Central Diagnostic Warehouse

–

ein Unternehmen entdeckt seine Daten wieder

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

Schlüsselworte

Architektur, Modellierung, Datenmodell, Data Warehouse, Konsolidierung

Einleitung

Der Vortrag beschreibt Vorgehensweise und Erfahrungen auf dem Weg, eine zentrale Datenbasis für alle Diagnose Anwendungen im Bereich Automotive Aftermarkets/Automotive Service Solutions bei Bosch zu schaffen. Er geht auf Architekturfragen ein und beschreibt die Erfahrungen, eine solche Architektur in einer Organisation zu etablieren. Auch ausgewählte Themen in Bezug auf die Modellierung von Produktdaten über Unternehmensgrenzen hinweg und Typisierung bzw. Identifizierung von Informationsbausteinen werden mit Herausforderungen und Lösungsansätzen dargestellt.

Umfeld

Der 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

Das Informationssystem ESI[tronic] ist ein offline System, dessen Daten lokal auf dem Client in einer transbase Datenbank abgelegt sind. Mehrere Male pro Jahr werden die Daten und die zugehörige Software mit einem Update versorgt. Für verschiedene Zielgruppen und verschiedene Hardware des Systems sind Varianten gebildet worden. Jede dieser Varianten arbeitet auch mit einer eigenen Variante der Datenbank. Die Variante in Bezug auf die Datenbank bezieht sich sowohl auf das Datenmodell als auch auf die Inhalte.

Um die Updates vorzubereiten, werden alle relevanten Daten aus den internen und externen Datenquellen von einem externen Dienstleister oder intern in einer Vorverarbeitung aufbereitet und als flat files zur Verfügung gestellt.

Die bisherige Datenarchitektur ist davon geprägt, dass Datenbanken nur temporär erzeugt werden und nach bilden des Setups für die Anwendungssoftware wieder gelöscht werden. Eine Laufzeitreduktion der Ladeprozesse ist so nur erschwert möglich, da jedes Mal die Datenbank neu aufgebaut werden muss. Die Datenbanken können so auch nicht für Reportingzwecke genutzt werden.

Die Datenmodelle in der bisherigen Datenarchitektur sind nicht integriert (z.B. gibt es quellspezifische Tabellen) und nicht generisch, d.h. neue Merkmale eines Fahrzeugs führen zu Änderungen am Modell.

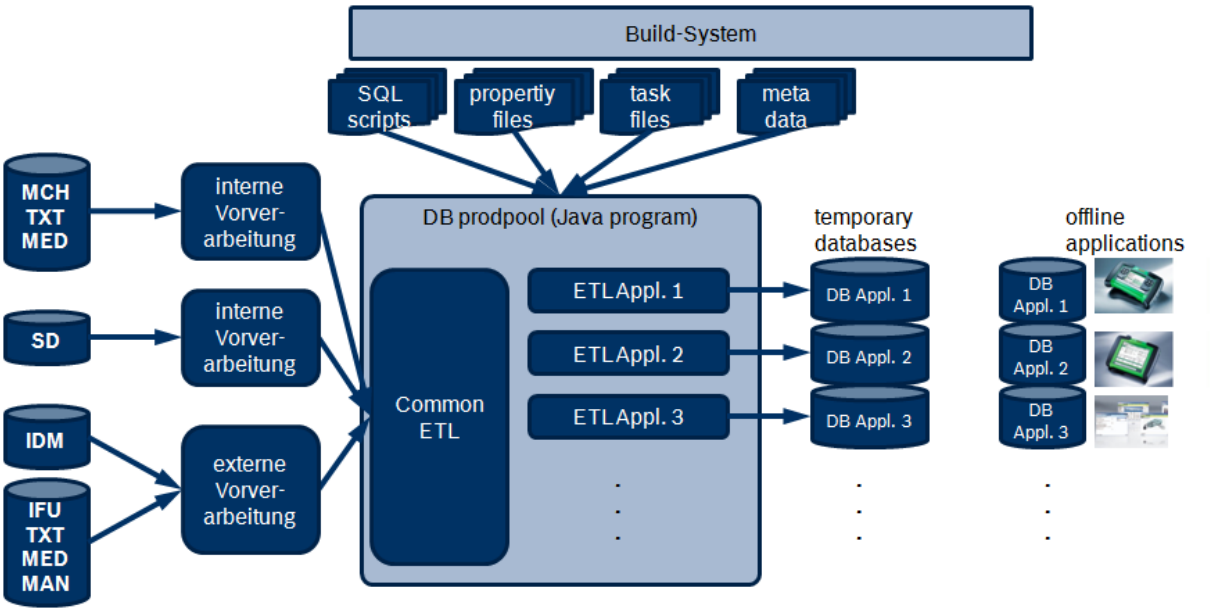

Abb. 1: IST-Architektur

Zielarchitektur

In der Zielarchitektur sind die Datenbanken persistent. Wo wird es möglich, Laufzeitreduktion (z.B. durch changed data) zu erreichen und die konsolidierten Daten auch für Analysen und Reporting zu verwenden. Die Historie der Daten wird nicht der Datenbank abgebildet und muss nicht durch den Vergleich von verschiedenen Datenbanken ermittelt werden.

Das Datenmodell ist über die Datenquellen hinweg konsolidiert. Gleichartige Informationen liegen in der gleichen Tabelle, unabhängig von ihrer Quelle. Das Datenmodell ist in weiten Teilen generisch. Dies sichert Stabilität des Modells, d.h. z.B. neue Merkmale eines Fahrzeugs führen nicht mehr zu neuen Spalten im Datenmodell

Die externe Vorverarbeitung wird sukzessive abgebaut, um Änderungen an den Inhalten und den ETL Strecken agiler umsetzen zu können.

Neue Daten sind in die Architektur integrierbar ohne den Implementierungsaufwand mehrfach leisten zu müssen.

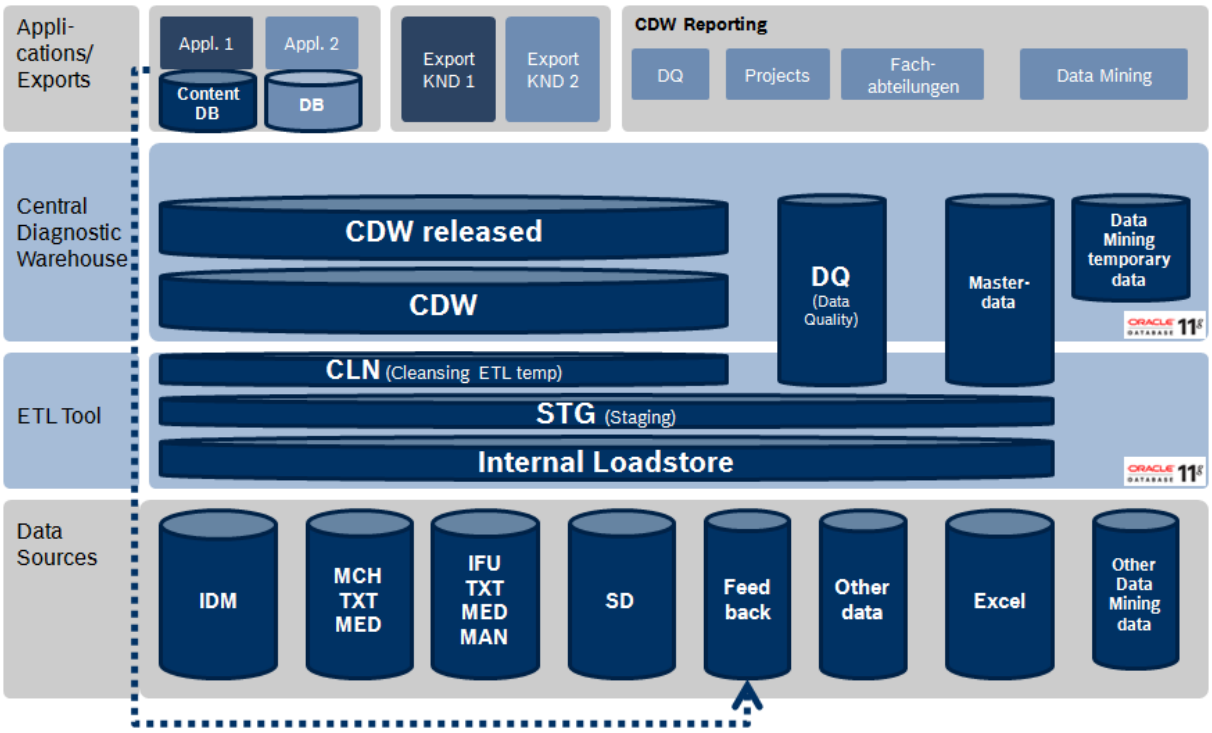

Abb. 2: Zielarchitektur

Datenmodell

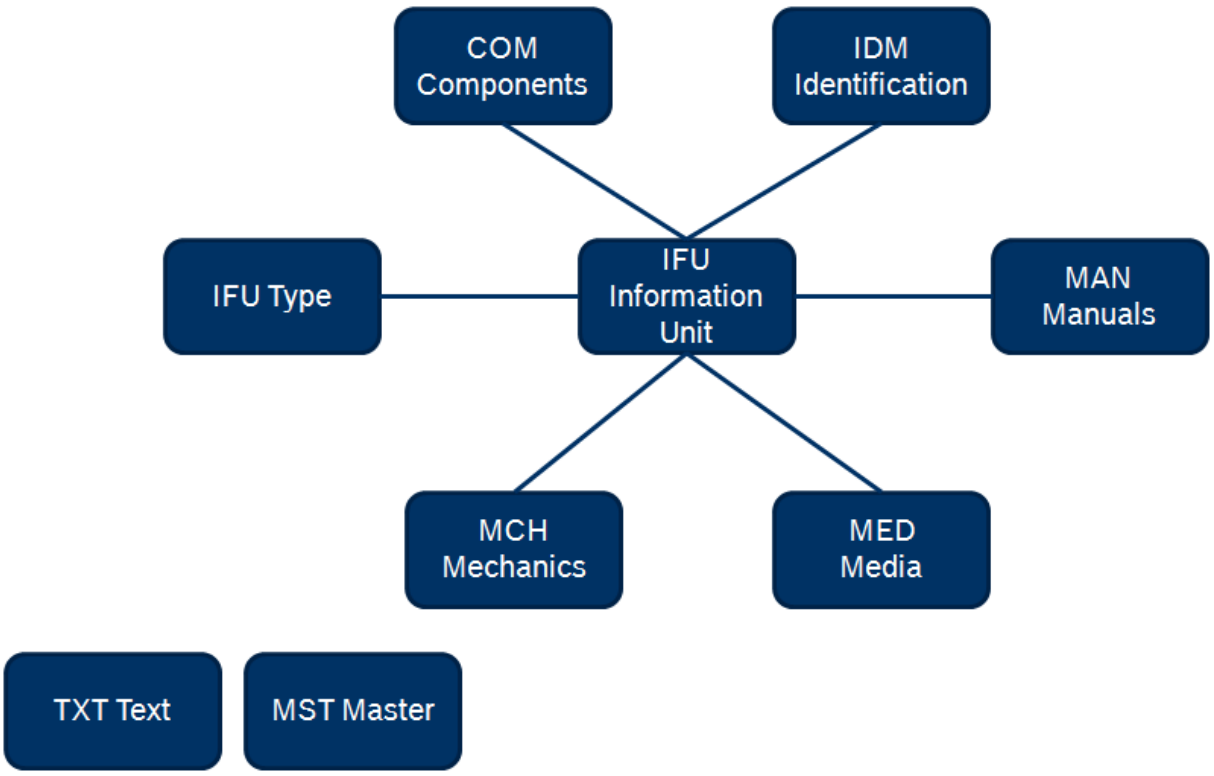

Abb. 3: CDW Data Model

Das Datenmodell stellt eines der wichtigsten Bausteine in der Zielarchitektur dar und wird hier stark vereinfacht dargestellt. Die Tabellen des physischen Datenmodells sind einzelnen Themenbereichen zugeordnet. Die wesentlichen Themenbereiche sind in Abbildung 3 dargestellt.

Das Modell umfasst ca. 130 Tabellen. Wesentliche Herausforderung ist nicht die Größe, die größte Tabelle hat „nur“ ca. 7 – 8 Mio. Datensätze. Wesentliche Herausforderung ist die Komplexität auf Grund von generischen Datenmodellansätzen.

Fast alle Texte in der Datenbank sind in mehrere Sprachen übersetzt. Deswegen hat fast jede Tabelle eine oder mehrere Beziehungen zum Themenbereich TXT Text.

Organisation

Um die neue Architektur in der Organisation zu verankern wurde eine Plattform neu eingeführt, die eine Diskussion und den Erfahrungsaustausch zur Datenarchitektur ermöglicht. Darüberhinaus wurden eine Governance und ein Change Control Board für die Datenarchitektur und die Technologien der ESI-Datenlandschaft eingerichtet.

Die neue Plattform dient auch dazu, die Wartungsaktivitäten an der ESI-Datenlandschaft abzustimmen.

Mehrere (Datenbank-)Experten agieren als think tank und entwickeln die Grundzüge der Datenarchitektur als „big picture“ weiter.

Die **Implementierung der Datenlandschaft** wird in einer Mischung von on site and off shore development mit internen und externen Ressourcen realisiert. Die Entwicklung des fachlichen und die Ableitung des technischen Datenmodells sowie die „Übersetzung“ der fachlichen Requirements in ETL und Datenbank Requirements erfolgt zentral. Die einzelnen Arbeitspakete werden dann an interne ETL Entwickler zu Implementierung weiteregegeben, teilweise auch externe ausgeschrieben.

Herausforderungen

Von Releasezyklen zu daily load

Die Organisation für die Entwicklung von Datenbanken ist auf wenige Releasezyklen (der Daten) pro Jahr ausgerichtet. Daran orientiert sich im Wesentlichen auch die Unterstützung durch die zentrale IT Abteilung mit Hard- und Software sowie Services. Die Erwartung an die neue Datenarchitektur ist, dass auch deutlich kürzere Zyklen bis hin zu einem daily laod oder neartime möglich sind.

Da in den ETL Prozess und in den Datenquellen noch keine Lieferung von changed data vorgesehen ist, muss dies in noch implementiert werden. Dazu wurden 6 Schritte identifiziert, die als einzelne Arbeitspakete ausgeschrieben werden können.

Bislang wurde die Jobsteuerung mit einem im ETL Toll selbst geschriebenen Jobframework gemacht. Dies wird durch eine Job Scheduling System ersetzt, um neben ETL Jobs auch reine DB-Jobs sowie File Transfers in einem Tool automatisiert steuern zu können.

Die Freigabeprozesse sind von der manuellen Abarbeitung von Testfällen geprägt. Bislang ist noch kein Set von Regeln definiert, die Datenqualität messbar und automatisierbar macht.

Grenzen des ETL Tools

Im Zuge der Weiterentwicklung des CDW Datenmodells und der ETL flows treten mehr und mehr die Grenzen eines ETL Tools in den Vordergrund, so dass der Einsatzzweck des ETL Tools in folgenden Fällen zu überdenken ist. Da es hier nicht darum geht, einen Toolhersteller zu „blamieren“ bleibt dieser ungenannt. Es geht vielmehr um die Weitergabe von Erfahrungen und Grenzen von ETL Tools, die evtl. anderen beim Einsatz von ETL Tools weiterhelfen.

Komplexe ETL flows: Spezielle Datenstrukturen insbesondere im Umfeld von komplexeren XML Datenquellen führen immer wieder zu der Situation, dass ein ETL flow nicht im ETL Tool entwickelt werden kann, sondern als PL/SQL Funktion/Procedure oder Package entwickelt werden muss. Das ETL dient in diesen Fällen nur noch als Wrapper, um den PL/SQL code im Rahmen der ETL flows ausführen zu können.

Auswirkungsanalysen: Diese sind prinzipiell möglich. Bei komplexen flows ist eine Filterung nicht möglich, um z.B. reine lookups nicht in der Darstellung zu haben. Damit werden zentrale ETL flows (z.B. Text Model) so unübersichtlich, dass ein Einsatz der Auswirkungsanalyse schwierig bis unmöglich wird.

Wechsel der Datenquelle: Für die Masterdaten war im Vorprojekt ein SQL Server eingesetzt worden. Diese Daten wurden auch in die CDW Oracle Datenbank überführt, um nur noch eine Datenbanktechnologie im Einsatz zu haben. Allerdings konnte nicht einfach die Datenquelle im ETL Tool von SQL Server nach Oracle unter Beibehaltung der Tabellen und Spaltennamen geändert werden. Es musste vielmehr eine neue Datenquelle angelegt und jeder ETL flow, der die bisherige SQL Server Datenquelle verwendet hat, auf die neue Datenquelle umgestellt werden.

Bugs im Tool: Analog zum Ausweichen auf PL/SQL bei komplexen Datenstrukturen müssen auch ein paar bugs imETL Tool durch manuelle Ausführung von ETL flows oder durch PL/SQL Lösungen umgangen werden.

Fehlende Funktionalität: Nicht alle Funktionen, die die Datenbank zur Verfügung stellt, sind auch im ETL Tool verfügbar und müssen teilweise mit entsprechendem Aufwand „nachprogrammiert“ werden. Ein Beispiel dafür ist die Funktion listagg.

Fazit

Architektur

In der Architektur werden die einzelnen Datensilos der Anwendungen zu einem integrierten Datenmodell konsolidiert. Der Wert der eigenen Daten wurde erkannt und der Weg von extern gemanagten Daten zu internem Know How und Handlungsfähigkeit angefangen. Es hat sich gezeigt, dass DWH Konzepte und Architekturen auch für einen Datenpool zur Versorgung von Anwendungen mit Diagnosedaten tragen.

Organisation

In der Organisation hat sich die Plattform zum Erfahrungsaustausch und die Einrichtung der Governance Funktion bewährt. Beides hilft, die neue Datenarchitektur mit Leben zu füllen.

Herausforderungen

Als Herausforderung bleiben die Umstellung der Prozesse (z.B. Test und Freigabe von Daten) und die Etablierung der neuen Prozesse. Zusätzlich sind auch im Jahr 2014 noch technologische Hürden zu meistern, weil auch bei relativ geringen Datenmengen z.B. ETL Tools an ihre Grenzen kommen und Bugs eine Automatisierung von ETL flows verhindern.

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