einheitlich, verständlich
Laufzeit:	9-10 Stunden	2 Stunden / 30 Minuten
Versionierung:	Proprietäre Lösung des ETL Tools	Standardtool (Subversion)
Flexibilität bei Änderungen:	Neuer Ladelauf erforderlich	Einzelprozesse können nachgestartet werden

Tab. 1: Gegenüberstellung Vorher/Nacher

Fazit / Erfahrungen

Durch die Umstellung auf PL/SQL und der damit implementierten Delta Load Logik ist die Beladung des CDW nun um einiges schneller und flexibler. Ohne den ETL-Server als Mittelsmann können die Features der Datenbank direkt ausgenutzt und die Daten innerhalb der Datenbank transformiert werden (ELT).

Aufgrund der einfacher zu findenden Ressourcen und des bereits vorhandenen PL/SQL Know-hows konnte die weitere Entwicklung vereinfacht und beschleunigt werden.

Lediglich die Verarbeitung von XMLType columns erwies sich weiterhin als herausfordernd. Selbst in Oracle 11.2.0.4 gibt es noch einige Bugs und ORA-600 Fehler, wenn z.B. XMLIndex oder hinterlegte XSD-Schemata verwendet werden. Dies erforderte etwas mehr Aufwand, um eine gute Performance zu erzielen, ist aber auch um einiges schneller und schlanker als die vorherigen ETL Prozesse.

Wartung in verteilten Teams mit SCRUM als Methode

Das CDW wird von einem SCRUM Team an 2 Standorten on- und offshore entwickelt und gewartet. Ein kurzer Sprintzyklus von 2 Wochen sichert eine große Flexibilität. Zum einen kann alle 2 Wochen auf geänderte Prioritäten reagiert werden. Zum anderen können Zeitanteile für die Behebung von Defects und die Entwicklung neuer Features alle 2 Wochen flexibel an die Notwendigkeiten angepasst werden.

Herausforderung auch und gerade in dieser Flexibilität der kleinen Sprintzyklen bleibt es, „große“ Stories (Epics) auf mehrere Sprints aufzuteilen und termingerecht zu liefern. Gute Erfahrungen haben wir damit gemacht, die Story so weit in kleine Tasks aufzusplitten, dass jeder im Scrum Team eine Schätzung der Story Points vornehmen kann. Anschließend müssen die Abhängigkeiten zwischen den Tasks aufgezeigt werden, um dann die Tasks über die notwendige Anzahl der Sprints zu verteilen. Aber genau diese Aufteilung auf die Sprints schränkt dann die Flexibilität wieder ein, zumindest wenn man den Endtermin der „großen“ Story nicht gefährden will.

Tool-Unterstützung für verteilte Teams

Die Toolunterstützung ist aus unserer Sicht divergent. Etablierte Entwicklungstools wie z.B. SVN liefern die Unterstützung und Flexibilität, die verteilte Teams erfordern. Z.B. kann über Locking Mechanismen sichergestellt werden, dass Entwickler sich nicht gegenseitig ihre Ergebnisse überschreiben. Auch PL/SQL Objekte können gut in verteilten Teams entwickelt werden, da der Code für jede Tabelle, Procedure etc. unabhängig aus und eingecheckt werden kann.

Bei den Modellierungstools (z.B. für Datenmodellierung) ist diese Möglichkeit nur eingeschränkt, da in der Regel, das gesamte Modell in einer Datei gespeichert und gelockt wird und damit immer nur ein Entwickler am Modell arbeiten kann. Eine Entwicklung im Team wird damit erschwert.

Integration von neuen Mitarbeitenden

Sehr gute Erfahrungen haben wir mit der Integration von neuen Mitarbeitenden in das Scrum Team gemacht. Hier hat sich über mehrere Scrum Zyklen eine starke Selbstorganisation im Team entwickelt, die eine qualifizierte Einarbeitung in die SCRUM Methode und die CDW Architektur sicherstellt.

Interkulturelle Unterschiede

In einem Projektteam über on- und off-shore Standorte hinweg nach SCRUM zu arbeiten, hebt nicht automatisch alle interkulturellen Unterschiede auf. Hier ist nach wie vor der Faktor Mensch gefordert, sich diese Unterschiede bewusst zu machen und entsprechend in der Ausgestaltung der Zusammenarbeit zu berücksichtigen.

Kontaktadresse:
Edgar Kaemper
Robert Bosch GmbH
AA-AS/EIS3-EU
Franz-Oechsle-Strasse 4
D-73207 Plochingen

E-Mail: edgar.kaemper@de.bosch.com
Internet:
http://www.bosch.de/de/de/our_company_1/business_sectors_and_divisions_1/automotive_aftermarket_1/automotive-aftermarket.html

Christoph Hisserich
Trivadis GmbH
Industriestrasse 4
D-70565 Stuttgart

E-Mail: christoph.hisserich@trivadis.com
Internet: http://www.trivadis.com