

# Datengetriebene Anwendungen

Hans Niedermeier, PRO-Software GmbH

(DOAG-Regionaltreffen München vom 23.3.2011)

## Einleitung

Für die theoretische Betrachtung von DAP5 möge der Leser die Website

**[www.dap5.de](http://www.dap5.de)**

besuchen.

Dort ist alles beschrieben, was DAP5 ist, wie es im Vergleich zu anderen Werkzeugen der Applikationsentwicklung einzustufen ist.

Es werden dort im Lauf der Zeit Doku und Info Dokumente zum Download eingestellt.

Dieser Vortrag bezieht sich auf einzelne Kernpunkte und zusätzlich auf praktische Erfahrungen, welche in der Entwicklung von DAP5 und danach beim Einsatz gemacht wurden.

Gliederung :

1 - Was ist datengetrieben ? - Implementierung in DAP5

2 - Das Werkzeug XDEV2 - ein kurzer Überblick - Stärken - Defizite

3 - Demobeispiel - Erweiterung einer DAP5-Applikation

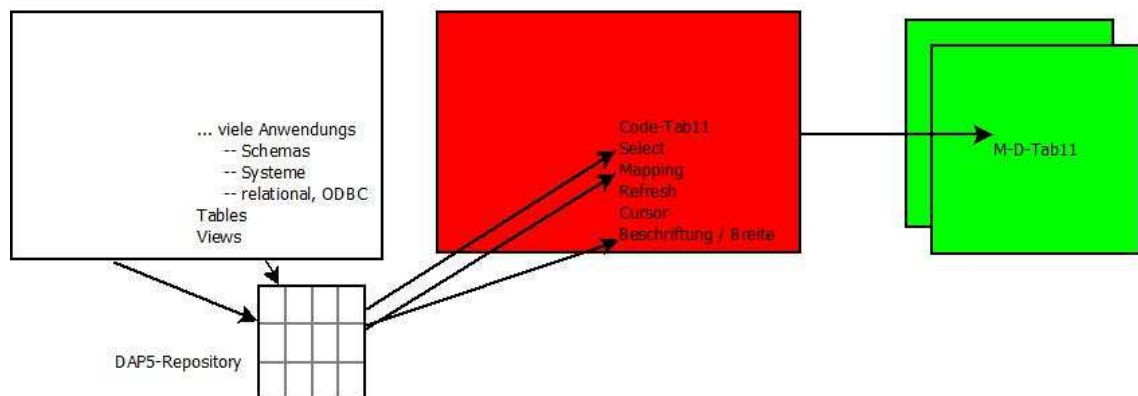
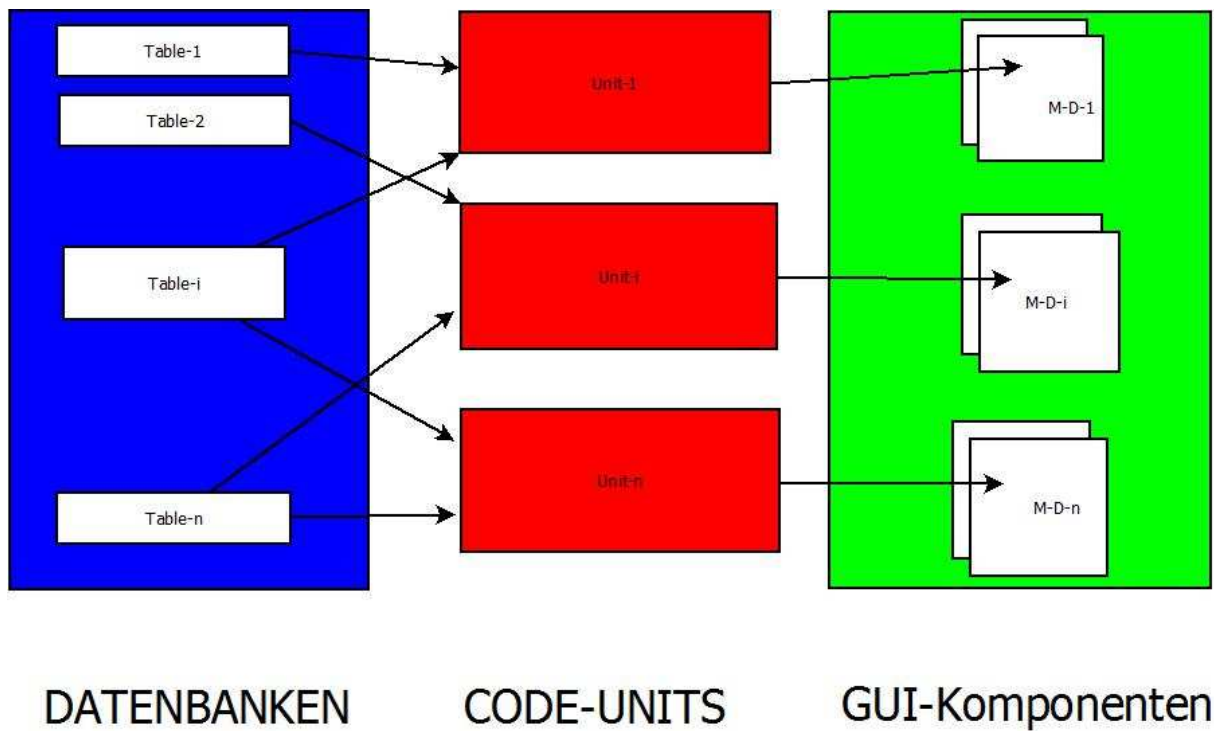
4 - Praxisbericht über den Einsatz innerhalb einer bestehenden Landschaft

# 1. Was ist datengetrieben - datenbankgetrieben ?

Am besten ist dies anhand eines Technikvergleiches zu erklären.

Der obere Teil zeigt Codestrukturen von codezentrierten Entwicklungs-Systemen

Der unter Teil stellt DAP5-Codestruktur dar.



Die Grafik zeigt den Unterschied zwischen Mainstream-Werkzeugen und DAP5.

DAP5 besitzt u.a. einen TAB, in welchem eine Master und eine Detail Table (View) untergebracht werden.

Die applikationsspezifischen Teile für diese GUI-Komponente werden erst zur Laufzeit an den Ausführungscode der Master-Detail Komponente herantransportiert

- Select-Statement
- Beschriftung
- Spaltenbreiten
- wie erfolgt die Block-Synchronisation von außen ?

Bei herkömmlicher Programmierung mit Forms, C#, C++, XDEV2 usw. erfolgt dies auf prinzipiell andere Art und Weise.

Für jeden Anwendungsfall einer Master-Detail-Darstellung wird eine eigene Code-Unit erzeugt, in welcher alles hart codiert (und auf viele Properties verteilt) enthalten ist

Alles was in DAP5 erst zur Laufzeit herangeschafft wird, fließt bei XDEV2 (als Stellvertreter aller anderen Werkzeuge) in den Entwicklungsprozess ein und generiert Klassen und Methoden, welche final compiliert werden müssen.

Dies ist ein Fakt, da alle diese Werkzeuge aus dem Datenmodell der APP

- Code ableiten
- Codeteile mehr oder weniger vom Entwickler vervollständigt werden müssen

Zur plastischen Erklärung der Unterschiede nehmen wir die berühmten

L E G O - Baukästen.

***In DAP5 sind weitgehende Baugruppen vorgefertigt, welche sofort für jedes gewünschte Gebäude zusammengesetzt werden können.***

**In XDEV2 (Forms,..) gibt es Assistenten, denen man sagt, welche LEGO-Baugruppen erst erzeugt werden sollen, um Sie dann zusammen zu setzten.**

Die Idee der datengetriebenen Anwendungen stammt aus einem Projekt Mitte der 90-er Jahre - ein frei definierbarer Anzeigemonitor war die Aufgabenstellung.

Innerhalb weniger Tage konnte man den Monitor an Kundenwünsche anpassen - Sprachdarstellungen, Berechtigungen, Anbindungen von Stored-Objects konnten flexibel ohne eine Zeile GUI-Programmierung erfolgen.

Basis der Flexibilität war ein Repository auf einem Oracle-Schema, welche alle

w e i c h e n

Teile eine Anwendung dort speicherte und verwaltete - das R e p o s i t o r y.

Im Gegensatz zum Oracle Designer-Repository war dieses stringent in viele Tabellen normalisiert.

Dieser DAP5 - Vorläufer war 100-Prozent datengetrieben !

Das damalige Repository war die theoretische Basis für das jetzt zugrunde liegende DAP5-REPOS.

Es erfuhr eine Erweiterung, um auch Eingabe-Masken mit Validierungen datengetrieben definieren zu können.

Das DAP5-REPOS ist ausschließlich in der Datenbank definiert - es gibt keine XML-Beschreibungen.

**Das meiste was Forms und Apex und auch XDEV2 können, sollte DAP5 beherrschen - jedoch mit dem Unterschied, dass man keine GUI-Objekte programmieren muss .**

Zudem hat DAP5 ein frei definierbares Tree-Control, was in Forms kaum oder nur mit hohem Aufwand darstellbar wäre.

Die Komplexität der Tree-Programmierung dürfte hinlänglich bekannt sein.

Wegen der vielfältigen Eigenschaften der Sprache Java fiel die Wahl des Werkzeuges zum Implementieren von DAP5 auf die XDEV2-Workbench.

### 3. Das Werkzeug XDEV2

Selbst aus der Oracle-Designer und Developer-Ecke kommend, wäre das Erlernen einer objektorientierten Sprache zu aufwändig gewesen.

***OR-Mapping, das Vietnam der IT (Artikel download siehe)***

***<http://www.odbms.org/download/031.01%20Neward%20The%20Vietnam%20of%20Computer%20Science%20June%202006.PDF>***

XDEV2 ist ein RAD-Tool auf Java-Basis - es erzeugt Java-Anwendungen.

Durch die XDEV2-Object-Script-Schicht und den komfortablen

Designerkomponenten ist man ähnlich wie in SQL-Forms in der Lage, sehr schnell Oberflächen zu erstellen.

Alles ohne Java Programmierung beherrschen zu müssen.

Man kann mit der Object-Script sehr schnell und einfach eigene Funktionen entwickeln - diese münden beim kompilieren (exportieren) in privaten Java-Klassen.

Mit dem integrierten Dateneditor schafft man in kurzer Zeit, aus Datenbank-Objekten Virtuelle Tabellen (VT) zu erzeugen.

Man kann virtuelle Tabellen auch mit dem XDEV2-Dateneditor frei erzeugen

Die virtuellen Tabellen kann man als die XDEV2-spezifische Implementierung von Persistenz-Klassen betrachten - **Darstellungsform ist die Tabelle.**

Mit hoch entwickelten Methoden, die im Kontext zu den VT oder zu den designten GUI-Komponenten angeboten werden und unter Anwendung einiger Events schafft man mit wenigen Zeile Code sehr schnell komplette Anwendungen.

Auch das Debugging lässt keine Wünsche offen.

An jedem Haltepunkt des Debuggings kann man die Inhalte beliebiger VT's als CSV-Datei ausgeben um dort Analysen weiterzuführen.

Nicht zuletzt die Verwendung von XDEV-Anwendungen auf vielen Plattformen und auch die Connectivity gegen sehr viele Datenbanken werden geschätzt.

## **Schwächen von XDEV2 - Ausblick auf XDEV3**

Trotz aller Begeisterung für XDEV2 sollen auch die derzeitigen Defizite genannt werden. Derzeitig deshalb, weil XDEV2 eine Weiterentwicklung erfährt und als XDEV3 in Kürze erscheinen wird.

Viele der XDEV2-Schwächen werden in XDEV3 ausgemerzt sein.

Beispiele :

- jede Operation gegen die DB erfordert intern einen Connect  
persistent\_connection true wird in XDEV3 möglich sein und funktionieren
- Oracle-BLOB's nur mit Umwegen handlebar - Lösung in XDEV3
- Transparente Verfolgung der VT-Inhalte während XDEV-Transaktionen nicht sauber
- gleichzeitiges Maus-Event und Tastatur-Event auf ein Objekt schließen sich aus
- relativ langsamer Ablauf von Funktionen (XDEV-Object-Script) entfällt in XDEV3, weil jede eigene Funktion als reines Java übersetzt wird.  
Beispielsweise ca. 5000 Sätze aus einer VT intern weiterverarbeiten und als Tree aufbauen dauert im 2-stelligen Sekundenbereich
- Das in XDEV2 verwendete Swing-Framework bietet nur relativ einfache GUI-Komponenten. Eine optional zu erwerbende [XDEV Component Suite](#) bietet weitere komfortable Erweiterungen wie z.B. TreeTable.
- Datensatz-Integrität (Locking) muss man selbst implementieren - ist mit geringem Zusatzaufwand machbar.

Weitere Informationen über XDEV3 siehe

**<http://cms.xdev-software.de/content.php?68-startseite>**

### 3 Ein DEMO Beispiel

Bevor an einem Demo-Beispiel einer Applikation gezeigt wird, wie man diese um einige Komponenten erweitert, seien grundsätzliche Eigenschaften von DAP5 gelistet :

- zum Definieren einer Anwendung erstellt man eine Script-Datei, die mit SQLPLUS in das DAP5-REPOS geladen wird
- eine Anwendung in der Script-Datei ist einfach und sequentiell lesbar - jede Änderung z.B. an Beschriftungen können mit einem einfachen Editor intuitiv vorgenommen werden
- beim Ladevorgang erfolgen Prüfungen - analog eines Compile-Vorganges
- es gibt einige System-Anwendungen (DAP5-Workbench), welche das Erstellen von Script-Teilen generisch aus dem DB-Katalog unterstützen
- das komplette Anwendungsschema einer APP-DB wird in Explorer Darstellung angeboten
- Verwendungsnachweise von DB-Objekten in DAP5-Applikationen sind direkt möglich
- Struktureller Aufbau der Applikationen in Baumstruktur erleichtert Information und Analyse

**Die gesamte DAP5-Workbench selbst ist 100 % datengetrieben definiert.**

- an S03 erklären Aufbau der DAP5-Oberfläche
- NAVT, einige Kerneigenschaften dieses Tree - Vergleich mit Forms
- TAB01 bis TAB30 überfliegen und erklären
- Synchronisation TREE -> TAB11X -> TAB12Y -> TAB13Z
- Synchronisation von Blöcken innerhalb eines TAB
- Wirkungsweise der PARAM - DAP5-Databinding
- Button-Eigenschaften - Aufruftypen
- Modale Dialoge
- ....

## **Anhand S03 Schemas und Statement-Generierung**

- zeigen der A24- Auftragsabwicklung
- erweitern Tree um einen statischen Knoten
- erweitern Tree um einen dynamischen Knoten
- definieren eines TAB M-D
- definieren eines Buttons
- definieren eines weiteren TAB (TAB25 Pivot)
- zeigen der TAB-Synchronisation
- zeigen Testphase - rechte Maustaste - interne Bereiche

## **Anhand A24 Aussehen und Funktionalität**

- zeigen der A24- Auftragsabwicklung
- zeigen Statistik und Analyse mit Pivot-Tab
- erweitern Tree um einen statischen Knoten
- erweitern Tree um einen dynamischen Knoten
- definieren eines TAB M-D
- definieren eines weiteren TAB (TAB25 Pivot)
- zeigen der TAB-Synchronisation



## 4 Praxisbeispiel

Innerhalb einer bestehenden Anwendungslandschaft bei einem Kunden wird sehr intensiv mit Schnittstellen gearbeitet.

Die Schnittstellen sind weitgehend per PL/SQL und Database-Links implementiert und damit sehr komfortabel.

Auch ausgewählte Daten aus SAP (FI,CO,..) werden per Database-Link lesender Weise an andere Systeme angebunden.

Daten per Webanwendung gepflegt müssen mit Abrechnungsdaten konsolidiert werden.

Es gibt auch Systeme mit MS-SQL und sogar Paradox-basierte Standardsoftware. Aus letzteren werden derzeit nur CSV-Daten ausgetauscht.

Im Lauf der Jahre hat die Datenqualität bzgl. dieser Schnittstellen gelitten.

Gerade weiche Informationen aus Excel-CSV erfordern einiges an Nacharbeit und Kontrolle.

Ein zukünftiges geplantes Teilprojekt wird

- Daten aus einer Paradox-DB
- mit Daten einer SAP-DB (Oracle)
- und eines weiteren Systems (Oracle, PL/SQL,Forms,Reports)

in einer gemeinsamen Oberfläche in optimierte Prozesse überführen.

Für bestimmte Prozesse wird es anstelle von 3 zu bedienenden Oberflächensystemen dann eines geben (DAP5), welches jedoch Daten aller 3 Basis-Systeme zeigen, zurodnen, synchronisieren und bearbeiten kann.

Derzeit schon im Einsatz ist eine DAP5-Anwendung, welche SQLSERVER-Daten mit Daten einer Oracle-basierten Bestandsverwaltung koppelt muss.

**Mit DAP5 ist dies ein Kinderspiel !**

Anhand von Beispiel MS3 demonstrieren !

***Kann man das mit Forms oder mit APEX sauber und einfach implementieren ?***

#### Demo S02 :

- Zeigen für MS3, wie die Zuweisung von Datasources zu APP gemacht ist
- wie die Berechtigungen gesteuert werden (APP\_STRUCT mit Rollen)
- wie unter Benutzer die Anmeldedaten an die Systeme (DS) verwaltet werden
- verschlüsseln Passworte
- Single-Sign-On

#### Demo MS3 :

- zeigen, wie sekundäre Datasources in DB-Aufrufen adressiert werden (20.800)
- zeigen im Kontext-Logging, was man aus den erfolgten Aufrufen analysieren kann
- weitere mögliche Informationen über alle VT's könnte man auch im Kontext ausgeben

#### Demo A24 :

- zeigen, was Hybridbetrieb ist - separate XDEV-Tabs programmieren
- Philosophie des HYBRID-Betriebes DAP5

#### Fazit :

Durch das in DAP5 implementierte System der DataSources (basierend auf XDEV-DS) und dem Parameter-Binding

können z.B. in einem Tab im Block

X Daten aus einem Oracle-System

Y Daten aus SAP

Z Daten aus Paradox (oder sonstiger ODBC-Quelle)

dargestellt und über die Datensätze der Blöcke synchronisiert werden.

## 5 Diskussion und Ausblick

### Noch auf der ToDo-Liste :

- Codebasis DAP5 auf XDEV3 migrieren .. evtl. Redesign Oberfläche
- Validierungs-Engine fertigstellen (Cursor-Tracking Maus und Tastatur !)
- weitergehende Formatierungen von Spalteninhalten (XDEV3 Unterst.)
- Excel-Import-Export erweitern
- Benutzerspezifische Einstellungen speichern und verwalten
- 
- APP-Repositories in DAP5-REPOS aufbauen ..... gemeinsames über alle DB-Systeme .... Oracle ... SQLSERVER ... MYSQL ... ODBC .....
- Abgleich und Update-Routinen für diese Repositories
- letztlich die DEMO-Doku-Anwendung fertigstellen
- Bereitstellung der DEMO-Doku-Anwendung im Web
- Bereitstellung der Workbench im Web
- Bereitstellung einer interaktiven Demo-Anwendung im Web zum Spielen
- Vorbereiten Workshops (1-2 Tage)

### Diskussion

- Sicherheit , Schutz gegen SQL-Injection
- Sicherheit , Schutz gegen REPOS-Manipulation
- Zeitersparnis, wie agil kann man entwickeln ?
- wie lange braucht man, um DAP5 zu beherrschen ?
- .....
- .....

## **6      Kontakt**

**Hans Niedermeier  
PRO-Software GmbH  
84186 Vilsheim**

**08706 941024**

**0175 2713194**

**hn@dap5.de**