

Model Driven Architecture am Beispiel von zwei Entwicklungsprojekten

Hamburg, 4. September 2007

Christian Lurk-Güting, evodion Information Technologies GmbH
Jörg Gerdes, Bundesamt für Seeschifffahrt und Hydrographie

- **Das Kundenumfeld**
- Was ist MDA?
- MDA von Anfang an
- MDA bei laufenden Projekten
- Fazit



BSH „maritimer Dienstleister“ des Bundes

Bundesoberbehörde im
Geschäftsbereich des Ministeriums für
Verkehr, Bau und Stadtentwicklung

Historie:

1868 – Norddeutsche Seewarte

1875 – Deutsche Seewarte

1945 – Deutsches Hydrographisches Institut

1990 – Bundesamt für Seeschifffahrt und
Hydrographie



- Zwei Dienstsitze in Hamburg und Rostock
- zirka 900 Beschäftigte
- Fünf Schiffe zur Vermessung, Wracksuche, Überwachung und Forschung
- Maritime Fachbibliothek (160.000 Bände)
- Rechenzentrum (BSH / MaAGIE)
- Druckerei
- Chemisches Labor
- Versuchswerkstatt

- Etat von ca. 60 Mio. Euro



Seeaufgabengesetz

- S Sicherheit der Seeschifffahrt und Gefahrenabwehr bei Schiffen
- N Bereitstellung nautischer und hydrographischer Informationen
- M Meereskundliche Informationen und Dienstleistungen
- M Maritime Raumplanung
- M Schutz der Meeresumwelt

- Z Meeresnutzung, Verwaltung



- Wasserstandsvorhersagen
- Sturmflutwarnungen
- Gezeitenvorausberechnung
- Überwachung der Meeresumwelt
(Messnetz, Wasserqualität, Fernerkundung, ...)
- Baumusterprüfungen und Genehmigung von Navigations- und Funkausrüstung
- Schiffsvermessung
- Schifffahrtförderung, Zeugnisse für Seeleute
- Offizielle Seekarten (analog/digital)



- Zentraler IT – Dienstleister für das BSH
 - 37 Mitarbeiter (vom Telefon bis zum HPC)
- Angebote für die Behörden der BVBS
 - Rechnungswesen DWD
 - Anwendungshosting für KBA, BBR
- MaAGIE - Rechenzentrum (SAP/Oracle/ESRI)
- Vom Rechenzentrum zum ...
 - Vom Mainframe zu offenen Systemen
 - ISO9001 zertifiziert
 - BSI Grundschutz
 - Nach ITIL organisiert ...



- ca 1200 Desktop-Systeme
- ca 20 Midrange-Serversysteme (SUN/IBM)
z.B. Modellbetrieb auf IBM pSeries 570
(8 x Power5, 32GB)
- ca 40 x86 Serversysteme (MS/Linux/Solaris)
- ca 80 Spezialsysteme (HW) für Labor- & Messeinrichtungen
- 20 TB SAN + 5 TB DAS
- Ca 250 TB Kapazität Archivierung (HSM) u. Backup

- Eigenentwicklungen im wissenschaftlichem Bereich und DB-GUI
- Ausrichtung auf Standardanwendungen, RDMS Oracle, ERP SAP, ...



- 7 * 24h Betrieb
- 100 % Verfügbarkeit für ausgewählte Anwendungen
- Datenarchivierung über große Zeiträume
- Wartbarkeit der Anwendungen, extern abgesichert aber unabhängig
- Portabilität, heterogenes Umfeld



IT - Strategie der BVBS

- Standardisierung
 - Offene Standards wenn möglich, Marktstandards wenn nötig
- Zentralisierung ?
 - Nutzung von Infrastruktur und Know-how über Organisationsgrenze, Servicecenter

Ziele

- Optimierung der Nutzerunterstützung
- Verstärkte Nutzung von Virtualisierung
 - Verbessertes Fehlermanagement
 - Optimierung der HW-Ausnutzung



- Das Kundenumfeld
- **Was ist MDA?**
- MDA von Anfang an
- MDA bei laufenden Projekten
- Fazit



Vorstellung evodion IT

Gründung	1983
Geschäftssitz	Hamburg
Geschäftsführer	Wolfgang Wulf, Frank Hunold
Mitarbeiter	35
Consulting	Geschäftsprozessanalyse und –optimierung Architektur- und IT-strategische Beratung Erstellung von Konzepten und Studien
Lösungen	Individuelle Softwarelösungen Interaktive Websites und Portallösungen Integration und Migration von IT-Systemen



- Motivation:
 - Reduzierung von „Routearbeiten“
 - Konzentration auf die Fachlichkeit
 - Reduzierung der Komplexität eines Systems
 - Vermeidung von Inkonsistenzen im System
 - Standardisierung (fachlich, technisch)
 - Kosteneinsparungen durch Wiederverwendung
 - Investitionsschutz, Erweiterbarkeit des Systems
- Ziele:
 - Steigerung der Entwicklungsgeschwindigkeit
 - Bessere Handhabbarkeit aufgrund höherer Abstraktion (höhere Stabilität und Qualität)
 - Hersteller- und Plattformunabhängigkeit
 - Sprach- und Systemunabhängigkeit



- Model Driven (Software) Development
 - Vorgehensmodell für die Entwicklung von komplexer Software bzw. IT-Systemen
 - ein System wird durchgängig mit Hilfe von Modellen beschrieben (Abstraktionshierarchie)
 - Trennung von geschäftsrelevanten und technischen Informationen
 - definierter Softwareentwicklungsprozess
 - Überführung des fachlichen in das technische Modell durch (mehrstufige) Transformation
- Model Driven Architecture
 - ist ein spezieller Ansatz des MDD
 - Automation durch Formalisierung
 - Aus formal eindeutigen Modellen soll durch Generatoren automatisch Code erzeugt werden.
 - Nutzung einer Modellierungssprache
 - i.d.R. UML, XML



- Trennung von Funktionalität und Technik
- Verwendung von Modellen und Generatoren
- schrittweise Transformation der Modelle
- Generierung von Quellcode
- sinnvoller Anteil an Automatisierung
 - je nach System 20-80%
- vollständige Kontrolle während des Entwicklungsprozesses



- Unterteilung des Gesamtmodells in mehrere Schichten
 - Computation Independent Model (CIM)
 - die umgangssprachliche Beschreibung
 - Platform Independent Model (PIM)
 - Beschreibung der Abläufe und Beziehungen
 - plattformunabhängig (fachliches Modell)
 - Platform Specific Model (PSM)
 - Abbildung der Fachlichkeit je nach Art der Ziel-Laufzeitumgebung (Architektur, Services)
 - plattformabhängig (technisches Modell)
 - ein Codemodell
 - die jeweilige Zielplattform
- die Schichten bauen aufeinander auf



- Transformation der Modelle
 - Modelltransformation
 - von einem Modell in ein anderes Modell
 - Codetransformation
 - von einem (technischen) Modell in Code
- Die Transformationen erzeugen aus den Elementen des Quellmodells die Elemente des Zielmodells (Generierung).
- Die Transformation schließt auch manuelle Ergänzungen für das Zielmodell mit ein.
- Die Transformation geschieht üblicherweise von der abstrakteren Ebene in die konkretere Ebene (CIM-PIM-PSM-Code).
- Transformation durch Generatoren
 - Standardprozesse (z.B. XSLT)
 - selbst erstellte Programme



- Rahmenstruktur, Programmiergerüst
- vereinheitlicht den Ablauf innerhalb einer Anwendung
 - fachlich (Benutzeroberfläche)
 - z.B. Auswahlmenü, Suchmaske, Einzelsatzmaske
 - technisch (Implementierung)
 - Mehrschichtarchitektur
- bietet definierte Schnittstellen für spezielle Funktionalität (Einstiegspunkte)
 - Umkehrung der Kontrolle beim Aufruf
 - Don't call the framework, the framework calls you
- unterstützt die Codegenerierung
 - z.B. Standardoberflächen, Datenzugriffsschicht
- nicht expliziter Bestandteil der MDA, aber sinnvolle und praktische Erweiterung

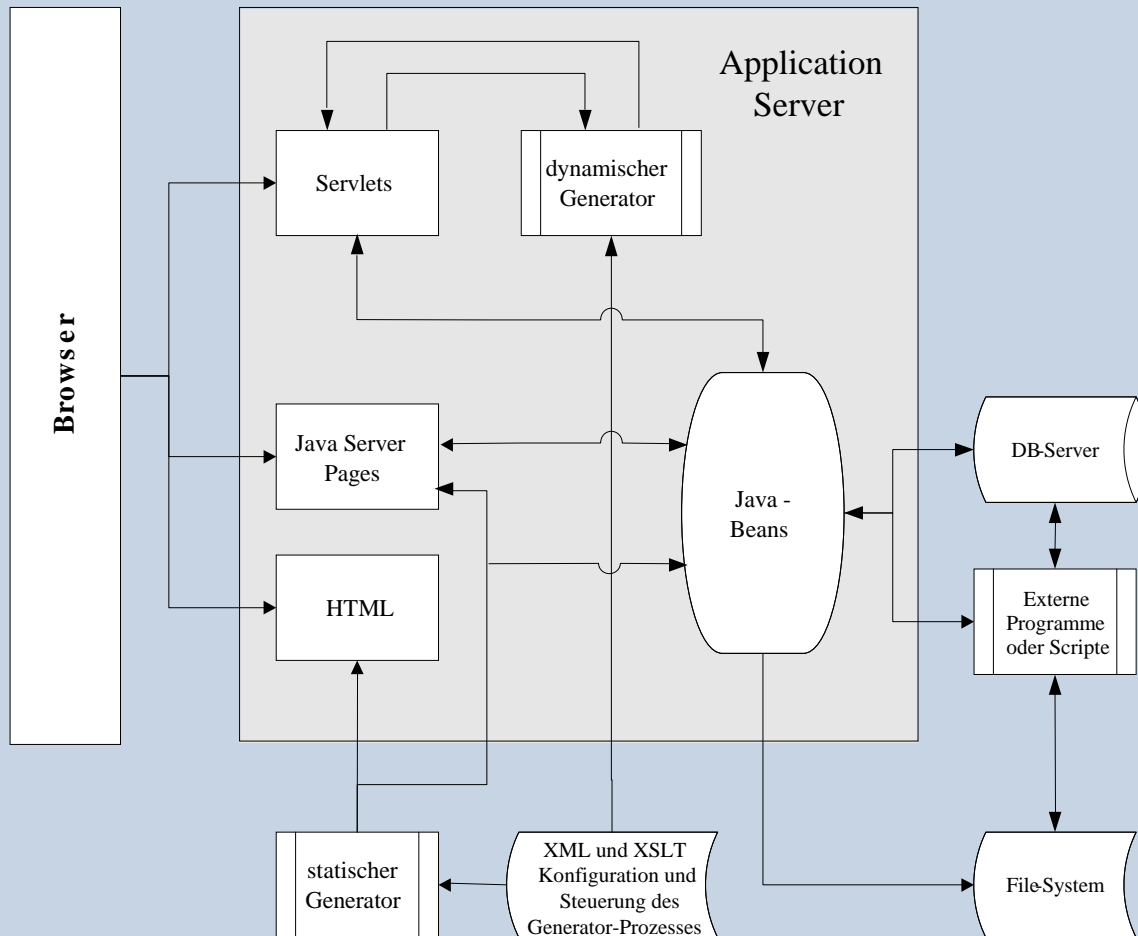


- Das Kundenumfeld
- Was ist MDA?
- **MDA von Anfang an**
- MDA bei laufenden Projekten
- Fazit

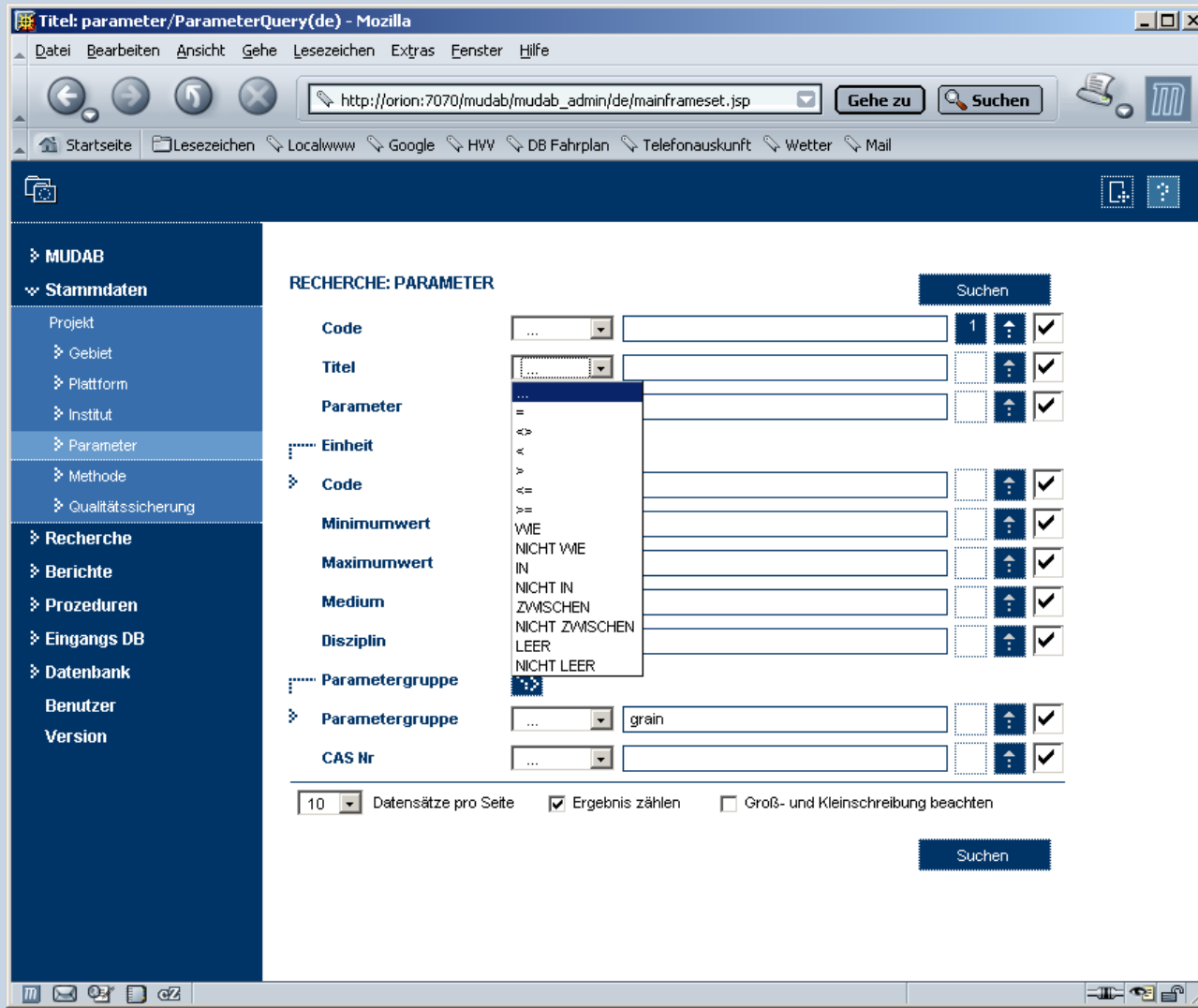


- entstanden im Rahmen eines umfangreichen Projektes zur Migration mehrerer Datenbank-Fachanwendungen von Ingres nach Oracle
- web-basiertes Anwendungs-Framework
- Datenmodellierung mit Power Designer
- Modellgetriebene Architektur mit
 - Modelltransformation
 - Code-Erzeugung (statisch, dynamisch)
- entwickelt von 2001-2005
 - Umsetzung von 5 Fachinformationssystemen
- seit 2006 Pflege und Wartung
 - Umsetzung von 3 weiteren Fachanwendungen





Benutzeroberfläche - Menü und Suchmaske



MUDAB

- Stammdaten
 - Projekt
 - Gebiet
 - Plattform
 - Institut
 - Parameter
 - Methode
 - Qualitätssicherung
- Recherche
- Berichte
- Prozeduren
- Eingangs DB
- Datenbank
 - Benutzer
 - Version

RECHERCHE: PARAMETER

Suchen

Code	[...]	[]	1	↑	↓	✓
Titel	[...]	[]	[]	↑	↓	✓
Parameter	[...]	[]	[]	↑	↓	✓
Einheit	[...]	[]	[]	↑	↓	✓
Code	[...]	[]	[]	↑	↓	✓
Minimumwert	[...]	[]	[]	↑	↓	✓
Maximumwert	[...]	[]	[]	↑	↓	✓
Medium	[...]	[]	[]	↑	↓	✓
Disziplin	[...]	[]	[]	↑	↓	✓
Parametergruppe	[...]	grain	[]	↑	↓	✓
CAS Nr	[...]	[]	[]	↑	↓	✓

10 Datensätze pro Seite Ergebnis zählen Groß- und Kleinschreibung beachten

Suchen



Benutzeroberfläche - Recherche-Ergebnis

Titel: parameter/ParameterResult(de) - Mozilla

http://orion:7070/mudab/mudab_admin/de/mainframeset.jsp

Funktionen...

- MUDAB
 - Stammdaten
 - Projekt
 - Gebiet
 - Plattform
 - Institut
 - Parameter
 - Methode
 - Qualitätssicherung
 - Recherche
 - Berichte
 - Prozeduren
 - Eingangs DB
 - Datenbank
 - Benutzer
 - Version

Zur letzten Recherche
RECHERCHE-ERGEBNIS: PARAMETER

1-10 11-20 21-30 31-40 41-45

<input type="checkbox"/>	Code	Titel	Parameter	Einheit	Code	Minimumwert	Maximumwert	Medium	Disziplin	Parametergruppe	CAS Nr
<input type="checkbox"/>	<1	Teil < 1	Amount of sediment fraction < 0.98 um	g/kg		1	600	L	P	Grain size	
<input type="checkbox"/>	<2	Teil < 2	Amount of sediment fraction < 1.95 um	g/kg		1	750	L	P	Grain size	
<input type="checkbox"/>	<3	Teil < 3	Amount of sediment fraction < 3 um	g/kg		0	0.50	L	P	Grain size	
<input type="checkbox"/>	<4	Teil < 4	Amount of sediment fraction < 3.9 um	g/kg		1	950	L	P	Grain size	
<input type="checkbox"/>	<5	Teil < 5	Amount of sediment fraction < 5 um	g/kg		0	0.50	L	P	Grain size	
<input type="checkbox"/>	<6	Teil < 6	Amount of sediment fraction < 6 um	g/kg		0	0.50	L	P	Grain size	
<input type="checkbox"/>	<7	Teil < 7	Amount of sediment fraction < 7 um	g/kg		0	0.50	L	P	Grain size	
<input type="checkbox"/>	<8	Teil < 8	Amount of sediment fraction < 7.8 um	g/kg		1	990	L	P	Grain size	
<input type="checkbox"/>	<9	Teil < 9	Amount of sediment fraction < 9 um	g/kg		0	0.50	L	P	Grain size	
<input type="checkbox"/>	<10	Teil < 10	Amount of sediment fraction < 10 um	g/kg		0	0.50	L	P	Grain size	

10 1-10 11-20 21-30 31-40 41-45

Min/Max Leer



Benutzeroberfläche - Einzelsatzmaske

Titel: parameter/ParameterShow(de) - Mozilla

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Fenster Hilfe

http://orion:7070/mudab/mudab_admin/de/mainframes Gehe zu Suchen

Startseite Lesezeichen Localwww Google HWV DB Fahrplan Telefonauskunft Wetter Mail

MUDAB

Stammdaten

- Projekt
- Gebiet
- Plattform
- Institut
- Parameter
- Methode
- Qualitätssicherung

Recherche

- Berichte
- Prozeduren
- Eingangs DB
- Datenbank
- Benutzer
- Version

Zurück [Zum letzten Recherche-Ergebnis](#)

ÄNDERUNG: PARAMETER

Code < 1

Titel Teil < 1

Parameter(En) Amount of sediment fraction < 0.98 um

Parameter(De)

Einheit g/kg

Minimumwert 1.0

Maximumwert 600.0

Medium L

Disziplin P

Parametergruppe Grain size

CAS Nr

Schlagwort En Fraction

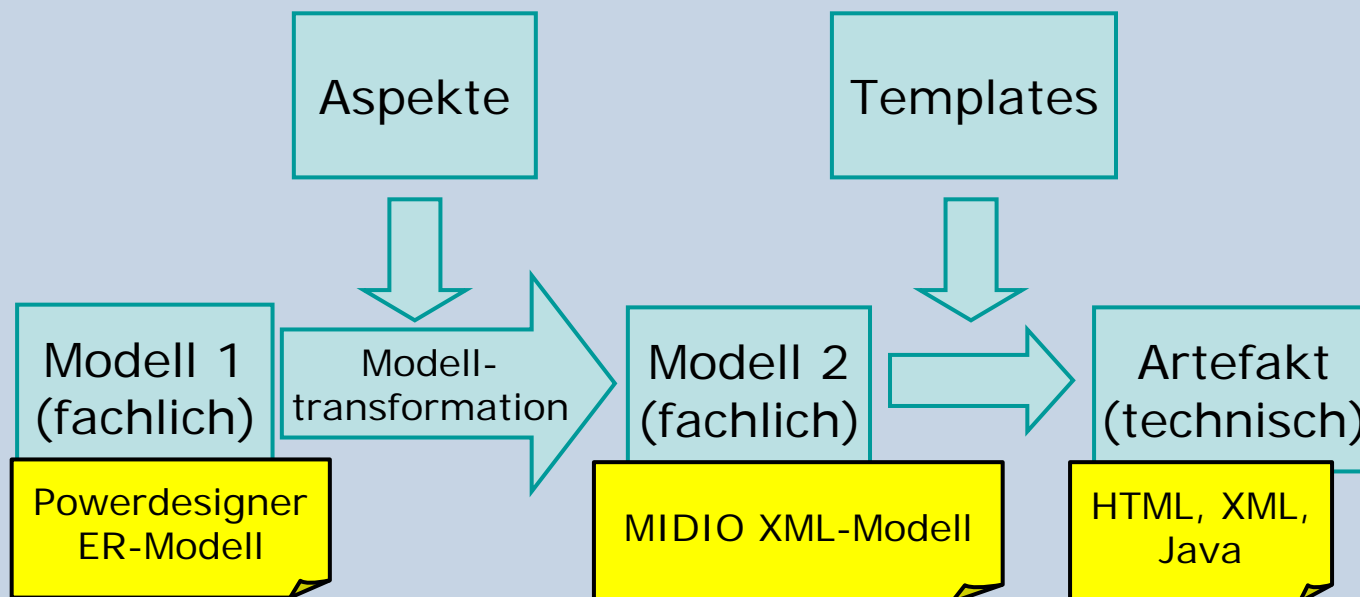
Schlagwort De Korngröße

Parameterkennung 204

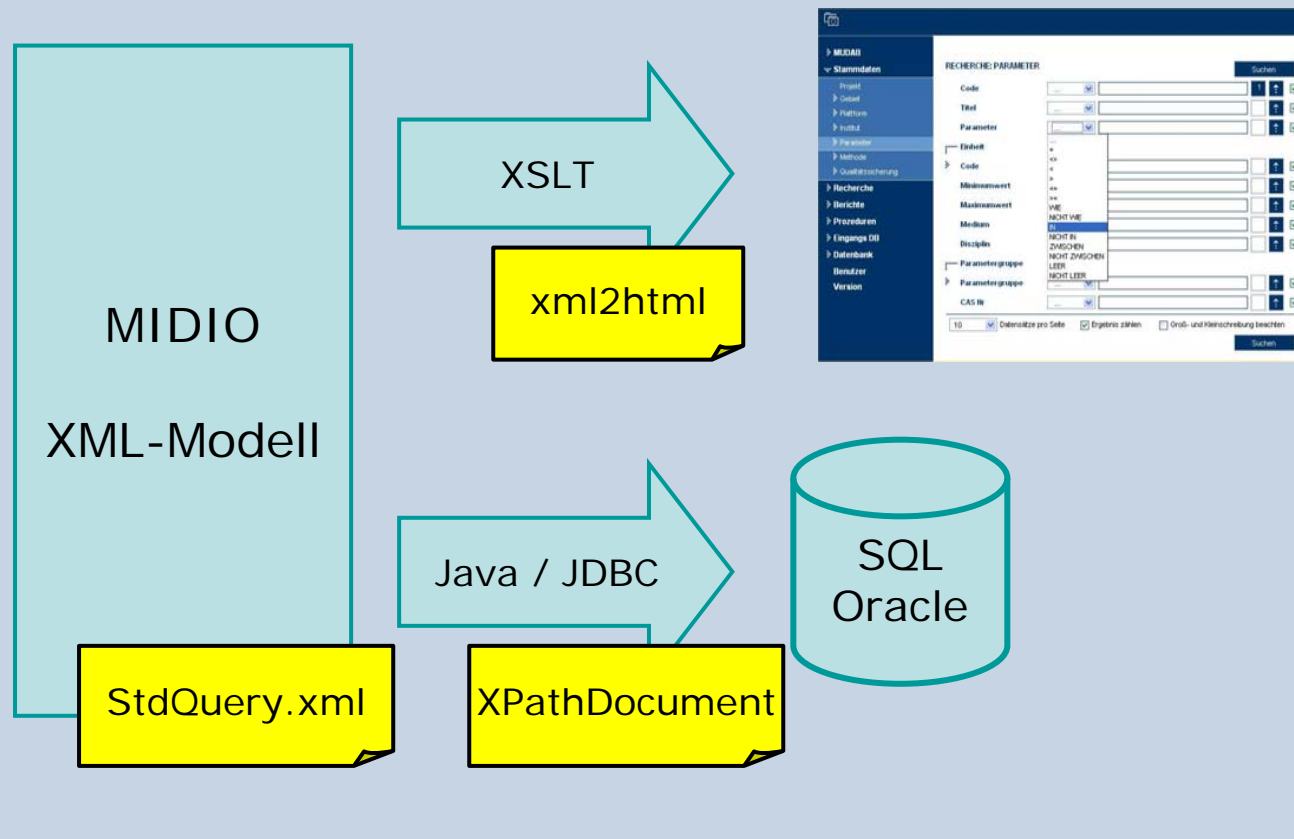
Min/Max



- Aspekte: u.a. Oberflächengestaltung
- Template: Vorlagen mit Steuerelementen



- Beispiele für Transformationen
 - XML nach HTML (Masken-Layout)
 - XML nach Java (DB-Abfragen)



- Statische Generierung
 - Basisstruktur der einzelnen Anwendungen
 - Verknüpfung der DB-Struktur mit der GUI
 - Definition der Benutzerführung
 - Menüaufbau, Maskenabfolge, Funktionsflächen
 - Definition der Rollen und Berechtigungen
 - Ausführung per ant-Aufruf (rekursiv)
- Dynamische Generierung
 - konkretes Masken-Layout
 - Einbindung der DB-Abfrage-Ergebnisse
 - marshalling, unmarshalling
 - Verlinkung der Masken („zurück“)
 - Ausführung bei entsprechender Benutzeraktion
 - im Hintergrund, Laufzeitverhalten beachten



- Problem: Mischung von generierten und von Hand erstellten Programmteilen
 - manuelle Erweiterungen dürfen beim nächsten Build nicht verloren gehen
- mehrere Möglichkeiten
 - Verschmelzung von Dateien (z.B. XML)
 - Erweiterung durch Ableitung (bei OOP)
 - Basisklasse wird generiert, manuelle Änderungen oder Erweiterungen nur in der abgeleiteten Klasse
 - getrennte Verzeichnisstruktur
 - generierte und manuell bearbeitete Dateien liegen in verschiedenen Verzeichnissen (gleicher Name)
 - beim Build werden erst die generierten Dateien in einen TEMP-Bereich kopiert und dann mit den manuell bearbeiteten Dateien überschrieben
 - geschützte Bereiche im Quellcode (proprietär)



- So weit wie möglich/sinnvoll Open Source
- Entwicklungsumgebung
 - JDK 1.5 (JSP, Servlets, Java-Klassen)
 - Ant 1.5 (Build-Tool)
 - CVS (Quellcode-Verwaltung)
 - Eclipse 3.1/3.2 (Entwicklungsumgebung)
 - Sybase Power Designer (ER-Modellierung)
- Laufzeitumgebung
 - Tomcat 4.1 (JSP-Engine)
 - Laufzeitbibliotheken
 - JDBC-Zugriff
 - XSLT-Transformationen (Saxon, Xalan)
 - Oracle 10.2 (Datenhaltung)
 - Solaris (Server-Betriebssystem)
 - WindowsXP (Client-Betriebssystem)

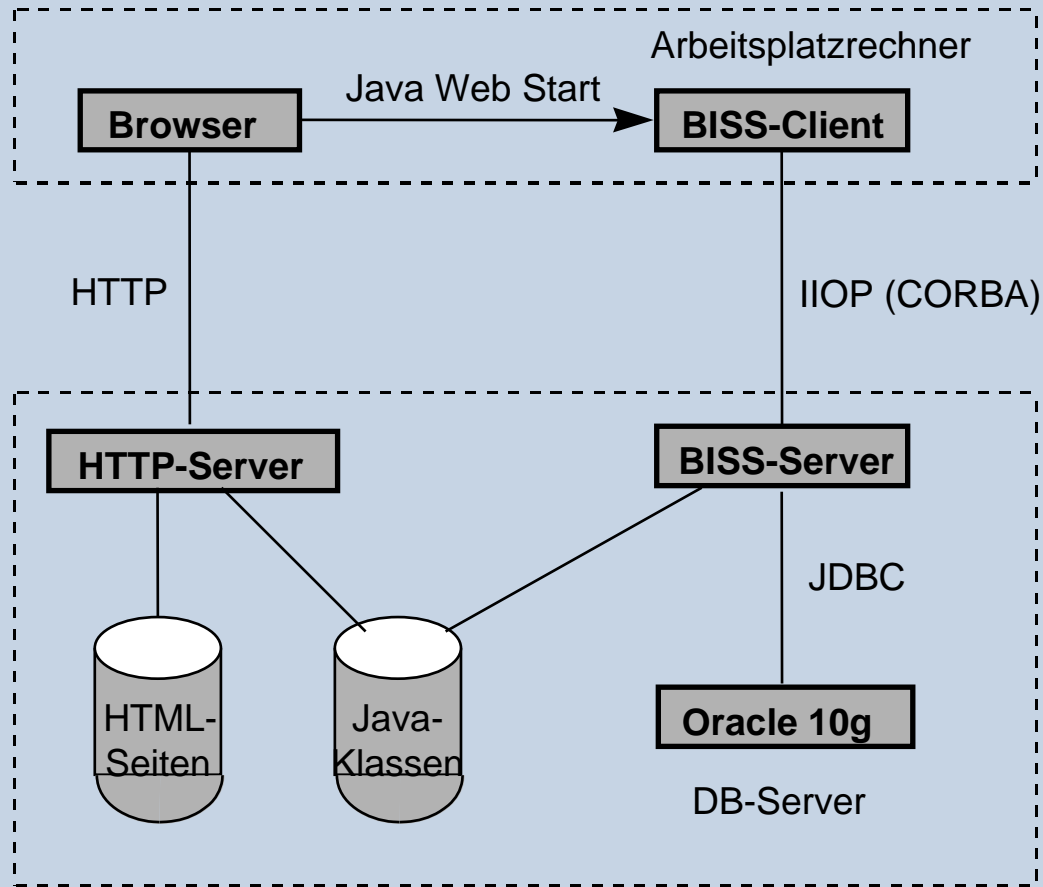


- Das Kundenumfeld
- Was ist MDA?
- MDA von Anfang an
- **MDA bei laufenden Projekten**
- Fazit



- BSH Informationssystem Schiffe
- Fachanwendung der Abteilung Schifffahrt
- Client/Server-Mehrschicht-Architektur
- über Browser aufrufbar (Java Web Start)
- Java-Anwendung mit Swing-Oberfläche
- CORBA als Middleware
 - Borland VisiBroker for Java
- Oracle 10.2 als DB-System
- nutzt ein Anwendungs-Framework
 - Java/CORBA-Factory (JCF) von evodion
- entstanden 1998/1999
- ständige Weiterentwicklung





Benutzeroberfläche - Hauptmenü



Benutzeroberfläche - Suchmaske

Institutionen Auswählen

BSH Informationssystem Schiffe (BISS)

Name (Original)	Ort (Original)	Schiffseigner	Reeder	Bereederer	Korresp.-F
BUNDESAMT FÜR SEESCHIFFFAHRT UND HYDROGRAPHIE	ROSTOCK	Nein	Nein	Ja	Nein
BUNDESAMT FÜR SEESCHIFFFAHRT UND HYDROGRAPHIE	HAMBURG	Nein	Nein	Ja	Nein
BUNDESAMT FÜR WEHRTECHNIK UND BESCHAFFUNG	KOBLENZ	Ja	Ja	Nein	Nein

Suchbegriff

Name: gleich

Ort: - Keine -

Schiffseigner: - Keine - Ausrüstungs-Hersteller: - Keine - Behörde: - Keine -

Reeder: - Keine - Antragsteller Ausrüstung: - Keine - Amtsgericht: - Keine -

Bereederer: - Keine - Zulassungsinhaber Ausrüstung: - Keine - Finanzamt: - Keine -

Korrespondent-Reeder: - Keine - Anerkannter Betrieb: - Keine - Ausstellende Behörde: - Keine -

Charterer: - Keine - Werft: - Keine - Hersteller des Antriebs: - Keine -

Antragsteller Vermessung: - Keine - Umbauwert: - Keine - Unternehmen (ISM): - Keine -

Status: Bearbeiter:



Benutzeroberfläche - Einzelsatzmaske

Institution Ändern

BSH InformationsSystem Schiffe (BISS)

Klassifizierung

<input type="checkbox"/> Schiffseigner	<input type="checkbox"/> Ausrüstungs-Hersteller	<input type="checkbox"/> Behörde
<input type="checkbox"/> Reeder	<input type="checkbox"/> Antragsteller	<input type="checkbox"/> Amtsgericht
<input checked="" type="checkbox"/> Bereederer	<input type="checkbox"/> Zulassungsinhaber	<input type="checkbox"/> Finanzamt
<input type="checkbox"/> Korrespondent-Reeder	<input type="checkbox"/> Anerkannter Betrieb	<input type="checkbox"/> Ausstellende Behörde
<input type="checkbox"/> Charterer	<input type="checkbox"/> Wert	<input type="checkbox"/> Hersteller des Antriebs
<input type="checkbox"/> Antragsteller für Vermessung	<input type="checkbox"/> Umbauwerft	<input type="checkbox"/> Unternehmen (ISM)

Name (Original)

Kurzname Straße Haus-Nr.

Postfach PlzPostfach

Postleitzahl Ort (Original)

Bundesland Land

Tel.-Nr. Fax-Nr.

E-Mail E-Mail-X400

Ansprechpartner

Zulassungsinhaber-Nr. Behörden-Code

Status Bearbeiter



- ein von evodion entwickeltes Framework
- Basisklassen für
 - die Oberflächengestaltung (suchen, bearbeiten)
 - die Client/Server-Kommunikation
 - den Datenzugriff (Business-Objekte)
- herstellerunabhängig durch Kapselung
 - der Client/Server-Kommunikation (CORBA)
 - des DB-Zugriffes (JDBC für relationale RDBMS)
- generisches DB-Mapping
 - auf Basis der Tabellen- bzw. View-Strukturen
 - ein Datensatz wird als String-Array abgebildet
 - indizierter Zugriff über Feldkonstanten



- FELD.java
 - für alle DB-Strukturen (Tables, Views)
 - Feldindizes für Array-Zugriff
 - Konstante für Array-Größe
- Institution.java (biss.bo.institution)
 - Konstante für Tabellennamen
 - Array mit Spaltennamen und Datentyp
- Pflege von Hand ...

- InstitutionenAuswahl.java
 - protected boolean setAllAttributes()
- InstitutionenEinzelsatz.java
 - protected boolean setAllFields()
 - protected boolean setAllAttributes()
- Zugriff auf Struktur-Array über Feldindex



- FELD.java
 - ... static public int INSTITUTION_NAME = 23; ...
 - static public int INSTITUTION_FELD_ANZAHL= 44;
- Institution.java (biss.bo.institution)
 - private static String dbName = "institution";
 - private static String[][] fieldDef =
{ ..., {"Name", "a"}, ... };
- InstitutionenAuswahl.java
 - InstSuchObjekt.setWert
(FELD.INSTITUTION_NAME, sql);
- InstitutionenEinzelsatz.java
 - NameText.setText(InstitutionPflege.getAttribut
(FELD.INSTITUTION_NAME));
 - InstitutionPflege.setAttribut
(FELD.INSTITUTION_NAME, NameText.getText ());



- Probleme:
 - die Datenbank und die BO-Klassen müssen bzgl. der Datenstrukturen konsistent sein
 - die BO-Klassen und FELD.java müssen bzgl. Felddefinitionen und Indizes konsistent sein
 - FELD.java enthält ALLE Tabellen- und View-Strukturen (muss überall eingebunden werden)
 - eine DB-Änderung muss an 3 Stellen von Hand gemacht werden (fehleranfällig)
- Lösung:
 - Generierung der Programmteile, die direkt aus dem DB-Schema abgeleitet werden können
 - Redesign der Architektur, indem die BO-Klassen in eine generierte und eine darauf basierende manuell erstellte Schicht aufgeteilt werden
 - alle zusammengehörenden Infos in einer Klasse



- Institution.java (biss.gen.bo.table)
 - Konstante für Tabellennamen
 - Feldindizes für Array-Zugriff
 - Konstante für Array-Größe
 - Array mit Spaltennamen und Datentyp
 - neu: wird aus dem DB-Modell generiert
- Institution.java (biss.bo.institution)
 - wenn benötigt: Pflege von Hand
 - neu: Ableitung von generierter BO-Klasse
- InstitutionenAuswahl.java (wie bisher)
- InstitutionenEinzelsatz.java (wie bisher)
- Zugriff auf Struktur-Array über Feldindex
- Verweise auf FELD ersetzen durch BO-Klasse



- Institution.java (biss.gen.bo.table)
 - private static String dbName = "INSTITUTION";
 - ... static public int NAME = 23; ...
 - static public int FELD_ANZAHL= 44;
 - private static String[][] fieldDef =
{ ..., {"Name", "a"}, ... };
- Institution.java (biss.bo.institution)
 - ... extends biss.gen.bo.table.Institution
- InstitutionenAuswahl.java
 - InstSuchObjekt.setWert (Institution.NAME, sql);
- InstitutionenEinzelsatz.java
 - NameText.setText(InstitutionPflege.getAttribut
(Institution.NAME));
 - InstitutionPflege.setAttribut
(Institution.NAME, NameText.getText ());



- MetaDataAnalyser
- von evodion erstelltes Java-Programm
- Generierung von BO-Klassen
 - für Tabellen und Views
 - nutzt die DB-Systemtabellen
 - Templates mit Steuerelementen
 - variables Datentyp-Mapping
- Ausgabe von Schema-Informationen
 - im CSV-Format
 - nutzt die DB-Systemtabellen
- über properties-Datei voll konfigurierbar
- RDBMS-unabhängig
- auch in anderen Projekten nutzbar ...



- Table-Definitionen, BO-Generierung
 - all_tables, all_tab_columns
- View-Definitionen, BO-Generierung
 - all_views, all_tab_columns
- Index-Definitionen
 - all_indexes, all_ind_columns
- Constraints-Definitionen
 - all_constraints, all_cons_columns
- Foreign-Key-Referenzen
 - all_constraints, all_cons_columns (2x)
- Tablespace-Zuordnungen
 - all_indexes, all_tables
- Triggers
 - all_triggers



- Das Kundenumfeld
- Was ist MDA?
- MDA von Anfang an
- MDA bei laufenden Projekten
- **Fazit**



- Vorteile MDA:
 - definierter Entwicklungsprozess
 - Konzentration auf die Fachlichkeit
 - weniger Inkonsistenzen (z.B. DB-Änderungen)
 - Kapselung von technischem Know-how
 - unterstützt eine iterativ-inkrementelle Vorgehensweise (Prototyping)
- Nachteile MDA:
 - z.T. sehr hoher Abstraktionsgrad (Doku !!!)
 - Abweichungen vom Standardverhalten sind aufwändig („Wildwuchs“ vermeiden!!)
 - insbesondere bei der Neuentwicklung einer MDA sind fundierte Software-Architektur-Kenntnisse erforderlich (Konzeption, Design)
- Empfehlung: pragmatische Anwendung ;-)



- Vorteile MDA:
 - Kürzere Entwicklungszeiten
 - Steigerung der Softwarequalität
 - Migrationen einfacher
- Nachteile MDA:
 - Längere Planungsphasen
 - Teilweise zu starke Abstraktion
- Vorteile überwiegen !



Quellen, Verweise, Links

Website der OMG zum Thema:

<http://www.omg.org/mda/>

MDA ist eine Strategie der Object Management Group.

Die OMG erstellt herstellerneutrale Spezifikationen zur Verbesserung der Interoperabilität und Portabilität von Software.

Die OMG fordert, dass die Spezifizierung von der Implementierung auf einer gegebenen Plattform zu trennen ist.

Einträge bei Wikipedia.de:

http://de.wikipedia.org/wiki/Modellgetriebene_Softwareentwicklung

http://de.wikipedia.org/wiki/Model_Driven_Architecture

http://de.wikipedia.org/wiki/Generative_Programmierung

<http://de.wikipedia.org/wiki/Framework>

MDA-Tools (open source, commercial):

http://www.modelbased.net/mda_tools.html

<http://www.software-kompetenz.de/?5348>

Dieses Portal wurde über ein vom Bundesministerium für Bildung und Forschung (BMBF) gefördertes Projekt von acht der führenden deutschen Forschungseinrichtungen aufgebaut.



Kontakt Daten

Ihr Kontakt:

Christian Lurk-Güting
040/399999-34
christian.lurk@evodion.de

Jörg Gerdes
040/3190-2420
joerg.gerdes@bsh.de

Bundesamt für Seeschifffahrt und Hydrographie

Bernhard-Nocht-Straße 78
D-20359 Hamburg

Telefon: + 49-40-3190-0
Telefax: + 49-40-3190-5000

E-Mail: posteingang@bsh.de
Web: www.bsh.de

evodion Information Technologies GmbH

Alter Teichweg 25A
D-22081 Hamburg

Telefon: + 49-40-399999-0
Telefax: + 49-40-399999-99

E-Mail: info@evodion.de
Web: www.evodion.de

