

Szenarien DPE Reporting

Das nachfolgende Dokument beschreibt mögliche Szenarien zur Generierung von Reports aus dem Delmia Process Engineer (DPE).

1 Einleitung

Der DPE ist eine Lösung zur Prozeßplanung und basiert auf einer Client-Server- Architektur (Multi-Tier-Architektur).

- **Client**
Der Client bildet die Schnittstelle des DPE zum Anwender. Der DPE-Client ist ein Thin- Client, d.h. er bildet keine Business-Rules ab, dient zur Visualisierung der Ergebnisse und übergibt Anwenderaktionen an den DPE- Server zur weiteren Verarbeitung.
- **Server**
Der DPE-Server bildet den zentralen Kern des DPE. Hier finden sich die Business- Rules, Transaktionsverwaltung, Rechteverwaltung, etc.. Zudem verwaltet der Server das Objektmodell. Der DPE-Server ist ein Applikationsserver und beinhaltet die "Intelligenz" des DPE. Da der DPE die Möglichkeit bietet, Rechnerlast auf mehrere Applikationsserver zu verteilen, sind zudem spezielle Server vorhanden, die die Koordination beim Multiserver- Betrieb unterstützen.
- **Persistenzschicht**
Die Persistenzschicht hat die Aufgabe Daten dauerhaft zu speichern. Basis für die Persistenzschicht ist eine Oracle-Datenbank. Zur Konvertierung des DPE- Objektmodells in ein relationales Modell (Oracle-Datenbank) wird eine zusätzliche Transformationsschicht verwendet. Derzeit wird das Objektmodell in ca. 300 Oracle- Tabellen und ca. 100 Views abgebildet.

Zur Generierung von Reports bzw. zur Bereitstellung von Daten des DPE an Drittsysteme bietet der DPE folgende, von der Entwicklung unterstützte, Möglichkeiten an:

- Nutzung des internen Report-Generators List&Labels.
- Verwendung von API-Funktionen zur Datenextraktion (Scripting).

Eine Stand-Alone Reporting- bzw. Schnittstellen-Lösung ist derzeit nicht im Standardumfang des DPE enthalten.

Der Direktzugriff auf Datenbank-Objekte des DPE wird seitens DELMIA nicht unterstützt.

2 Detaillierung der Szenarien

Nachfolgend werden folgende Szenarien beschrieben:

- Report-Generierung mit Hilfe von List&Labels.
- Report-Generierung bzw. Datenextraktion durch Nutzung der API-Funktionen (Scripting).
- Verwendung von generischen Konzepten zur Bereitstellung von Daten aus dem DPE.
- Direktzugriff auf die Oracle-Objekte der Persistenzschicht und Bereitstellung von Views zur Weiterverarbeitung durch Drittsysteme.

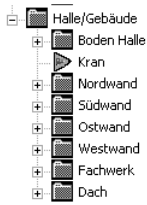
Für die Szenarien erfolgt die Beschreibung nach folgendem Schema:

- Nutzung des Szenarios aus Anwendersicht.
- Kurze Beschreibung der notwendigen technischen Lösung.
- Einsatzbereich des Szenarios.
- Aufwand zur Bereitstellung des Szenarios (qualitative Bewertung).

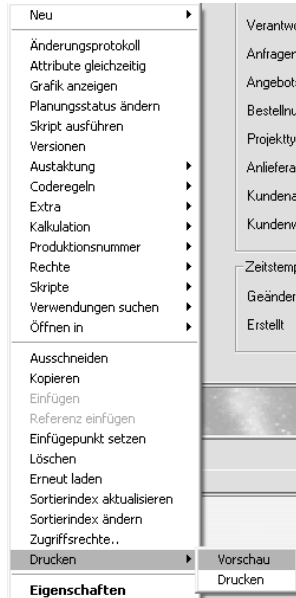
2.1 List&Labels.

- Nutzung des Szenarios aus Anwendersicht:

Der Anwender navigiert innerhalb des DPE zu dem Objekt, für das eine Auswertung gewünscht ist.



Nach Auswahl des Objekts wird über das Kontextmenü der Druck ausgewählt.



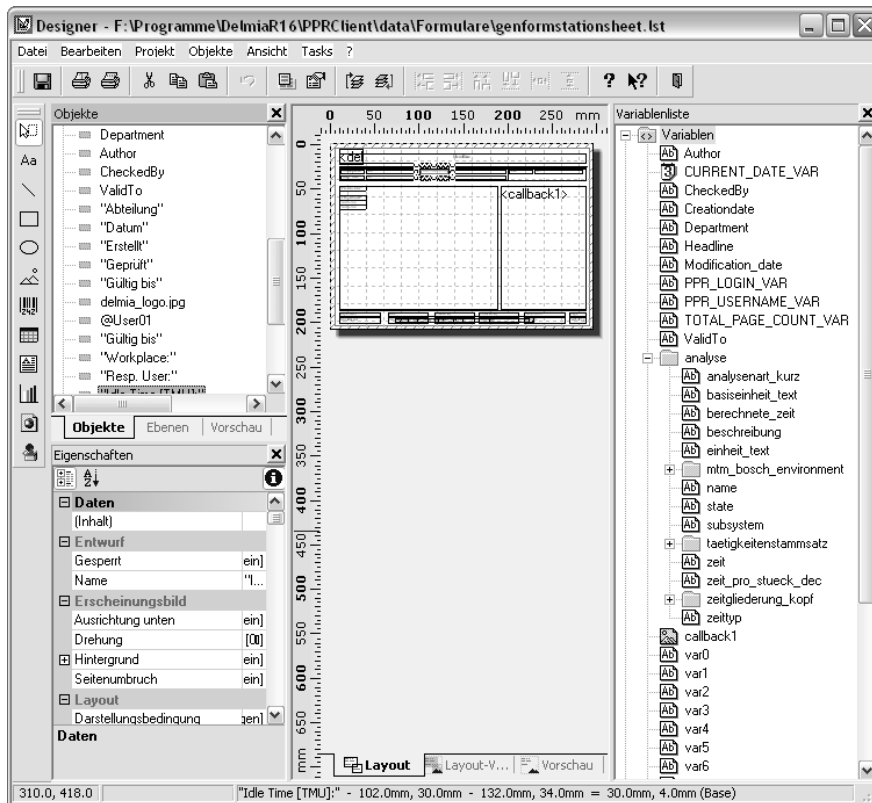
Je nach Konfiguration des Druckformulars wird der Report direkt zu einem Drucker geschickt oder als PDF-Datei gespeichert.

- Kurze Beschreibung der notwendigen technischen Lösung:

Bevor ein Druckformular zur Reporterzeugung verwendet werden kann, muß es definiert und konfiguriert werden. Eine Ausnahme bilden jene Anwendungsfälle, in denen die bereits vorhandenen Druckformulare ausreichend sind.

Die Definition und Konfiguration eines Druckformulars erfolgt mit Hilfe des List&Labels Formulargenerators. Hier werden Layout und Daten entsprechend der Anforderungen an den Report gestaltet und definiert.

Für komplexere Reports, z.B. die Daten sollen von Objekten stammen, die über verschiedene Relationen mit dem auszuwertenden Objekt verknüpft sind, müssen die Daten mit Hilfe sog. Script-Variablen bereitgestellt werden. Script-Variablen sind Objekte, deren Werte mit Hilfe von Scripten (API-Zugriff) bestimmt werden.



- Einsatzbereich des Szenarios:

Die Nutzung der Reporting-Funktionalität auf Basis der List&Labels Druckformulare bietet sich in den Fällen an, in denen ein aktueller Datenbestand ausgewertet werden muß. Bei Änderung der Anforderungen an die Druckformulare müssen diese entsprechend angepaßt werden. Die Druckformulare bieten direkte Druckausgabe oder Ausgabe in eine PDF-Datei.

- Aufwand zur Bereitstellung des Szenarios (qualitative Bewertung):

Sind für die Anforderungen an den Report die bereits vorhandenen Druckformulare ausreichend, so kann dieses Szenario unmittelbar genutzt werden.

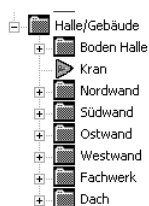
Ergeben sich weitergehende Anforderungen an einen Report, so sind folgende Aufwände notwendig:

- Anforderungsanalyse mit dem Anwender.
- Definition von Layout und Daten.
- Ggf. Aufwand zur Erstellung von Script-Variablen.
- Umsetzung mit Hilfe des List&Labels Report-Generators.

2.2 Nutzung der API-Funktionen (Scripting).

- Nutzung des Szenarios aus Anwendersicht:

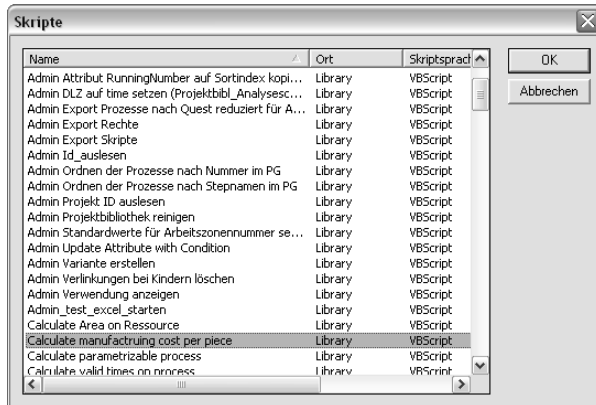
Der Anwender navigiert innerhalb des DPE zu dem Objekt, für das eine Auswertung gewünscht ist.



Über das Kontextmenü kann mittels "Skript ausführen" die Liste der möglichen Auswertescripte aufgerufen und ein Auswertescript zur Ausführung gebracht werden.

Hinweis:

Diese Funktionalität kann dem Anwender komfortabler zur Verfügung gestellt werden, indem dem auszuwertenden Objekt direkt eine Liste der möglichen Auswertescripte zugeordnet wird. Damit stehen dem Anwender nur jene Scripte zur Ausführung bereit, die für das selektierte Objekt Sinn ergeben.



- Kurze Beschreibung der notwendigen technischen Lösung:

Eine Reporting-Lösung auf Script-Basis baut sich prinzipiell aus 3 Bestandteilen auf:

- DPE-Script zur Navigation der auszuwertenden Strukturen und Datenextraktion.
- Aufbereitung der Daten entsprechend dem gewünschten Ausgabeformat.
- Ausgabe der Daten nach Excel, XML oder andere Formate.

- Einsatzbereich des Szenarios:

Die Nutzung von Scripting-Funktionalitäten zur Erzeugung von Auswertungen bietet sich dann an, wenn komplexe Auswertungen gewünscht sind. Durch die Navigation und Datenextraktion mit Hilfe der API-Funktionen können zudem diverse Filter bei der Auswertung berücksichtigt werden. Script-basierte Auswertungen können bis zu einem bestimmten Grad flexibel entwickelt werden, aber der Normalfall ist, daß derartige Auswertungen unverändert bleiben.

Ein weiterer positiver Aspekt der Script-basierten Auswertungen ist die Möglichkeit, die Auswertung batch-fähig zu gestalten, so daß z.B. zeitintensive Auswertungen auch über Nacht möglich sind.

- Aufwand zur Bereitstellung des Szenarios (qualitative Bewertung):

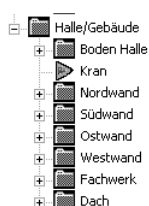
Script-basierte Auswertungen erfordern Entwicklungsaufwand. Daher ist für den Erfolg die aktive Mitwirkung an der Anforderungsanalyse unbedingt notwendig.

- Anforderungsanalyse mit dem Anwender.
- Definition von Layout und Daten.
- Erstellung der Script-Lösung.
- ggf. Definition von Report-Schablonen (z.B. Excel-Templates).

2.3 Generische Mechanismen

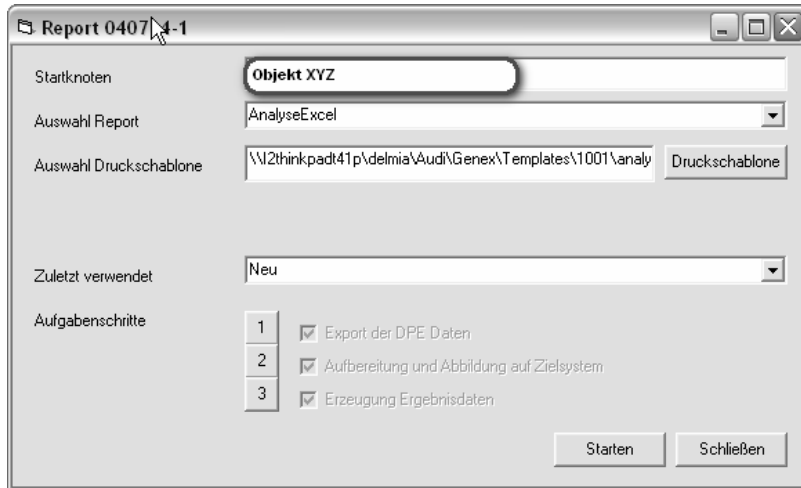
- Nutzung des Szenarios aus Anwendersicht:

Der Anwender navigiert innerhalb des DPE zu dem Objekt, für das eine Auswertung gewünscht ist.



Für das auszuwertende Objekt wird über das Kontextmenü eine Liste der möglichen

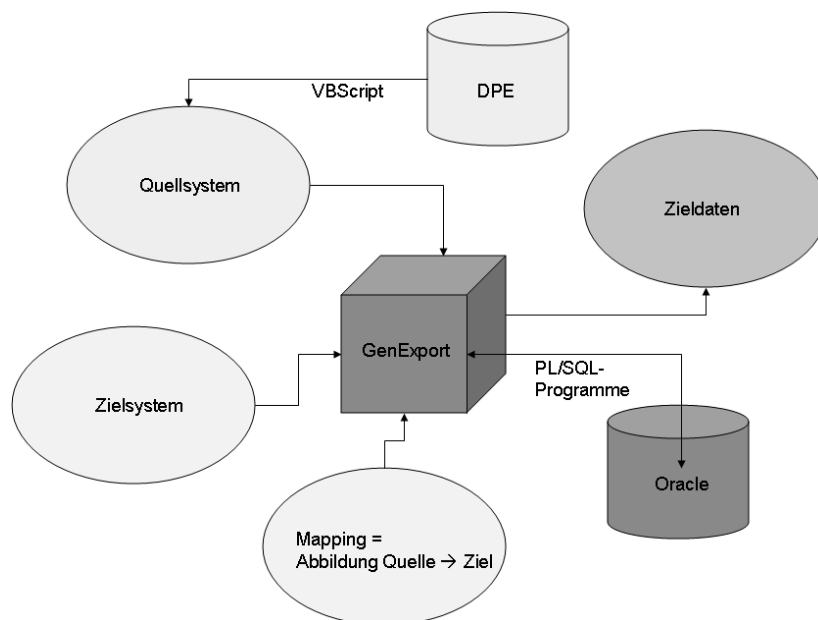
Auswertescripte dargestellt und vom Anwender ausgewählt.
 Nach dem Start des Auswertescripts erhält der Anwender einen Dialog, aus dem er bestimmen kann, welche Auswertung für das Objekt erfolgen soll ("Auswahl Report") und in welcher Form die Auswertung gewünscht wird ("Auswahl Druckschablone").
 Diese Funktionalität ermöglicht es dem Anwender unterschiedlichste Auswertungen in unterschiedlichen Ausgabeformaten für ein und dasselbe Objekt zu nutzen.



- Kurze Beschreibung der notwendigen technischen Lösung:

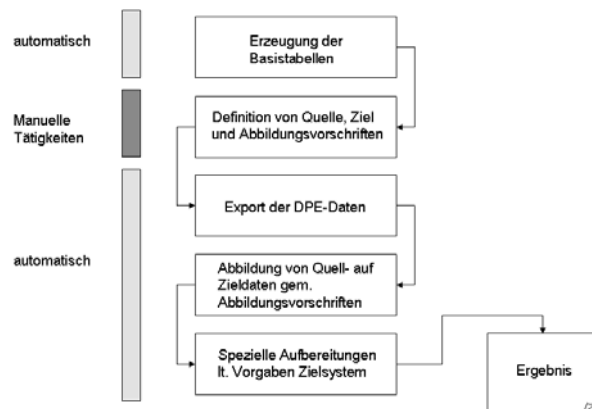
Generische Mechanismen nutzen die Scripting-API-Funktionalität zur Navigation und Extraktion der DPE-Daten.

Die Definition der auszuwertenden Objekte (Plantypen, Attribute, Relationen, ...) erfolgt mit Hilfe von Konfigurationen (nicht identisch mit dem Begriff der "Konfiguration" des DPE). Es können beliebig viele Konfigurationen erstellt und verwaltet werden.



Die entsprechend der Konfiguration extrahierten Daten werden in Struktur- und plantype-relevanten Attribute-Werte-Tabellen gehalten.
 Diese Tabellen bilden die Grundlage für die Aufbereitung der Daten entsprechend der in der Konfigurationen hinterlegten Anforderungen.
 Die momentan verfügbare Lösung verwendet für die o.g. Tabellen eigene Oracle-Tabellen und für die Aufbereitung Oracle-PL/SQL-Programme.
 Eine weitere Lösung, die sich derzeit noch im Beta-Stadium befindet, nutzt XML-basierte Mechanismus und baut interne Strukturen auf, um die extrahierten Daten zu halten.

Der Ablauf der einzelnen Bausteine ist in der nachfolgenden Abbildung dargestellt.



Die aufbereiteten Daten können in beliebiger Form dargestellt werden. Die Definition der Ergebnisdarstellung erfolgt ebenfalls in der Konfiguration.

- Einsatzbereich des Szenarios:

Die generischen Mechanismen sind dann sinnvoll, wenn eine große Zahl unterschiedlichster Schnittstellen bedient werden müssen oder Auswertungen notwendig sind.

Bisher wurden mit dem beschriebenen Mechanismus folgende Anforderungen umgesetzt:

- Erzeugung von Schnittstellendaten für Dritt-Systeme.
- Erzeugung von XML-Daten zur Weiterarbeitung durch XML-Prozessoren.
- div. Druckberichte und Auswertungen.

Der Vorteil des Einsatzes generischer Ansätze ist immer dann gegeben, wenn aufgrund von Anforderungsänderungen oder -anpassungen kurzfristig Änderungen notwendig sind.

Da die generischen Bausteine (Scripte, PL/SQL-Programme) ausschließlich über die Konfigurationsdaten gesteuert werden, sind neue Anforderungen über neue oder angepasste Konfigurationen zu realisieren, so daß Entwicklungsaufwand nicht notwendig ist.

- Aufwand zur Bereitstellung des Szenarios (qualitative Bewertung):

Die generischen Mechanismen bestehen aus bereits vorhandenen Programmbausteinen. Diese Bausteine decken bereits einen Großteil möglicher Anwendungsfälle ab. Sollen sehr spezielle Mechanismen über den generischen Ansatz abgebildet werden, so wäre Aufwand für diese Spezialanforderungen notwendig.

Der "normale" Aufwand zur Bereitstellung der generischen Mechanismen umfaßt i.A. folgende Schritte:

- Initiale Einrichtung von Scripten und ggf. Datenbanktabellen (einmaliger Aufwand).
- Anforderungsanalyse mit dem Anwender und Erstellung Lastenheft.
- Ableitung des Pflichtenheftes in Form einer Konfiguration.
- Definition von Report-Schablonen (z.B. Excel-Templates).
- Einspielen der Konfiguration.
- ggf. Definition von Postprozessor-Aktionen.

2.4 Bereitstellung von Views zur Weiterverarbeitung durch Drittsysteme

- Nutzung des Szenarios aus Anwendersicht:

Der Anwender startet das Drittsystem zur Generierung der Auswertung. Eine direkte Auswahl des auszuwertenden Objekts ist mit Hilfe von DPE-Client-Mechanismen nicht möglich, da die Kommunikation DPE-Drittsystem nicht gegeben ist.

- Kurze Beschreibung der notwendigen technischen Lösung:

Zur Bereitstellung von Views, die zur Weiterverarbeitung durch Drittsysteme verwendet werden

sollen, müssen Teile der Serverlogik in die Definition der Views einfließen.
 Die Verwendung der Standard-DDL-Mechanismen zur Definition erlaubt nur die Definition relativ einfacher Views (auch bei Nutzung von Joins über mehrere Datenbanktabellen), so daß Teile der Serverlogik unter Verwendung einer noch unbestimmten Anzahl von Stored-Procedures nachgebildet werden müssen.

Offen bleibt die Frage, wie z.B. Filter, Projektauswahl, in die bereitgestellten Daten einfließen sollen. D.h. sämtliche Selektionskriterien, die zur Laufzeit über den DPE- Client vom Anwender definierbar sind, müßten zusätzlich in die Views einfließen, damit sie auswertungsbezogen verwendet werden können.

- Einsatzbereich des Szenarios:

Der Ansatz Auswertungen mit Hilfe von weiterverarbeitbaren Views zu generieren bietet eine Unabhängigkeit der Auswertung von der aktuellen Verfügbarkeit des DPE. Die Anzahl und Art der Auswertung ist vom Anwender selbst definierbar (und verantwortbar, da z.B. eingeschränkte Zugriffsrechte beim Direktzugriff nicht ausgewertet werden).

Die Views bieten den universellsten Zugang zu den Daten des DPE und ermöglichen Lösungen, die Stand-Alone betrieben werden.

Da eine View eine logische Sicht auf bestimmte Objekte bietet, muß die View ständig aktualisiert werden, so daß mit einer gesteigerten Grundlast des Systems zu rechnen ist.

- Aufwand zur Bereitstellung des Szenarios (qualitative Bewertung):

Zur Definition der Views sind genaue Definitionen der bereitzustellenden Daten notwendig. Sind zudem komplexe Strukturen abzubilden, dann ergeben sich zusätzliche Entwicklungsaufwände für die Erstellung der Stored-Procedures, die für die DDL- Definition der Views notwendig werden.

Der Aufwand zur Bereitstellung der Views umfaßt folgende Schritte:

- Anforderungsanalyse mit dem Anwender und Erstellung Lastenheft.
- Entwurf der DDLs zur Generierung der Views.
- Entwicklung von Stored-Procedures, um DDL-Mechanismen zur erweitern.
- Generierung der Views.
- Definition von Layout und Daten mit Hilfe des Drittsystems.
- Umsetzung mit Hilfe des Report-Generators.

3 Fazit

Die nachfolgende Abbildung faßt die Einzelkriterien der vorgestellten Lösungsansätze zusammen. Ggf. erfordern einige der aufgeführten Bewertungen weitere Diskussion.

Standard 9-3-1		1	2	3	4
schlecht / nicht möglich		◆	○	▽	●
mittel		◇	○	▽	●
sehr gut / einfach		▽	○	▽	●
		List&Labels	Scripting	Generische Mechanismen	Bereitstellung von Views
		1	2	3	4
Aufwand für Erstellung	1	▽	○	▽	●
Laufzeit Report	2	○	○	▽	▽
Flexibilität für Anpassungen	3	○	●	▽	●
Serverlast	4	○	○	▽	○
Aufwand für Anwendung	5	▽	▽	▽	○
Definition komplexer Strukturen	6	○	▽	▽	●
Definition Filter	7	▽	▽	▽	●
Wartungsaufwand	8	▽	○	▽	●
Wartungsmöglichkeiten durch Anwender	9	▽	●	○	●
Stand-Alone Lösung möglich	10	●	●	●	▽
Anbindung 3rd-Party Lösungen	11	●	○	▽	▽
Generische Lösung möglich	12	●	●	▽	●