

[www.D-S-T-G.com](http://www.D-S-T-G.com)

Generative tools for  
automated software  
development and  
modernization

**Delta Software  
Technology**

provides Europe's  
leading companies with  
state-of-the-art software  
generator technology for  
more than 30 years



The Generator Company

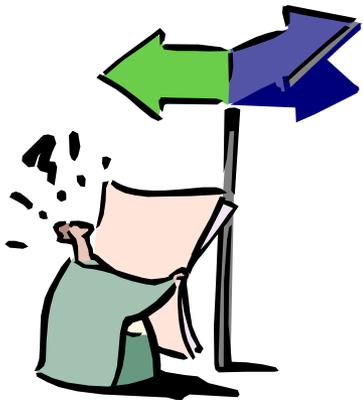
# Model-Driven Development and Product Lines, Student Day, Leipzig 2009

## Persistency Layer im Vergleich: Sind generierte Adapter besser?

Cord Giese

Delta Software Technology GmbH

Konzepte für Persistenzadapter



SCORE<sup>®</sup> Adaptive Bridges –  
Data Architecture Integration<sup>™</sup>

MINT – Adapter im Wettbewerb

Zusammenfassung

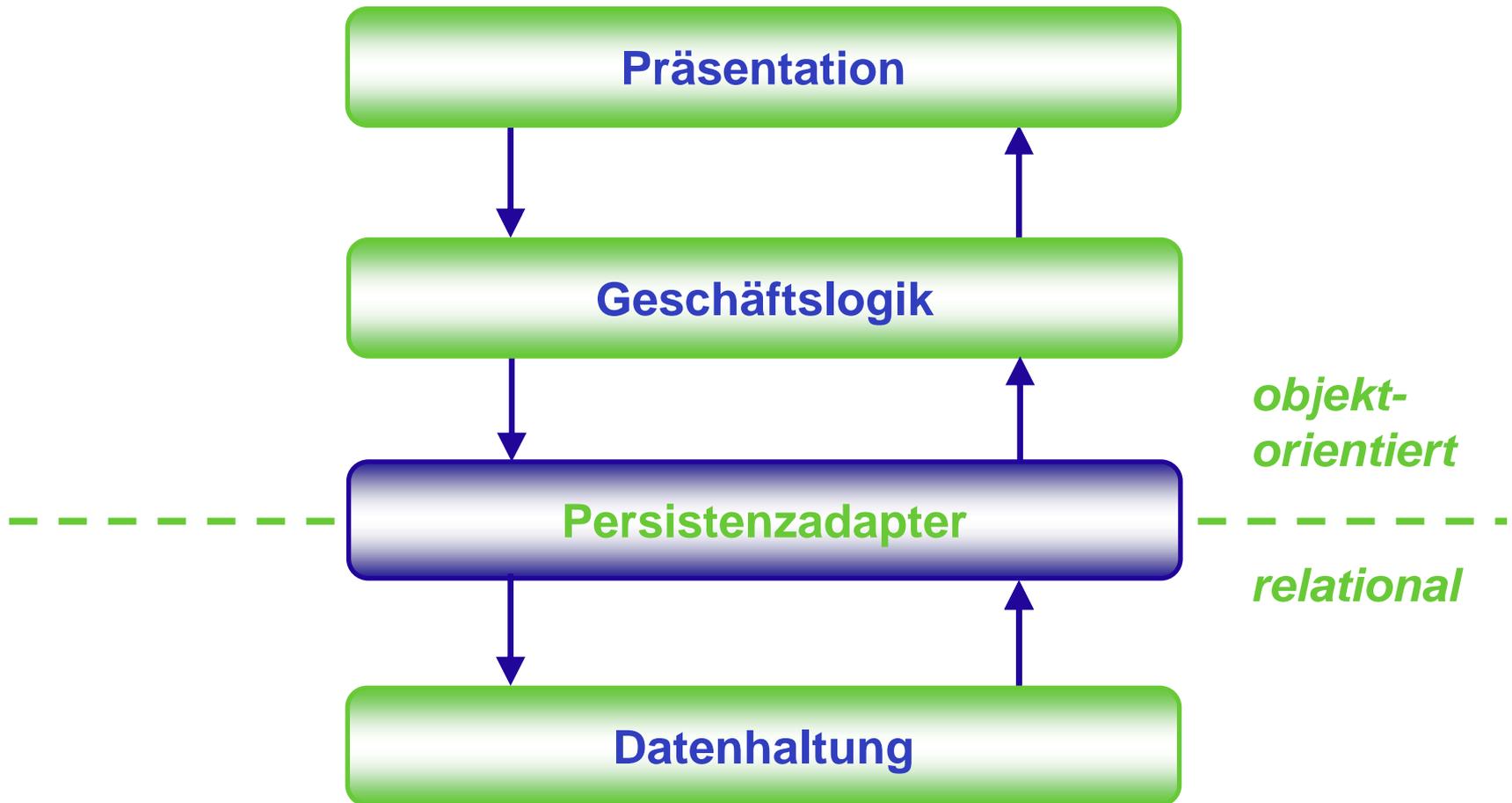
## Konzepte für Persistenzadapter



SCORE<sup>®</sup> Adaptive Bridges –  
Data Architecture Integration<sup>™</sup>

MINT – Adapter im Wettbewerb

Zusammenfassung



- Generische Adapter-Lösung
  - Adapter unabhängig vom Datenmodell (Objektmodell), vom DBMS, von den Anwendungen
  - → Persistenz-Framework mit Konfiguration eines Laufzeit-OR-Mappings
- Beispiele
  - Telerik OpenAccess
  - NHibernate

- Generierte Adapter-Lösung
  - Adapter zugeschnitten auf Datenmodell (Objektmodell)
  - → Persistenz-Framework mit Generierung aus Modelldaten
- Beispiele
  - AndroMDA
    - „Out-of-the-box“-Codegenerator
  - openArchitectureWare
    - Modulares Generator-Framework, vor allem zur Entwicklung eigener Modelltransformationen

- Manuelle Implementierung eines Adapters
  - Spezifisch für Objektmodell und gegebene Zugriffstechnologie, z.B. mit ADO.NET
  
- Adapter-Generierung aus Modelldaten mit ...
  - SCORE<sup>®</sup> Adaptive Bridges – Data Architecture Integration<sup>™</sup>

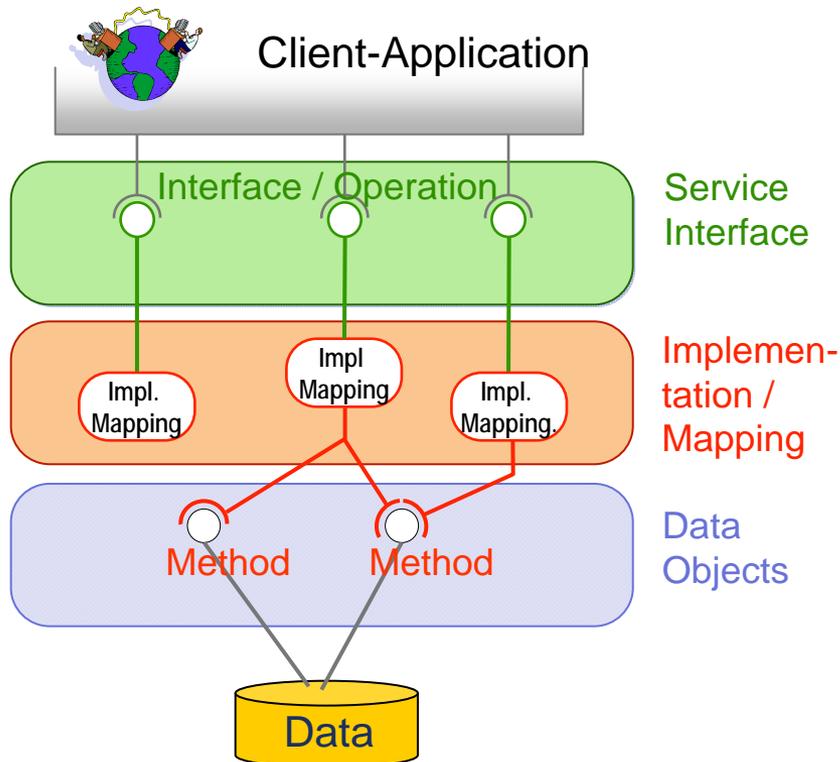
Konzepte für Persistenzadapter



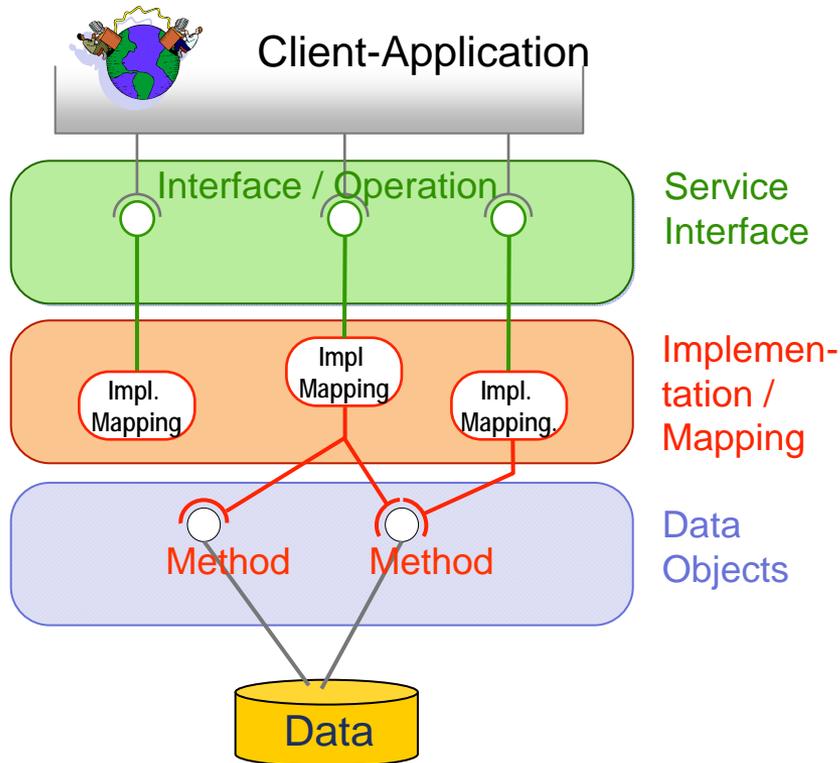
**SCORE<sup>®</sup> Adaptive Bridges –  
Data Architecture Integration<sup>™</sup>**

MINT – Adapter im Wettbewerb

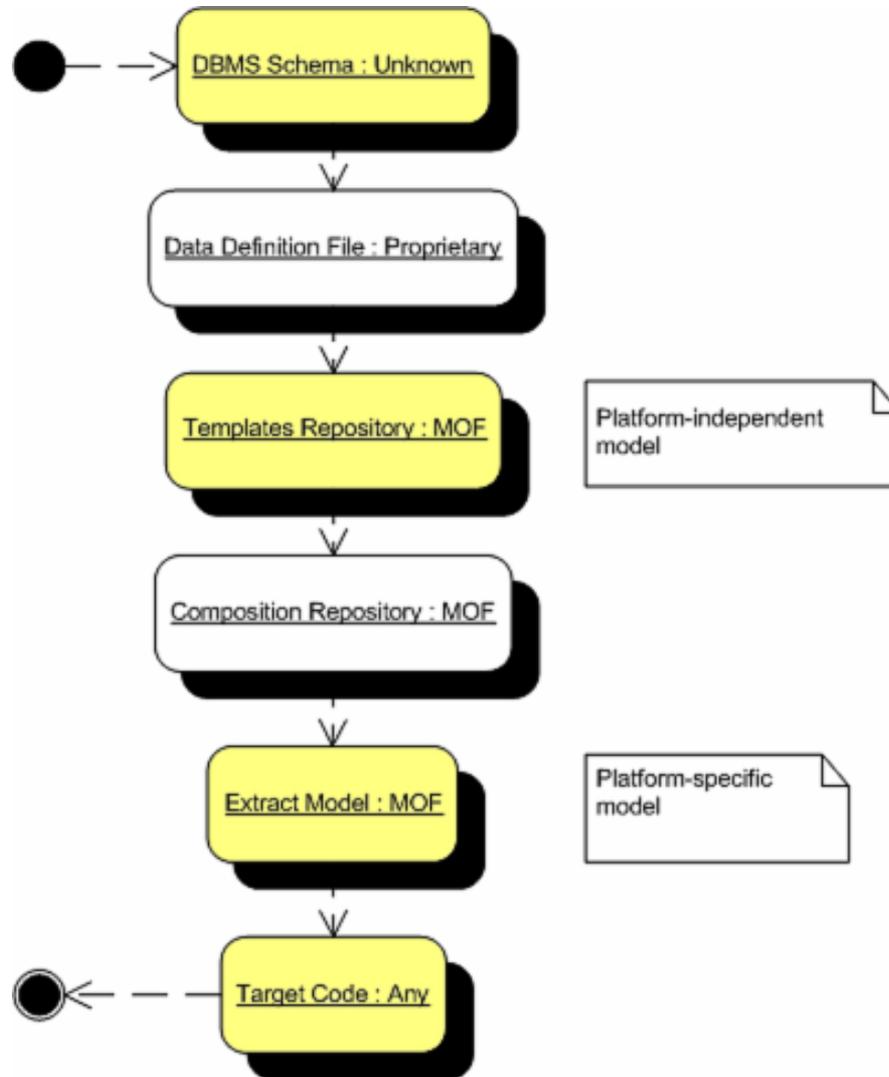
Zusammenfassung



- Datenobjekte:
  - Definition von Datenzugriffen auf Basis von DB-Schema
  - **Plattformunabhängig**
- Service-Interface:
  - Beschreibung der Schnittstelle einer Komponente
  - Entkoppelt von Datenobjekten durch Implementierung/Mapping



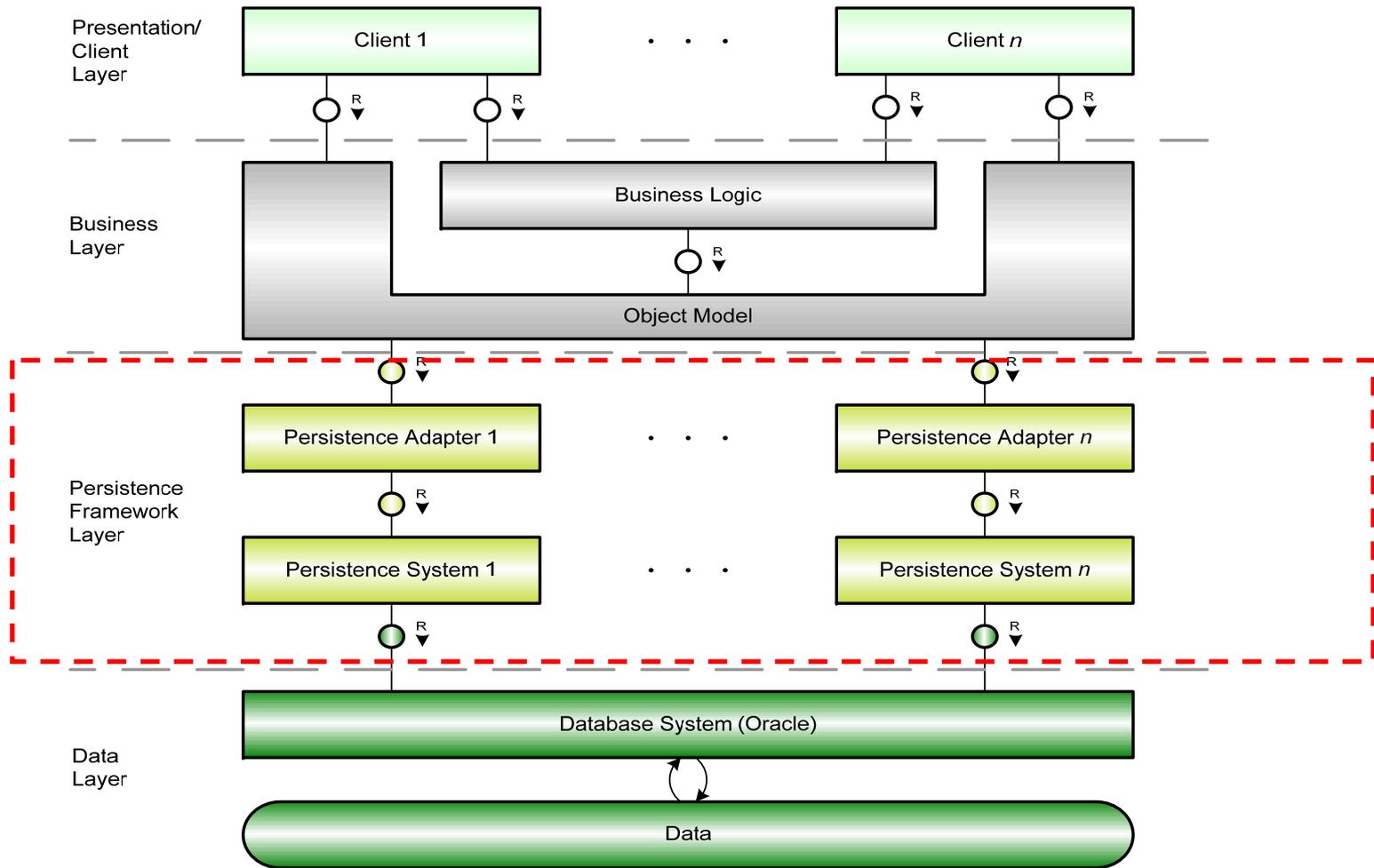
- Implementierung/Mapping:
  - Verbindung von Operationen mit Methoden
  - Mapping von Operationsparametern auf Methodenparameter (Attribute) und vice versa
  - Optional: zusätzliches Anwendungscoding



- Werkzeugkette:
  1. DataDef-Generator
  2. Templates-Generator
  3. SCORE® Composition Manager™
  4. Extract Model-Generator
  5. Application Adapter-Generator
  
- Manuelle Ergänzungen möglich vor PIM- und PSM-Erzeugung

- Aufgabe: Kopplung objektorientiert modellierter Geschäftslogik mit bestehenden relationalen Datenbanksystemen
- Zielplattform: Oracle-DB, Clients in C#, Zugriff per ADO.NET
- DB und Testanwendungen entnommen aus dem MESCOR-System einer Bank
  - „Tool Suite“ zur Analyse von Unternehmen und deren Aktien
  - → **Realdaten!**

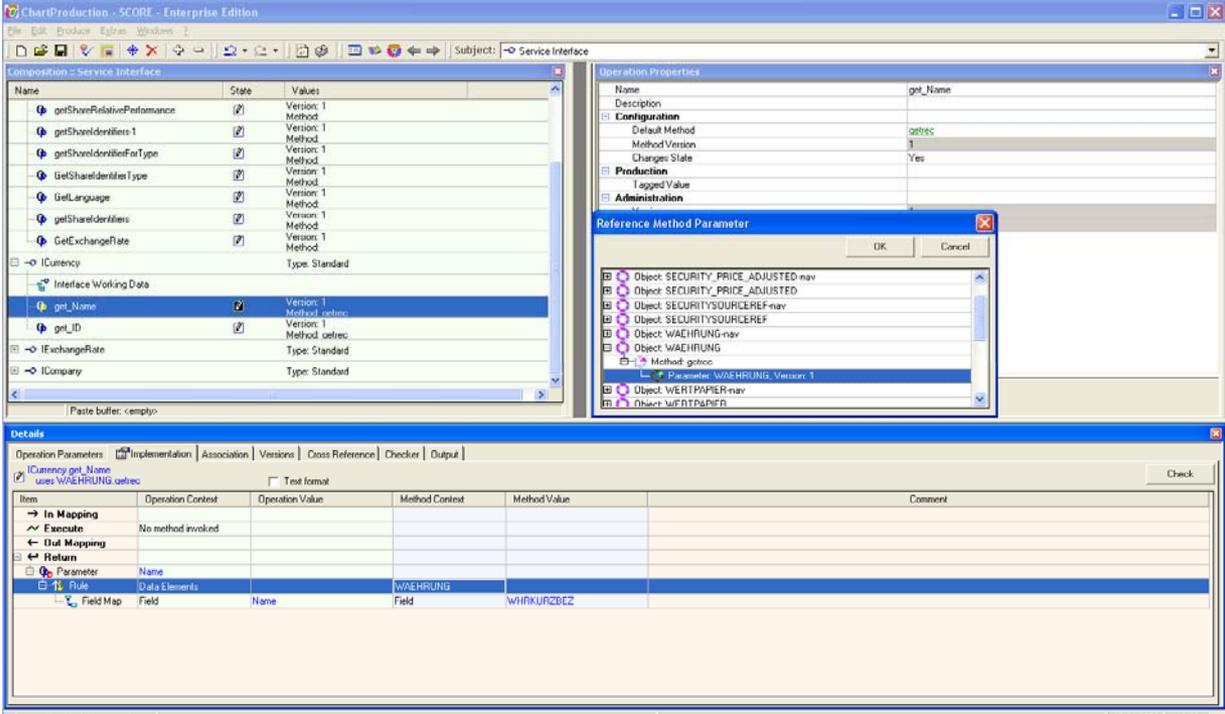




- Drei Testanwendungen:
  - 1) ChartProduction
    - Anzeige von Aktienkursen (Chart-Diagramme)
    - Nur lesende Zugriffe (Selektiv und Batch)
  - 2) Accounting
    - Unternehmensdatenverwaltung
    - Lesende und schreibende Zugriffe (Selektiv)
  - 3) ImportTool
    - Import von Aktienkurs-Daten in die DB
    - ca. 800000 Datensätze
    - Nur schreibende Zugriffe (Batch)

- **Data Definition File**, automatisch aus DB-Schema erzeugt
  - DBMS-unabhängig
  - Vollständig (nicht immer alle Informationen im Schema)
    - Include-Dateien für Ergänzungen
  - Editierbares Textformat
- **Templates Repository** = automatisch erzeugtes PIM
  - Basis für Definitionen in SCORE<sup>®</sup> Composition Manager<sup>™</sup>
- Eine Oracle-DB, eine Data Definition, ein Templates Repository für alle drei Testapplikationen

# Schritt 3/5: Composition Repository



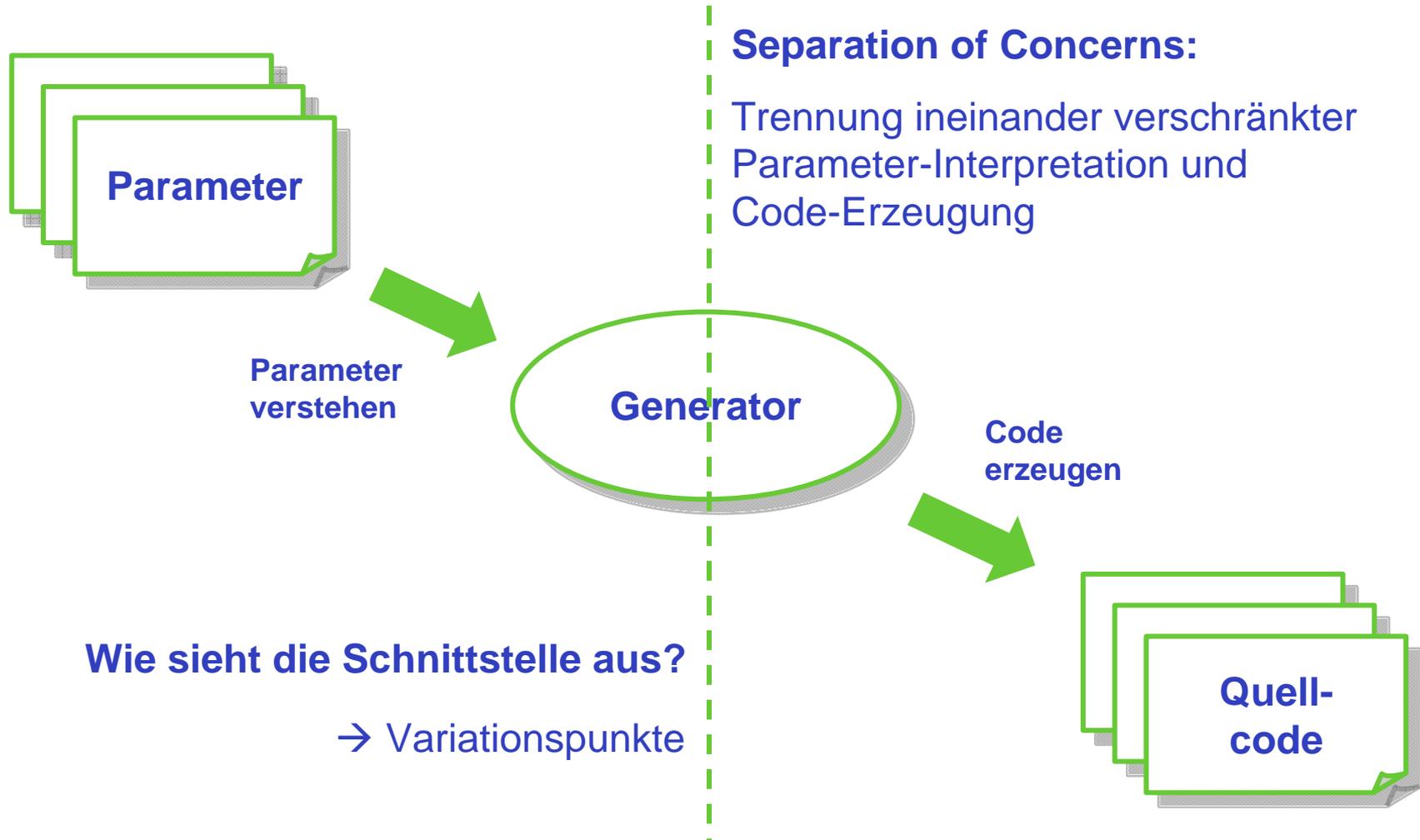
The screenshot shows the SCORE Enterprise Edition interface. The main window displays a tree view of service interfaces, with 'get\_Name' selected. A 'Reference Method Parameter' dialog box is open, showing a list of objects including 'Object: WAERHUNG, Version 1'. The 'Details' pane at the bottom shows the configuration for the 'get\_Name' operation, including a table for 'In Mapping' and 'Out Mapping'.

Item	Operation Context	Operation Value	Method Context	Method Value	Comment
→ In Mapping					
↔ Execute	No method invoked				
← Out Mapping					
← Return					
Parameter	Name				
Rule	Data Elements	Name	WAERHUNG	WAERHUNG	
Field Map	Field	Name	Field	WAERHUNG	

- SCORE® Composition Manager™: Konfiguration von Datenobjekten, Service-Interfaces und Mappings
- Hier: ein Composition Repository pro Applikation

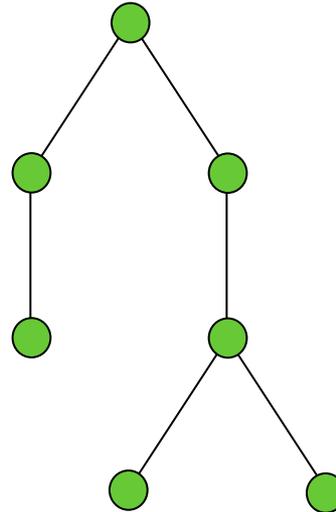
- Extract Model (PSM)
  - Automatisch erzeugt, temporär
- Resultat: Zielcode für ADO.NET / C# / Oracle
  - ... und die jeweilige Anwendung
  - "Datenobjekte" → Zugriffsklassen
  - "Service Interface" → Adapterklassen
    - Aufrufe der Zugriffsklassen

■ Object Model und Applikationen bleiben unverändert!



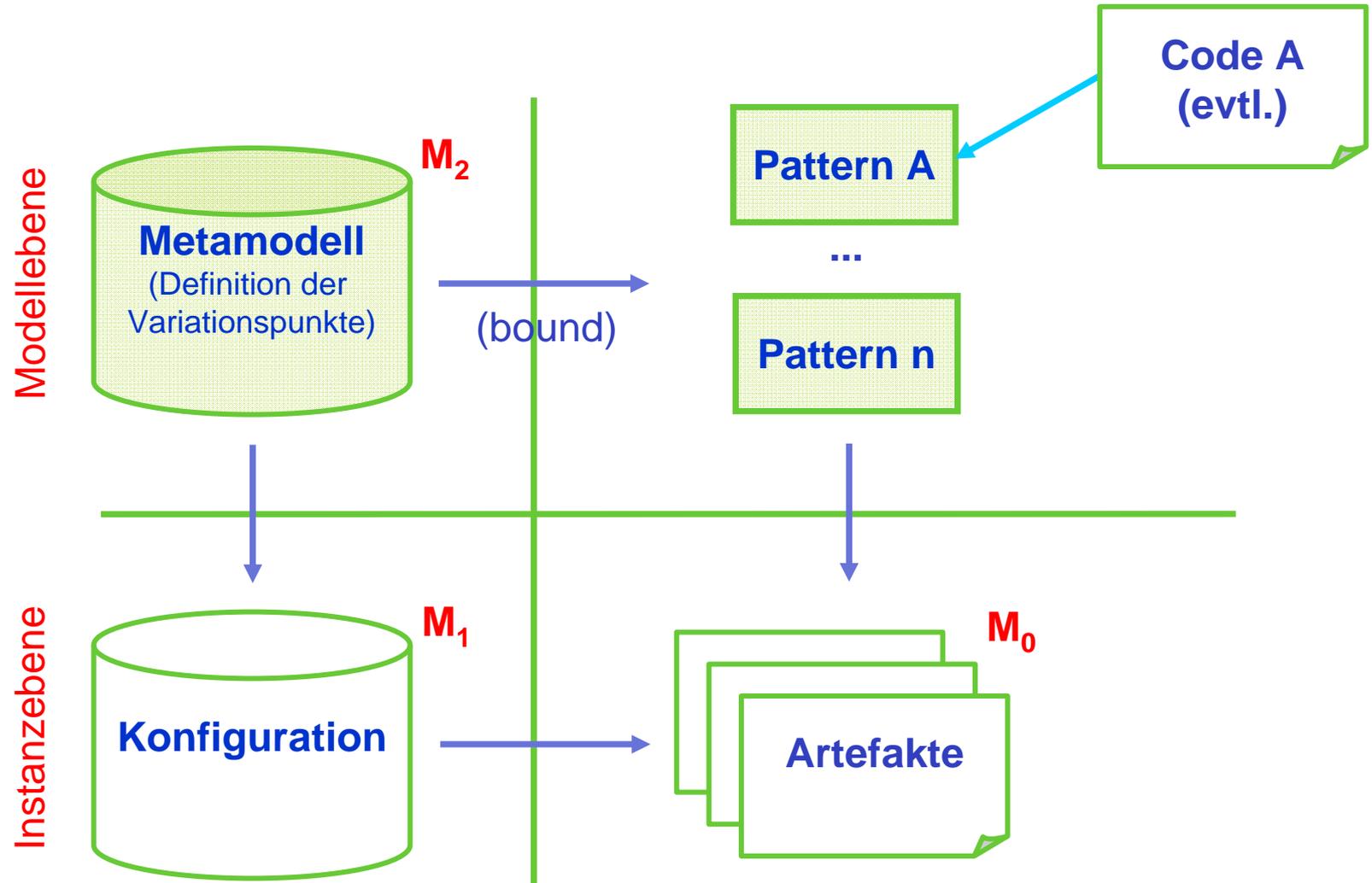
Import von  
Konfigurations-  
daten

«interface»



Zusammenbau  
von Code-  
Fragmenten

- HyperSenses: Generatorentwicklungs- und -anwendungssystem
  - Variabilitätsmodell (Metamodell)
  - Code Patterns, gekoppelt an das Variabilitätsmodell, formen „Renderings“ (Sichten)
- Pattern By Example™ (PBE)-Methode
  - Automatisierung von „Copy-Paste-Adapt“



- HyperSenses @ SCORE: Generator-Implementierung und -konfiguration
  - Application Adapter-Generator
  - Mapping-Definitionen in SCORE® Composition Manager™
- Application Adapter-Generator: Erstellung von HyperSenses-Code-Patterns
  - Genau: zu existierenden Code-Patterns wurde je eine Implementierung für ADO.NET (C#) hinzugefügt
  - Metamodell = Extract Model
  - Zielcode-Prototyp als Vorlage (→ PBE)



- Mapping-Definitionen in SCORE® Composition Manager™:  
Erstellung von Mapping-Patterns
  - → in Oberfläche eingebettete „Sub-DSL“
- Standard-Patterns für Standard-Mappings,  
z.B. zur Navigation:
  - **GetRelatedToOne** – Fremdschlüssel-Zugriff
  - **GetRelatedToMany** – Zugriff entlang einer Assoziation  
mit Liste als Resultat
- Möglich: projektspezifische Patterns



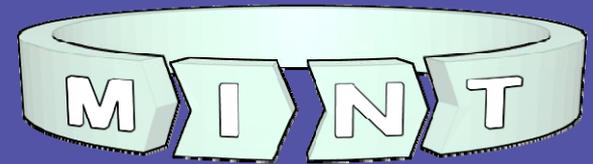
Konzepte für Persistenzadapter



SCORE<sup>®</sup> Adaptive Bridges –  
Data Architecture Integration<sup>™</sup>

**MINT – Adapter im Wettbewerb**

Zusammenfassung



- BMBF-Projekt MINT
  - Modellgetriebene Integration von Informationssystemen
- Schwerpunkt Datenintegration
  - Anbindung relationaler Legacy-Datenbanken an OO-Anwendungen durch Persistenzadapter
- Evaluierung verschiedener konkurrierender Ansätze
  - → MESCOR-Testumgebung
- Partner: andrena objects, BTC, FZI, OFFIS, Uni Oldenburg
- [www.mint-projekt.de](http://www.mint-projekt.de)

- Auszug aus den Vergleichsmetriken zu Wartungsaufwänden für Accounting
- Metrik 1.2.1: Anzahl der Personentage um die Abbildungen der jeweiligen Applikation zu implementieren
- Metrik 1.2.2: Zeit die für das Testen und Debuggen der Applikationen gebraucht wurde

Metrik[Einheit]	ADO	NHibernate	OpenAccess	oAW
M1.2.1[PD]	4,2	3	3,25	<b>0,9</b>
M1.2.2[PD]	8,1	3,2	4,2	<b>1,5</b>

- AndroMDA
  - Schwierigkeiten bei Abbildung der Altdatenbank sowie bei Anpassung an Interfaces
  - → nur bedingte Eignung von Out-of-the-box-Generatoren für Legacy-OR-Abbildungen
  
- NHibernate und OpenAccess
  - Gute Ergebnisse bei Migrations- und Evolutionsszenarien
  - Workarounds notwendig

- ADO.NET (manuell)
  - Höchster Aufwand
- openArchitectureWare
  - „... verkürzt der Einsatz von modellgetriebenen Techniken die Implementierungszeit zusätzlich“ (Abschlussbericht MINT)
- SCORE DAI
  - Wartungsaufwand nicht im Projekt untersucht, da verquickt mit Entwicklungsaufwand
  - laut FZI/andrena vergleichbar mit oAW

- Auszug aus den Vergleichsmetriken zur Performance von Chartproduction
- Metrik 2.1.1: Zeit um die Anwendung zu initialisieren
- Metrik 2.1.2: Der initiale Speicherverbrauch nachdem die Anwendung komplett gestartet wurde

Metrik[Einheit]	ADO	NHibernate	OpenAccess	SCORE
M2.1.1[ms]	5331,3	6211,2	6961,4	<b>1462,5</b>
M2.1.2[MB]	<b>51,8</b>	56,1	51,9	52,9

- Metrik 2.2.3a: Zeit welche für eine spezifische Aufgabe benötigt wird
- Metrik 2.2.3b: Durchsatz Aufgaben pro Minute
- Metrik 2.2.4: Maximale Speichernutzung während der Ausführung des Szenarios

Metrik[Einheit]	ADO	NHibernate	OpenAccess	SCORE
M2.2.3a[ms]	87,2	194,0	114,8	<b>84,1</b>
M2.2.3b[1/m]	11,5	5,15	8,7	<b>11,9</b>
M2.2.4[MB]	<b>2,2</b>	6,5	24,0	2,3

- Metrik 2.4.1: Antwortzeit der Methoden der generischen Geschäftslogikfassade, gemessen pro Methode mit wechselndem Benutzungsprofil
- Hier: Summe über alle Methoden der ChartProduction-Fassade

Metrik[Einheit]	ADO	NHibernate	OpenAccess	SCORE
M2.4.1[ms]	2162,8	3403,9	2559,0	<b>2017,2</b>

- Auszug aus den Vergleichsmetriken zur Performance von Accounting
- Metrik 2.2.3a: Ausführungszeit des Gesamtszenarios
- Hier: komplexe Lese-Schreib-Aktionen

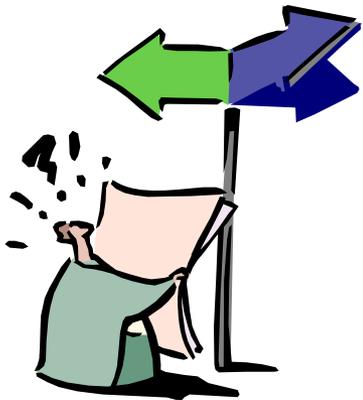
Metrik[Einheit]	ADO	NHibernate	OpenAccess	SCORE
M2.2.3a[ms]	72077,4	113328,0	79938,8	<b>63493,6</b>

- oAW: Verzicht auf Test, da Code weitgehend identisch zur manuellen NHibernate-Implementierung
- OR-Frameworks sind projektspezifischen Adaptern (ADO.NET, SCORE DAI) klar unterlegen

- SCORE DAI:

- Generierte Klassen sind Oracle-spezifisch **und** applikationsspezifisch
- Applikationsspezifische Optimierung von Queries und Datenstrukturen
- Verwendung Oracle-spezifischer ADO.NET-Klassen

Konzepte für Persistenzadapter



SCORE<sup>®</sup> Adaptive Bridges –  
Data Architecture Integration<sup>™</sup>

MINT – Adapter im Wettbewerb

**Zusammenfassung**

- Was manuell effizient programmierbar ist, kann auch generiert werden
  - + Zuverlässigkeit
  - + Konservierung von Expertenwissen
- Anspruchsvolle Generierungsaufgaben verlangen nach MDD-Techniken, um wirtschaftlich zu sein ...
- ... oder: wenn man Generatorbau als eigene Entwicklungsdomäne begreift, kann man leistungsstarke Generatoren bauen!
- → HyperSenses™

- SCORE DAI = DSL für Persistenzadapter
  - Plattformunabhängigkeit: von Cobol, PL/1 bis Java, C#, ..., sowie verschiedenste DBMS
- SCORE DAI = Generator für Persistenzadapter
  - Wirtschaftlichkeit, Effizienz, Wartbarkeit
- Dienstag, 15.15 Uhr:  
„Automatisch und fehlerfrei – eine Vision?“ (R. Schilling)
- [www.D-S-T-G.com/MDD2009](http://www.D-S-T-G.com/MDD2009)

Vielen Dank für Ihre Aufmerksamkeit!  
Gibt es noch Fragen?