

D E L T A



software
technology

SCORE[®]

DATA ARCHITECTURE INTEGRATION[™]

Effiziente Entwicklung hochperformanter Daten-Services

*Persistency Layers im Vergleich:
Die besten Entwicklungsverfahren und Werkzeuge*

*Das vom Bundesministerium für Bildung und Forschung (BMBF) geförderte Projekt „MINT – Modellgetriebene **I**ntegration von Informationssystemen“ untersuchte Konzepte und Werkzeuge zur modellgetriebenen Kopplung moderner, objektorientiert modellierter Business-Logik mit relationalen Datenbanksystemen.*

MODELLGETRIEBENE INTEGRATION VON INFORMATIONSSYSTEMEN DAS PROJEKT „MINT“

Wie können Softwaresysteme kosteneffizient an sich ändernde Geschäftsprozesse und neue Anforderungen angepasst werden?

Um diese Frage zu beantworten, wurden im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts „MINT – Modellgetriebene Integration von Informationssystemen“ Entwicklungsverfahren für die modellgetriebene Kopplung moderner, objektorientiert modellierter Business-Logik mit bestehenden relationalen Datenbanksystemen untersucht und bewertet.

Hier stellen wir ausgewählte Ergebnisse vor.

ARCHITEKTUR DER TESTUMGEBUNG

Alle Applikationen verwenden die gleiche verallgemeinerte Testumgebungsarchitektur. Sie konzentrieren sich auf verschiedene Nutzungsszenarien, um die konkurrierenden Persistenzsysteme vergleichen zu können. Die Veränderung liegt einzig in der Persistenzschicht, die die bestimmte Persistenztechnologie und den spezifischen Persistenzadapter enthalten. Die anderen drei Schichten bleiben für die Applikationen unverändert. Jede Technologie muss die gleiche Funktionalität zur Verfügung stellen, um vergleichbare Merkmale zu bekommen.

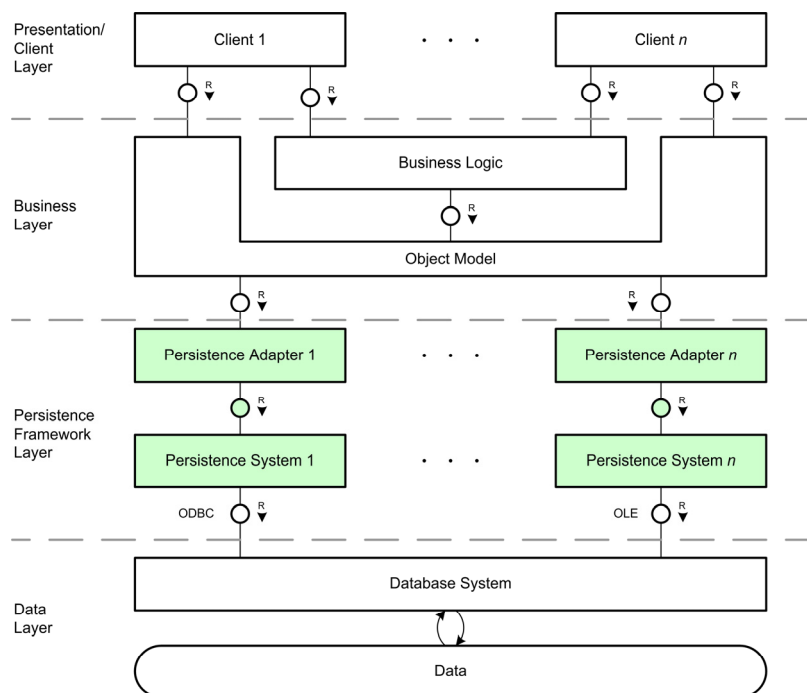


Abb: Architektur der Testumgebung

ADAPTER-KONZEPTE

Die Kopplung, insbesondere das objekt-relationale Mapping, wurde durch unterschiedliche Adapter-Konzepte realisiert.

- Generische Adapter unterstützen alle Testapplikationen, sind jedoch spezifisch für das eingesetzte DBMS sowie das DB-Schema. Sie sind datenzentriert und implementieren ein dem DB-Schema entsprechendes Objektmodell.
- Statische Lösungen, die für das konkrete Objektmodell entwickelt wurden
- Dynamische Lösungen, die ein Laufzeit-Mapping durchführen und entsprechend konfiguriert werden.
- Generierte Adapter implementieren im Gegensatz dazu ein Mapping, das spezifisch für die jeweilige Applikation ist. Sie sind also anwendungsorientiert.

Testapplikationen für die unterschiedlichen Datenzugriffsvarianten ermöglichten einen aussagekräftigen Vergleich der sehr verschiedenen Adapter-Konzepte sowie Aussagen über Stärken und Schwächen der einzelnen Lösungen.

BESTE ERGEBNISSE FÜR SCORE DATA ARCHITECTURE INTEGRATION

Die mit SCORE Data Architecture Integration der Delta Software Technology vollständig automatisch generierten Adapter stellen eine der untersuchten Lösungsalternativen dar. Die Abbildung der Service Interfaces auf die Datenobjekte wurde dabei von SCORE maßgeschneidert für die Applikationen in C# implementiert, spezifisch für den verwendeten Datenbankausschnitt und das eingesetzte DBMS.

Die Tests erbrachten für die mit SCORE Data Architecture Integration generierten Adapter den Nachweis hoher Performance, geringer Entwicklungskosten, absoluter Zuverlässigkeit des generierten Codes sowie drastisch reduzierten Testaufwands.

TESTSZENARIEN

Für die Vergleichstests wurden C#-Anwendungen aus einer kommerziellen Banken-Software zur Unternehmens- und Aktienanalyse genutzt:

- Chart Production
 - Anzeige von Aktienkursen (Chart-Diagramme)
 - Nur lesender Zugriff, selektiv und batch
- Accounting
 - Unternehmensdatenverwaltung
 - Lesende und schreibende Zugriffe, selektiv
 - Import von Aktienkurs-Daten

Für den Zugriff auf die in einer Oracle-Datenbank gespeicherten Realdaten wurde ADO.NET genutzt.

ADAPTER IM WETTBEWERB

Für den Vergleich wurden Adapter mit Werkzeugen unterschiedlicher Lösungskonzepte bereitgestellt:

- Manuell implementierte Adapter mit ADO.NET-Zugriffstechnologie (nachfolgend bezeichnet als „ADO“)
- Persistenz-Frameworks mit Konfiguration des Runtime-O/R-Mappings (generische Lösung):
 - Telerik OpenAccess („OpenAccess“)
 - NHibernate („NHibernate“)
- Generierte Adapter
 - „Out-of-the-box“ Code-Generator
 - AndroMDA
 - Modulares Generator-Framework (speziell zur Entwicklung individueller Modelltransformationen)
 - openArchitectureWare („oAW“)
 - Modell-getriebene Adapter-Generierung (maßgeschneidertes Mapping für die jeweilige Applikation, spezifisch für den verwendeten Datenbankausschnitt und das eingesetzte DBMS)
 - SCORE Data Architecture Integration („SCORE“)

1. DISZIPLIN: ENTWICKLUNG UND WARTUNG

Die angeführten Ergebnisse wurden bei der Wartung und Erstellung der Adapter für die Anwendung „Accounting“ gewonnen.

	ADO	NHibernate	OpenAccess	oAW/SCORE
Implementation (PT)	4,2	3	3,25	0,9
Test und Debug (PT)	8,1	3,2	4,2	1,5

* PT = Personentage

Einzelbewertung:

- Out-of-the-box-Generator AndroMDA:
Schwierigkeiten bei der Abbildung der Datenbank und bei der Anpassung an Interfaces
Fazit: Out-of-the-box-Generatoren sind nur bedingt für das O/R-Mapping für Business-Logik <-> DBMS geeignet.
- Manuell implementierter Adapter („ADO“):
Höchste Aufwände
- Persistenz-Frameworks NHibernate und OpenAccess:
Gute Ergebnisse bei Migrations- und Evolutionsszenarien
Workarounds notwendig
- Generierte Adapter:
oAW: Gute Werte, Einsatz modell-getriebener Techniken verkürzt die Implementierung zusätzlich
SCORE: Ergebnisse vergleichbar mit oAW

2. DISZIPLIN: PERFORMANCE

Für die Anwendung „Chart Production“ wurden folgende Performance-Werte gewonnen:

	ADO	NHibernate	OpenAccess	SCORE
Initialisierung der Anwendung	5.331,3 ms	6.211,2 ms	6.961,4 ms	1.462,5 ms
Initialer Speicherverbrauch nach vollständigem Start der Anwendung	51,8 MB	56,1 MB	51,9 MB	52,9 MB
Zeitbedarf für eine ausgewählte Aufgabe	87,2 ms	194,0 ms	114,8 ms	84,1 ms
Durchsatz Aufgaben pro Minute	11,5/min	5,15/min	8,7/min	11,9/min
Max. Speichernutzung während Ausführung des Szenarios	2,2 MB	6,5 MB	24,0 MB	2,3 MB
Summe der Antwortzeiten aller Methoden der Chart Production-Fassade	2.162,8 ms	3.403,9 ms	2.559,0 ms	2.017,2 ms

Ausführungszeiten des gesamten Szenarios für „Accounting“:

	ADO	NHibernate	OpenAccess	SCORE
Ausführungszeit des Gesamt-Szenarios (komplexe Lese-Schreib-Aktionen)	72.077,4 ms	113.328,0 ms	79.938,8 ms	63.493,6 ms

Ergebnis:

Es wird deutlich, dass projektspezifische Adapter (vollständig automatisch generiert: SCORE, oder manuell erstellt: ADO) den generischen O/R-Frameworks (OpenAccess, NHibernate) klar überlegen sind.

Die mit SCORE Data Architecture Integration generierten Adapter überzeugten mit geringen Entwicklungskosten, Zuverlässigkeit des generierten C#-Codes, drastisch reduziertem Testaufwand und hoher Performance.

ZUSAMMENFASSUNG

Die Delta-Lösung, die Persistenzadapter maßgeschneidert zu generieren für die jeweilige Applikation und spezifisch für den verwendeten Datenbankausschnitt und das eingesetzte DBMS, sorgt für hohe Performanz und geht einher mit einer hohen Zuverlässigkeit des generierten Quellcodes sowie einem drastisch reduzierten Testaufwand. Die durchgehende Werkzeugunterstützung senkt den mit dem (genauer: jedem) Werkzeugeinsatz verbundenen Overhead – und damit die Entwicklungskosten – entscheidend ab.

Das Entwicklungsverfahren von SCORE Data Architecture Integration hat klar gepunktet:

- Die generierten Klassen sind *DB-spezifisch* (in diesem Fall Oracle) **und applikations-spezifisch**
- Die Queries und Datenstrukturen sind applikationsspezifisch optimiert
- Die ADO.NET-Klassen sind hochperformant, weil sie spezifisch für die tatsächlich benötigten Daten- und Zugriffsstrukturen erzeugt wurden

Hervorzuheben ist, dass bei der Adapter-Erstellung mit SCORE sowohl das Objektmodell als auch die Testapplikationen unverändert blieben. Die generierten Anwendungsadapter wurden im Projekt vollständig aus den Daten des Composition Repository generiert. Der generierte Code wurde in einer (menschlich) gut lesbaren Form erzeugt.

Das MINT-Projekt unterstreicht die Verwendbarkeit von SCORE für unterschiedlichste Umgebungen.

Die Tests haben erneut belegt:

- Was manuell **effizient** programmierbar ist, kann auch **generiert** werden!

Automation durch generative Entwicklung bringt:

- Zuverlässigkeit: Entwicklungs- und Wartungsaufwand werden reduziert.
- Einfache, konsistente und nachhaltige Nutzung von Expertenwissen

WEITERE INFORMATIONEN

- <http://www.d-s-t-g.com>
- <http://www.mint-projekt.de/>

Der ausführliche Projektbericht wurde als Buch veröffentlicht:
„MINT – Modellgetriebene Integration von Informationssystemen:
Projektbericht“, Hrsg. Prof. Ralf Reussner, Karlsruhe, 2009
Die Testwerte sind dem genannten Projektbericht entnommen.

DELTA



software
technology

SCORE DATA ARCHITECTURE INTEGRATION

Daten als echte Business Services:

Schnell, einfach und unabhängig von Datenarchitekturen und Speicherungsformen.

SCORE Data Architecture Integration liefert genau das, was benötigt wird – Daten als echte Business-Services, vollständig generierter, sauber gekapselter Zugriffscode zur Bearbeitung von Informationen, ohne Pseudo-SQL-Interfaces.

Die Arbeit mit SCORE ist einfach. SCORE stellt sicher, dass auch die komplexesten, heterogenen Datenplattformen einfach verwendet werden können. Selbst unterschiedliche TPMS und Sprachen werden unterstützt, ohne dass die Service-Nutzer die Besonderheiten der jeweiligen Umgebung kennen oder berücksichtigen müssen:

Für den Service Consumer gibt es keinen Unterschied zwischen Daten, Funktionen oder einer Kombination aus beidem – ein Service ist ein Service!

WWW.D-S-T-G.COM