



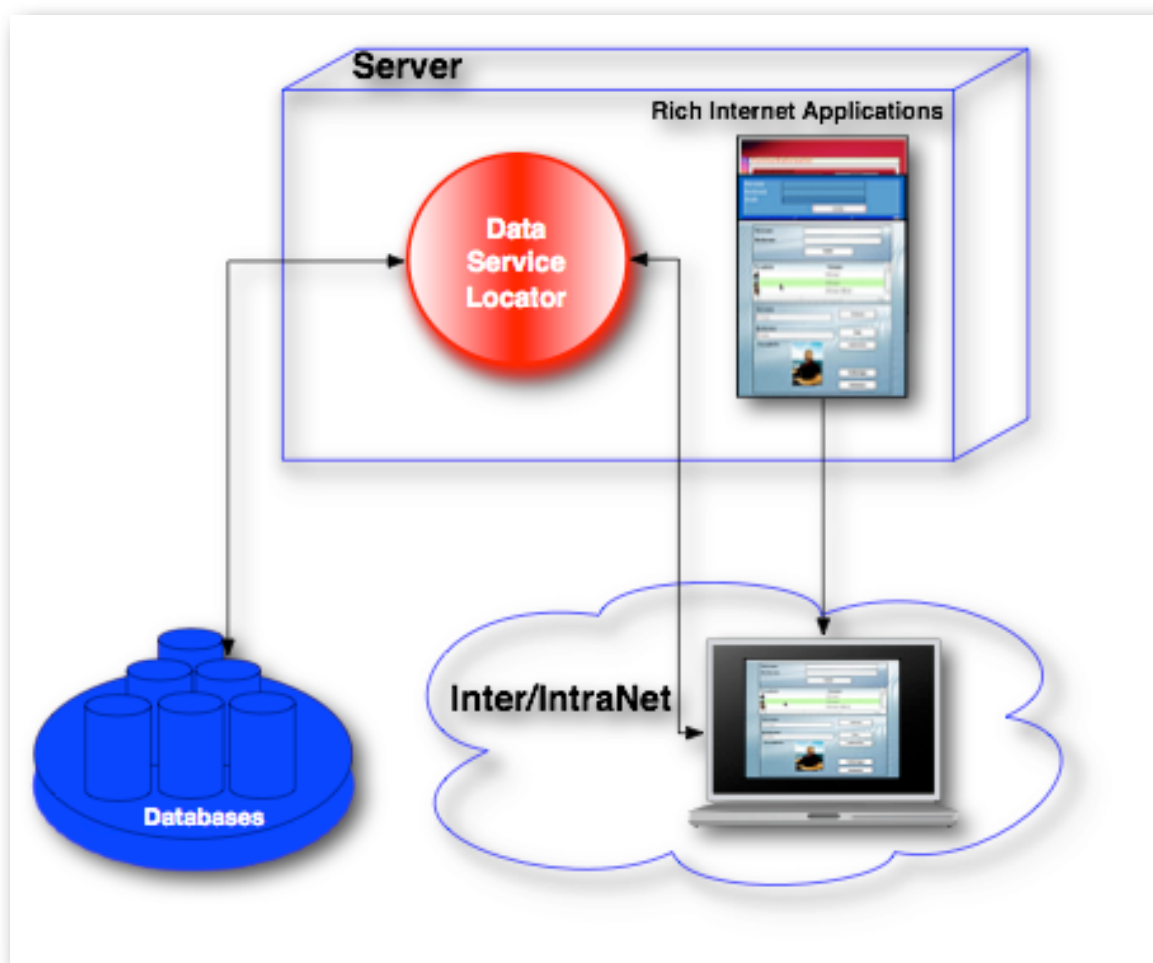
Inhalt

"Rapid RIA Development" Technologie	2
Datenbank Zugang	3
Datenbank Abstraktion	4
Komponieren statt Programmieren	5
Force4 Produkt Übersicht	5
Unterstützte Technologien	5
Arbeitsabläufe	6
Modeler	11
Composer	12
Server	13
Force4 Kernfunktionen	14
Force4 Vorteile	15
Über die A & L GmbH	16

"Rapid RIA Development" Technologie

Force4 ist ein Entwicklungssystem und Framework für das "Rapid Development" von Rich Internet Applications mit Anbindung von Datenbanken an verschiedene Präsentationstechnologien - doch was verbirgt sich dahinter? Dieses Whitepaper beschreibt die Ideen, Anforderungen und Implementationen von Force4 im Einzelnen.

Im Zuge der Entwicklung von Rich Internet Applikationen (RIAs), die herkömmliche Web Seiten durch intelligente Clients ersetzt und damit funktional reichhaltige User Interfaces bieten, wird sehr schnell deutlich, dass ein "schönes" Aussehen der Applikationen nur einer der Punkte ist, die eine RIA ausmachen. Erst durch ihre Funktionalität wird daraus eine durch den Benutzer akzeptierte Anwendung. Diese Funktionalitäten werden durch Business Logik und deren Anschluss an Backend Systeme erzeugt, in den meisten Fällen Relationale Datenbank Systeme (RDBMS).

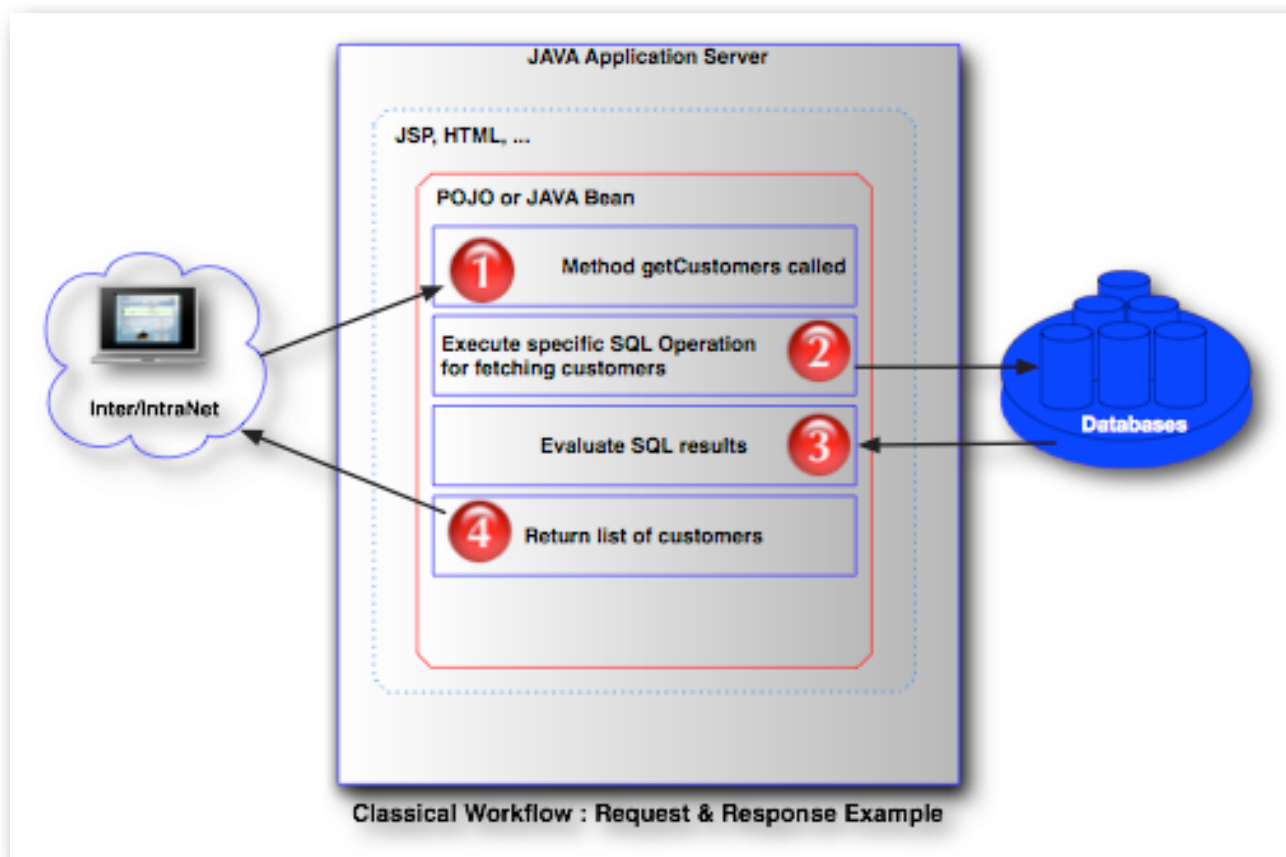


Die heutigen Präsentationstechnologien für RIAs verfügen zwar über einen mehr oder weniger rudimentären, universellen Ansatz für die Kommunikation mit Backend Systemen, dieser erfordert für die Integration von RDBMS aber einen immensen Programmieraufwand. An dieser Stelle setzt Force4 ein und bietet dem Entwickler ein Werkzeug, um schnell und ohne Programmierung sowohl den Zugriff auf Datenbank Systeme zu ermöglichen als auch deren funktionale Nutzung als vollständige Rich Internet Applikationen zur Verfügung zu stellen. Die Idee hinter Force4 war, ein Tool zu schaffen, das Web Client basierende Mutli-Tier Anwendungen erzeugen kann, das ganze ohne Programmieraufwand für den Force4 einsetzenden Entwickler auskommt. Zudem musste es möglich sein, das dieser immer die Möglichkeit hat, diese automatisch erzeugten Anwendungen nachträglich mit zusätzlicher Business Logik und Design Ele-

menten anzureichern. Force4 Applikationen können daher sowohl für Rapid Prototyping als auch für den Echteinsatz genutzt werden.

Datenbank Zugang

Der klassische Weg, um RDBMS Backend Systeme von RIAs aus zu nutzen, am Beispiel von Macromedia Flex oder Open Laszlo (beides Lösungen, die in J2EE Server eingebettet werden), ist die Benutzung von JAVA Plain Objects oder JAVA Beans. Diese werden als Business Objekte programmiert, führen bestimmte Datenbankoperationen aus und liefern die Ergebnisse, u.U. mit Business Logik angereichert, an den Web Client, z.B. den Macromedia Flash Player, zurück.



Dieser Weg ergibt sich aus den heute verfügbaren Applikations-Server Lösungen (JSP, ASP, .NET ...), da hier fast die gesamte Logik einer Web Anwendung auf der Server Seite abgehandelt wird und HTML als Präsentationlayer dient. Bekanntermassen ist HTML in seiner Funktionalität stark beschränkt ist. Zudem entspricht diese Trennung dem N-Tier Ansatz. Moderne intelligente Web Clients (Macromedia Flash, Raining Data OMNIS) erlauben jedoch, die Businesslogik ganz oder zumindest teilweise auf den Client zu verlagern da diese über eine umfangreiche Programmierbarkeit verfügen und damit ganze Applikationen selbstständig ausführen können. Der OMNIS Web Client erlaubt zudem, das Teile der Applikationen gezielt auf dem Server ausgeführt werden können ohne jedoch dabei das sonst übliche Technologie Splicing (z.B. Flex MXML/Actionscript und JAVA) zu erzeugen.

Eine Anbindung an Backend Systeme wird natürlich weiterhin benötigt um Datenbanken frequentieren zu können. Hier bieten sich sowohl "proprietäre" Wege wie Macromedias AMF oder OMNIS Binary Protokoll an als auch Webservices/SOAP. Den proprietären Wegen ist meist der Vorzug zu geben da diese für eine optimale Datenübertragung des jeweiligen Clients geschaffen wurden und einfach zu implementierende Persistenz zwischen Server Objekten und deren Client Abbildern erlaubt. Webservices dagegen sind das universellere Instrument, dafür aber deutlich weniger performant und Persistenz ist nur mit hohem Aufwand zu erreichen. Der Ein-

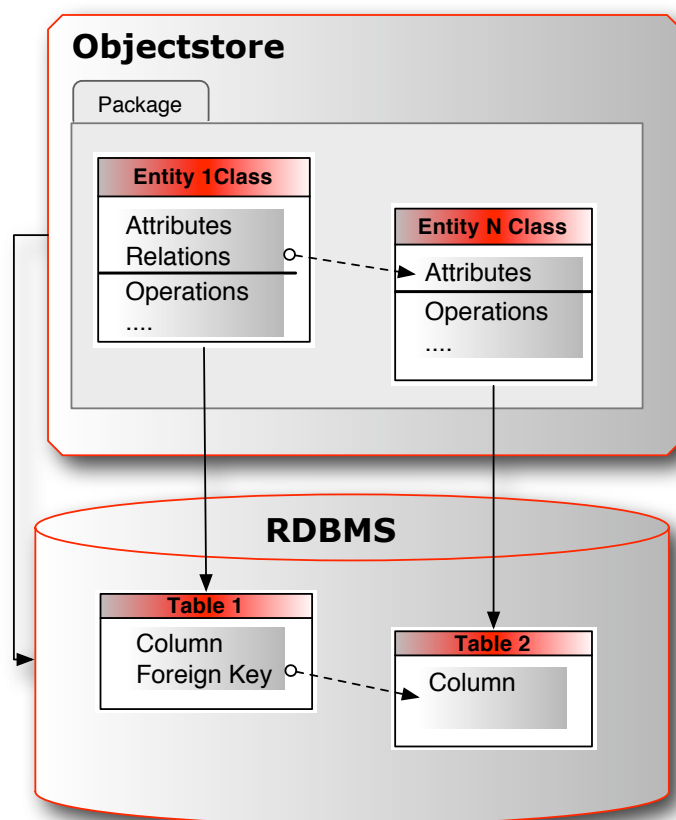
satz von HTML, JSP usw. als Träger der RIAs für eine Session-Steuerung wird hier nicht weiter betrachtet und liegt ausserhalb der Intention dieses Dokumentes da sich Force4 nur auf die RIAs konzentriert.

Eine Anforderung an Force4 war daher, eine Trennung zwischen den Web Client abhängigen Kommunikationswegen und den sonstigen, weiter unten aufgeführten, Force4 Komponenten (Persistenz Framework, Presentation Framework ...) vorzunehmen. Zur Zeit enthält Force4 (Version 2.0) eine JAVA Bridge, die von AMF basierenden Kommunikationsendpunkten (Flex AMF, OpenAMF, OpenLaszlo Java RPC) bedient wird. Die um OMNIS erweiterte Version wird dann eine entsprechende OMNIS Bridge enthalten.

Datenbank Abstraktion

In vielen Web Applikationen werden die auf dem Server stattfindenden Datenbank Operationen direkt in SQL durch die Entwickler implementiert. Dazu werden unter JAVA entweder JDBC oder JNDI, unter OMNIS die DAMs als SQL Endpunkte verwendet. Die SQL Operationen werden in die verschiedenen Business Objekten eingebaut und deren Ergebnisse an den Client weitergeleitet. Dies erzeugt statische und inflexible Server Objekte die auch bei den kleinsten Änderungen an der Datenbank Struktur, von RDBMS Systemwechseln ganz zu schweigen, jedes mal geändert und angepasst werden müssen.

In den letzten Jahren wurden nun Object/Relational Mapping (ORM) Umgebungen entwickelt, die es erlauben, Relationale Datenbanken durch Modelle dieser zu abstrahieren. Zugleich wurden universelle Frameworks geschaffen, die auf diesen Modellen aufsetzen, um ohne "Wissen" über das RDBMS und ohne direkte Programmierung von SQL, Operationen ausführen zu können. Die Modelle beschreiben Entitäten und Relationen und deren Attribute.



Daraus ergab sich für Force4 die Anforderung, den Web Clients ein Framework zu bieten, das unter Nutzung von ORM Technologie den universellen Zugriff auf Datenbanken (auch mehrere

gleichzeitig) bietet. Zudem muss Force4 in der Lage sein, die sich aus dem ORM ergebenden Objekte, die die Entitäten, Relationen u.s.w abbilden, automatisch zu erzeugen. Im Falle von JAVA müssen sogar Klassen zu erstellt werden ohne das der Entwickler zu einem JAVA IDE greifen muss. Allein das Erstellen der Objektabbildungen einer Datenbank ist ein immenser Aufwand wenn dies manuell gemacht wird. Weiterhin musste Force4 in der Lage sein, nicht nur neue Modelle zu erstellen sondern auch aus vorhandenen Datenbanken ein Model rückwärts zu entwickeln (Reverse Engineering) da der Einsatz von RIAs oft bestehende, HTML basierte, Anwendungen ablösen soll. Diese Datenbanken und deren Strukturen sind dabei bereits fest definiert.

Komponieren statt Programmieren

Die bestehenden RIA Entwicklungsumgebungen (FlexBuilder, Eclipse Plugins für Flex und OpenLaszlo, OMNIS Studio) erlauben das Platzieren und Attributieren von visuellen Containern, Controls, u.s.w. . Um der Applikation "Leben" zu geben ist zusätzlich mehr oder weniger Programmieraufwand in der Hochsprache des Web Clients erforderlich. In Datenbank basierenden Anwendungen bedeutet dies, das Queries, Ergebnishandhabung, Füllen von Master-Details Relationen, Transaktionen etc. etc. manuell programmiert werden müssen. Dies erfordert vom Entwickler, dass er die Dokumentation seiner Datenbank parallel zum Wissen über Methoden, die die Datenbank Operationen abbilden, zur Verfügung haben muss. Auch muss er selber Methoden programmieren, um Ergebnisse und Events zu handhaben. Dazu kommt das Erstellen der oben bereits angeführten Abbilder von ORM oder Business Objekten. Und natürlich müssen alle GUI Elemente der Applikation erstellt und miteinander verknüpft werden.

Die Anforderung an Force4 war daher, diese Aufgaben so weit wie möglich zu automatisieren. Dem Entwickler soll die Möglichkeit gegeben werden, mit Hilfe eines state-of-the-art GUIs alle anfallenden Aufgaben mit möglichst wenig Aufwand auszuführen. Das erfordert, dass ein Composer über das Design von visuellen Komponenten hinaus sogenannte funktionale Komponenten enthält, die z.B. das Erstellen einer Abfragemaske auf wenige Mouse-Clicks reduziert.

Force4 Produkt Übersicht

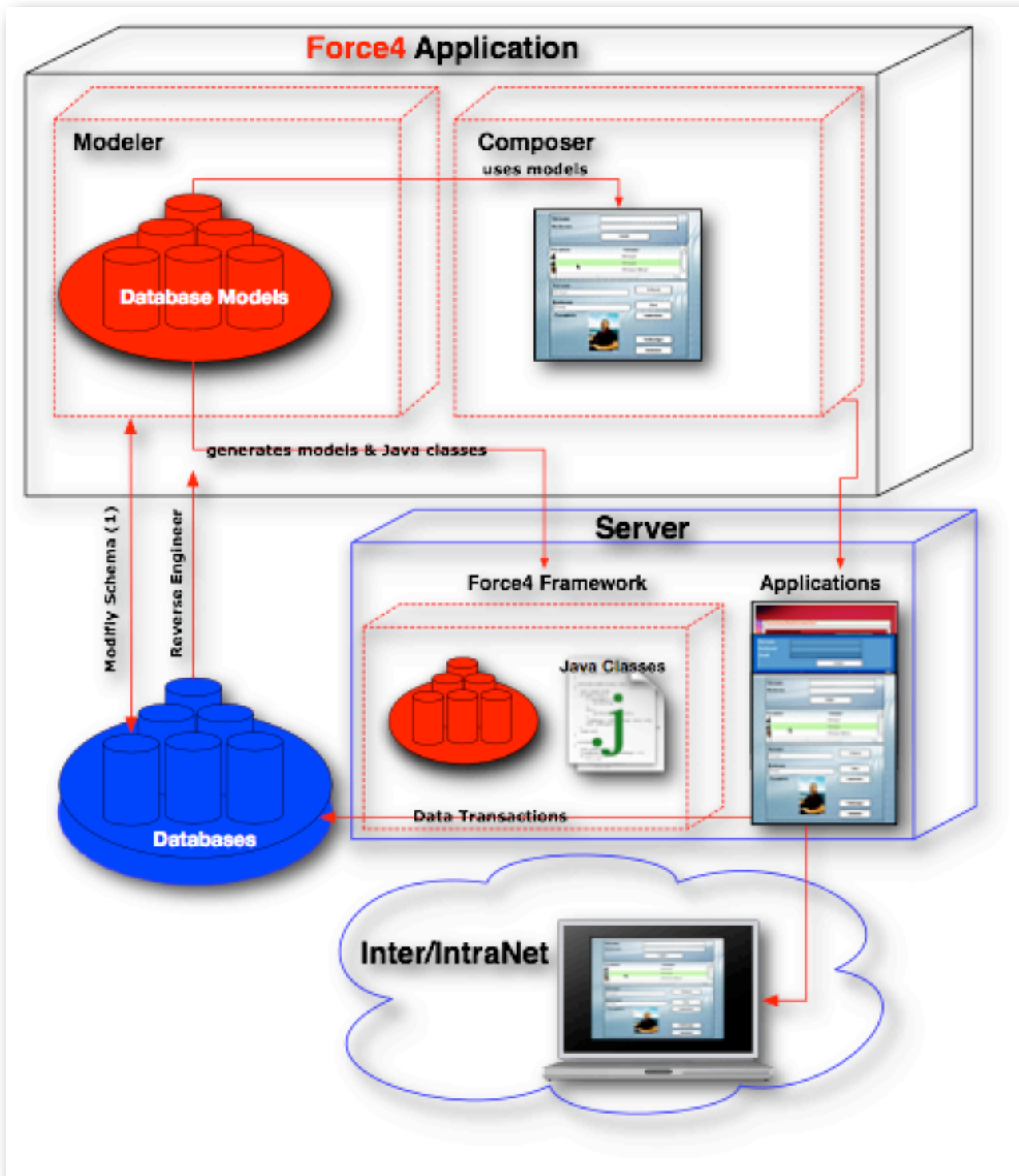
Force4 kombiniert die oben genannten Punkte in einem Produkt. Es integriert das Erstellen und Reverse-Engineering von Datenbank Modellen mit der Erzeugung der notwendigen Objekte und Zugriffsklassen sowie das Komponieren der darauf basierenden Rich Internet Applikationen.

Unterstützte Technologien

Force4 unterstützt z.Zt. (July 2005) Macromedia Flex als Presentation Server und J2EE Server als Backend Kommunikations Server. Demnächst wird Force4 auch OpenLaszlo als Presentation Server sowie RainingData OMNIS als Presentation und Backend Client/Server unterstützen. Force4 kann über seine Backend Frameworks mit allen SQL fähigen Datenbanksystemen kommunizieren, die über einen JDBC Treiber verfügen. Der Force4 Composer/Modeler ist für Microsoft Windows 98, 2000, XP und Apple Mac OS X 10.3 verfügbar. Die Force4 Serverkomponenten unterstützen z.Zt. Microsoft Windows basierende J2EE Server.

Arbeitsabläufe

Die Grafik zeigt einen Überblick über den Workflow von Force4. Dabei ist zwischen der Force4 Applikation und dem Force4 Server zu unterscheiden. Die Force4 Applikation wird auf Desktop Rechnern eingesetzt und besteht aus dem Modeler und dem Composer. Der Force4 Server ist das Kommunikations Backend für die durch den Composer erstellten Applikationen.



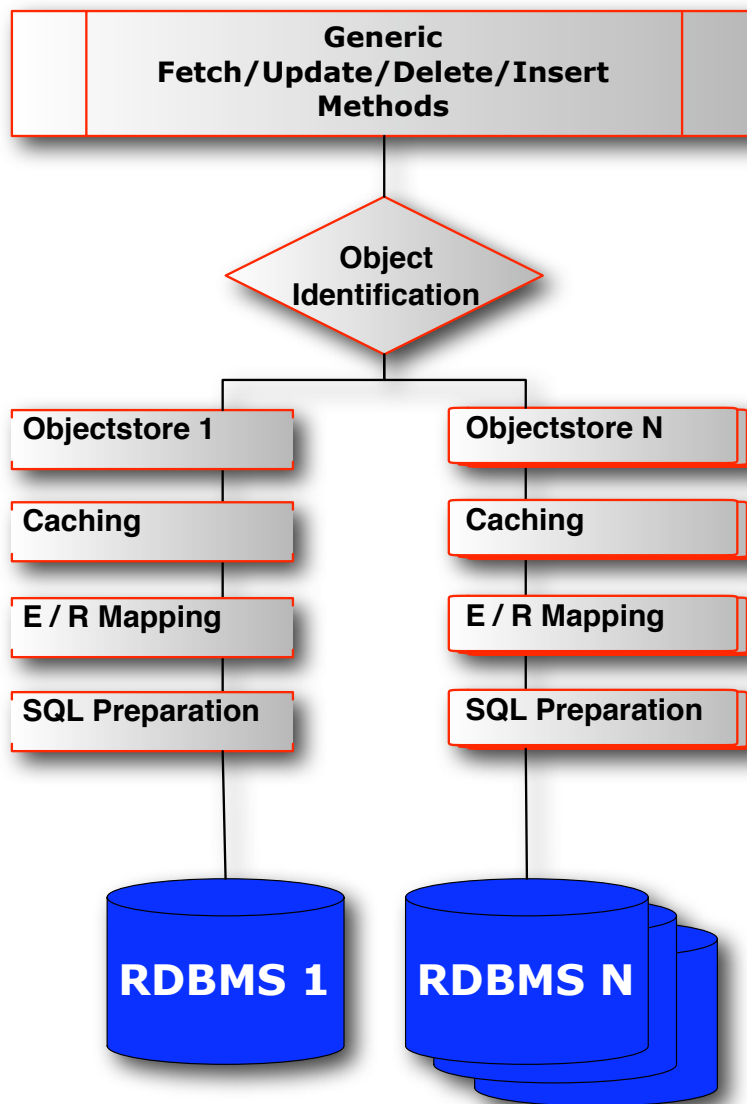
Das Erstellen von RIAs mit Force4 teilt sich in die Bereiche Datenbank Modeling und Applikation Komposition auf. Der Modeler kann mit den Datenbanken kommunizieren um deren Struktur auszulesen (Reverse Engineering) als auch deren Struktur ändern. Die Strukturen werden

in Modellen abgelegt. Diese abstrahieren Tabellen, Views, Columns und deren Verbindungen untereinander zu Entitäten und Relationen. Zusätzlich werden die relevanten Eigenschaften der Datenbank, wie z.B. verfügbare Daten Typen, JDBC Parameter, gespeichert. Die Modelle werden dann vom Composer benutzt um dem Anwender die Auswahl von Entitäten, Attributen und Relationen übersichtlich zu präsentieren und diese mit visuellen und funktionalen Komponenten zu verbinden.

Zudem werden die Modelle und, falls der Force4 Server J2EE basierend ist, die JAVA Klassen auf dem Server abgelegt. Diese werden vom Force4 Server Framework benutzt um das "Object Relational Mapping" vornehmen zu können. Zusätzlich erzeugt der Modeler die für die WebClient Programmierung notwendigen Klassen und Zugriffs Objekte. Auf diese referenziert der durch den Composer generierte Quellcode der Applikationen. Die vom Composer generierten Applikationen werden vom Presentation Server an die WebClients ausgeliefert.

Datenbanken

Die Schnittstelle der WebClients zu den Datenbanken wird im Force4 Server abgebildet. Der WebClient kommuniziert niemals mit der Datenbank direkt sondern nur durch ein unifiziertes Interface, das z.T. auf den durch Force4 erzeugten Modellen und zum anderen Teil auf fest definierten Funktionen beruht. Lesen Sie dazu auch die Abschnitte Anwendungen und Datenfluss.

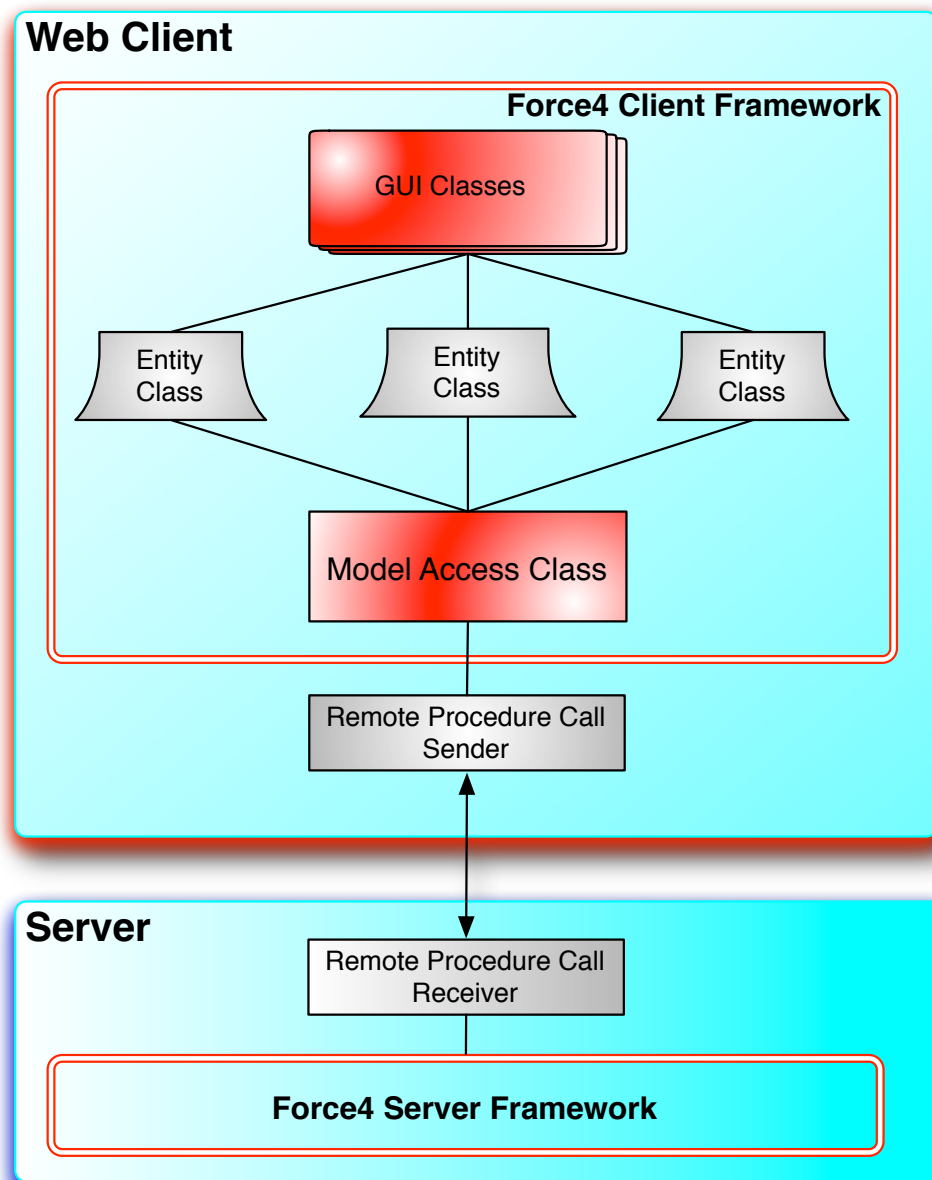


RDBMS werden im Falle von J2EE basierenden Force4 Servern generell über JDBC angesprochen. Aus der Leistungsfähigkeit des jeweiligen JDBC Treibers ergibt sich der Umfang des Reverse-Engineering als auch DDL Möglichkeiten da Force4 hier direkt über das java.sql Package auf die Datenbank zugreift. Diese Schritte werden im Force4 Modeler ausgeführt. Die zur Laufzeit nötigen Datenbank Zugriffe (z.B. Selects, Updates, ...) führt der ORM Layer in der Force4 Serverkomponente aus. Die Web Clients sprechen dazu Java Methoden (in Flex über Remote Objects, in OpenLaszlo über JavaRPC) an, die dann die entsprechenden Daten über den ORM Layer abholen oder an die Datenbank senden. Der ORM Layer benutzt dazu das im Force4 Modeler erstellte Model, um die in den Web Client Applikationen benutzen abstrakten Entitäten und Relationen in die der Datenbank entsprechenden Tabellen, Columns ... und den benötigten SQL Dialekt umzusetzen. Zudem wird die für das inserieren von neuen Daten notwendige Erzeugung von Primärschlüsseln vorgenommen. Diese kann im Modeler angepasst werden.

Primärschlüssel bilden die Grundlage für die Objekt Persistenz zwischen Server und Client. Während in den Clients die Datenbank Objekte mehrfach vorliegen können werden diese auf dem Server immer auf das selbe Objekt, das aus dem ORM Layer stammt, zurückgeführt. Änderungen etc. an den serverseitigen Objekten stehen dadurch sofort allen Clients zur Verfügung. Zudem sorgt der Force4 Server für die Integrität der Objekte, auch wenn mehrere Clients Manipulationen an dem selben Objekt vornehmen. Der ORM Layer sorgt für die Datenbank Integrität als auch für Objekt Caching und garantiert dadurch optimale Performance und Speichernutzung.

Aus den vorgenannten Gründen ergibt sich der Vorteil, das eine Web Applikation nicht den realen Datenbank Aufbau (Tabellen, Datentypen, Primary und Foreign Keys...) kennen muss sondern nur deren abstrahierte Ansicht. So ist es überhaupt möglich, ein unifiziertes Framework für die Applikationen anzubieten, Zudem erlaubt dieser Technik, Datenbank Strukturen (in gewissen Grenzen) zu ändern ohne die Applikationen ändern zu müssen. Ausserdem kann die Zieldatenbank durch ein anderes RDBMS System ausgetauscht werden da der Force4 Modeler Modelle zwischen Datenbanken konvertieren kann.

Anwendungen

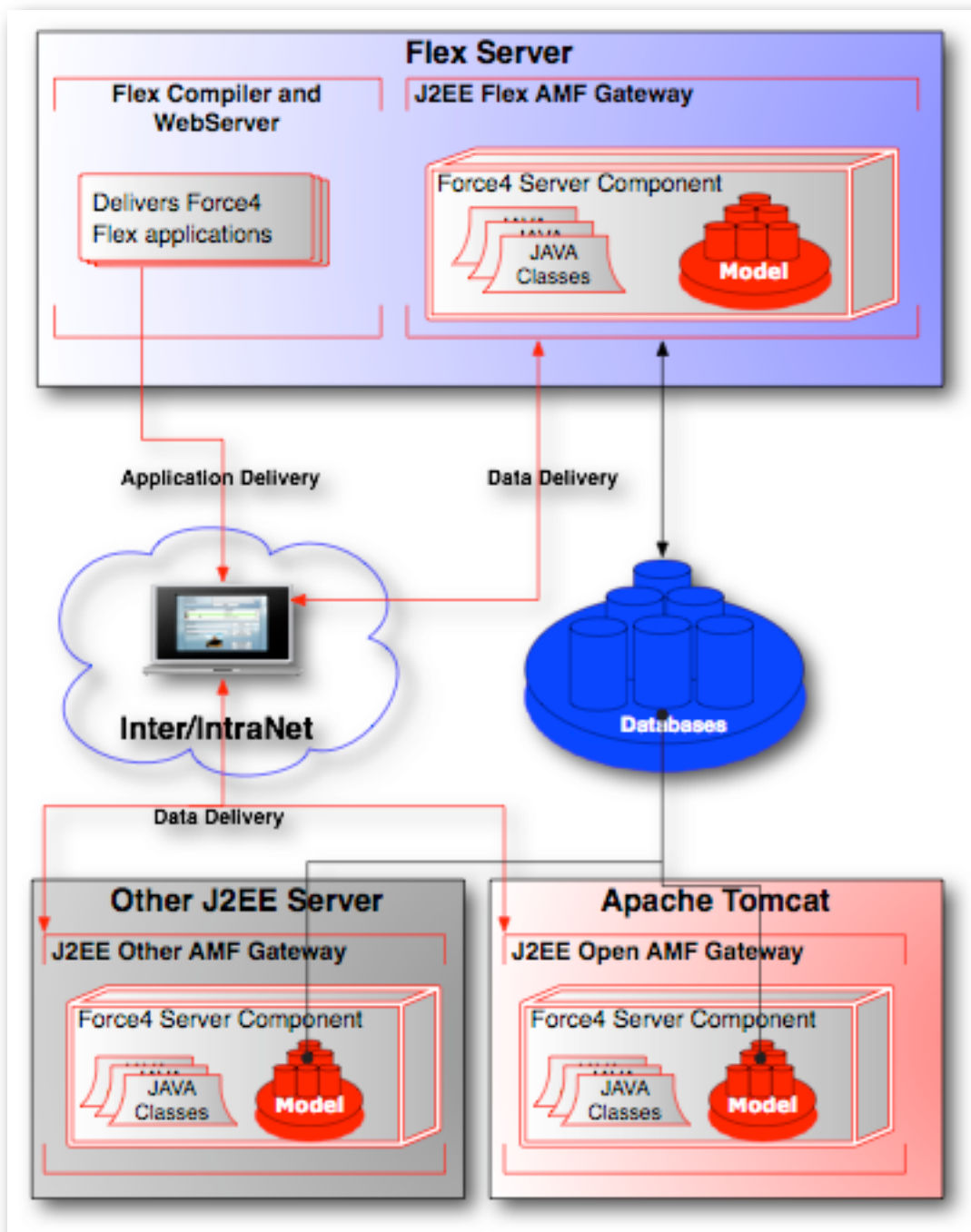


Die durch den Force4 Composer erstellten Applikationen benutzen das Force4 Client Framework sowie die durch den Modeler erzeugten Objekt- und Zugriffsklassen, um mit Datenbanken zu kommunizieren. Das Client Framework bietet Basis Datenbankzugriffsmethoden sowie ein "Event Management" an, die die funktionalen Force4 Client Komponenten (Suche, Liste, ...) benutzen. Das Client Framework liegt in Quell Code vor und ist in der Entwickler Dokumentation beschrieben sodass es auch in Applikationen eingesetzt werden kann, die nicht mit Hilfe des Force4 Composers erstellt werden.

Force4 RIAs bestehen aus zwei Arten von Komponenten. Einmal natürlich die, die sich aus dem Sprachumfang des Presentation Servers ergeben. Um jedoch automatische Funktionalität in die Applikationen einzuführen werden sogenannte Funktionale Komponenten benutzt. Diese bauen auf bestimmten Standardkomponenten auf. Die Funktionen umfassen Suchen, Listen- und Detail-Anzeigen. Eine Suche wird durch Eingabefelder (Text, Datum, Auswahllisten) präzisiert. Dabei werden aus dem Force4 Client Framework entsprechende Subkomponenten benutzt. Die Suchaktion löst eine entsprechende Anfrage über den Force4 Server an die Datenbank aus. Deren Ergebnisse werden dann in einer Liste zurückgeliefert, die in der Listenkomponente an-

gezeigt werden kann. Einzelne Einträge aus der Liste können wiederum in einer Detailkomponente spezifizierter angezeigt werden und dort auch geändert und gelöscht werden. Die funktionalen Komponenten kommunizieren untereinander über einen Eventmechanismus, der dafür sorgt das Änderungen in einer der Komponenten an bestimmte andere Komponenten verteilt werden. Damit können nicht nur einfache Suche-Liste-Detail RIAs erzeugt werden sondern auch komplexe Master-Detail Abhängigkeiten visualisiert werden.

Datenfluss



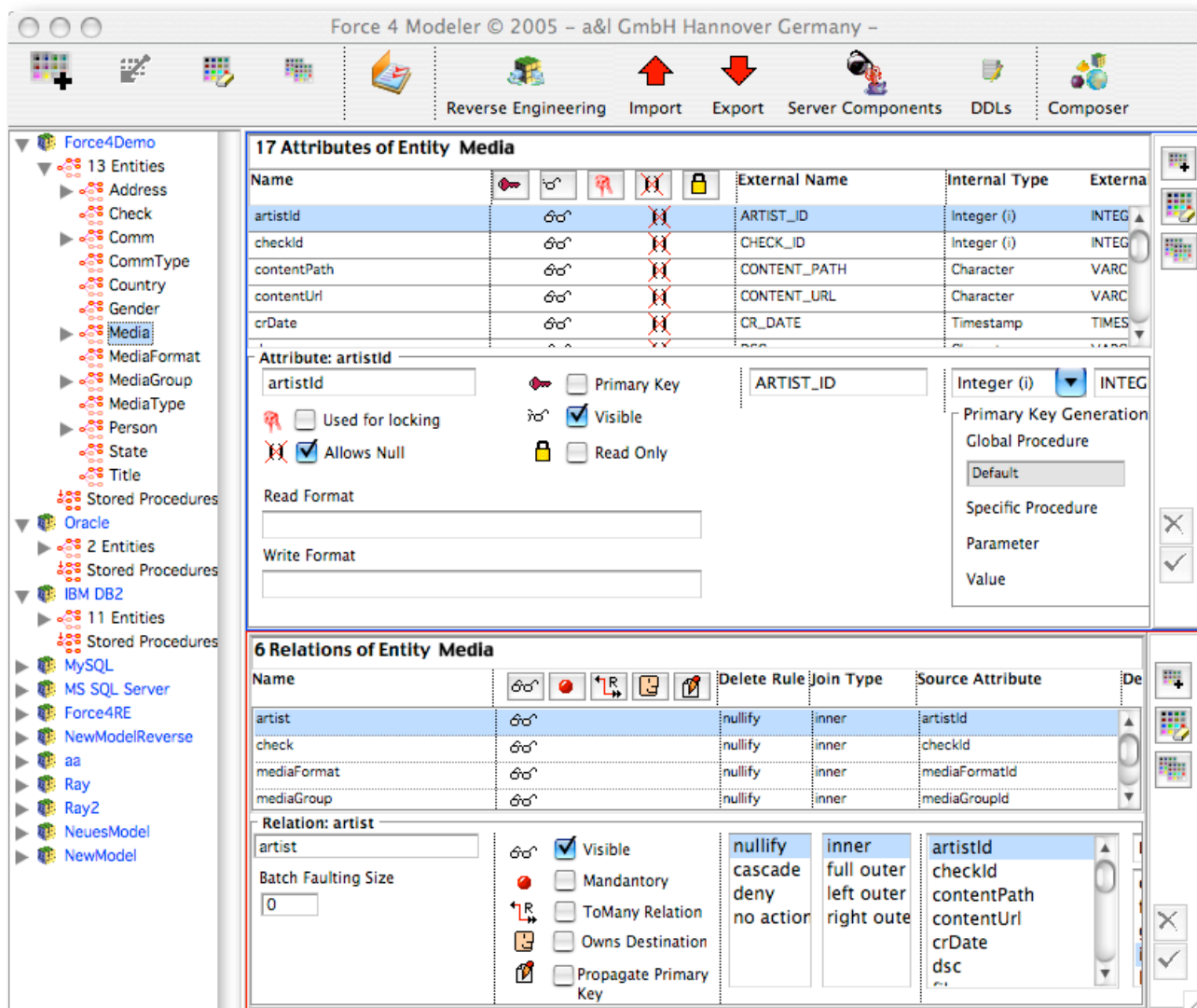
Der Datenfluss

gibt die schon öfters angesprochene Trennung zwischen dem Force4 Composer und der Force4 Server Komponente wieder. Die durch den Composer oder hand-programmierten Force4 Anwendungen werden von einem Presentation Server (Flex, OpenLaszlo) ausgeliefert. Von diesem werden u.U. auch Anwendungsbestandteile, Bilder, ... nachgeliefert. Die dann auf dem Client ausgeführte Anwendung benutzt, im Falle von Flex, die sogenannten "Remote Objects",

um über ein AMF Gateway mit der Force4 Server Komponente zu kommunizieren. D.h. dass alle Force4 Datenbank Operationen über diese Schnittstelle abgewickelt werden. Der Vorteil dieses Konzeptes liegt in einer möglichen Verteilung der Lasten zwischen Presentation Server und Force4 Server. Da die Force4 Server Komponente für J2EE Server nicht an Macromedia Flex gebunden ist, kann z.B. auch ein Standart Apache TOMCAT mit OpenAMF als Gateway zum Einsatz kommen.

Modeler

Der Force4-Modeler ist ein E/R-Modeling-Tool mit dem man auf einfache Art Datenstrukturen in einem "Repository" hinterlegt, um sie während der Modellierungs-Phase eines Prozesses einfach und schnell auf dem aktuellen Stand zu halten. Weiterhin fungiert der Force4-Modeler als Entwicklungs-Schnittstelle zwischen der Datenbank und JAVA, Macromedia Flex, OMNIS mit den jeweiligen spezifischen Nomenklaturen.



Das Model im Force4-Modeler ist zunächst ein abstraktes "Gebilde", bestehend aus Entitäten, Attributen und Relationen, um die Datenstrukturen aufzubauen und darzustellen, die sich zum Beispiel aus einem Business-Prozess ergeben. Der Aufbau kann manuell im Modeler vollzogen werden, das Einlesen von Strukturen aus bestehenden Datenbanken durch direkte Verbindung via JDBC, dem sog. Reverse Engineering ist auch möglich.

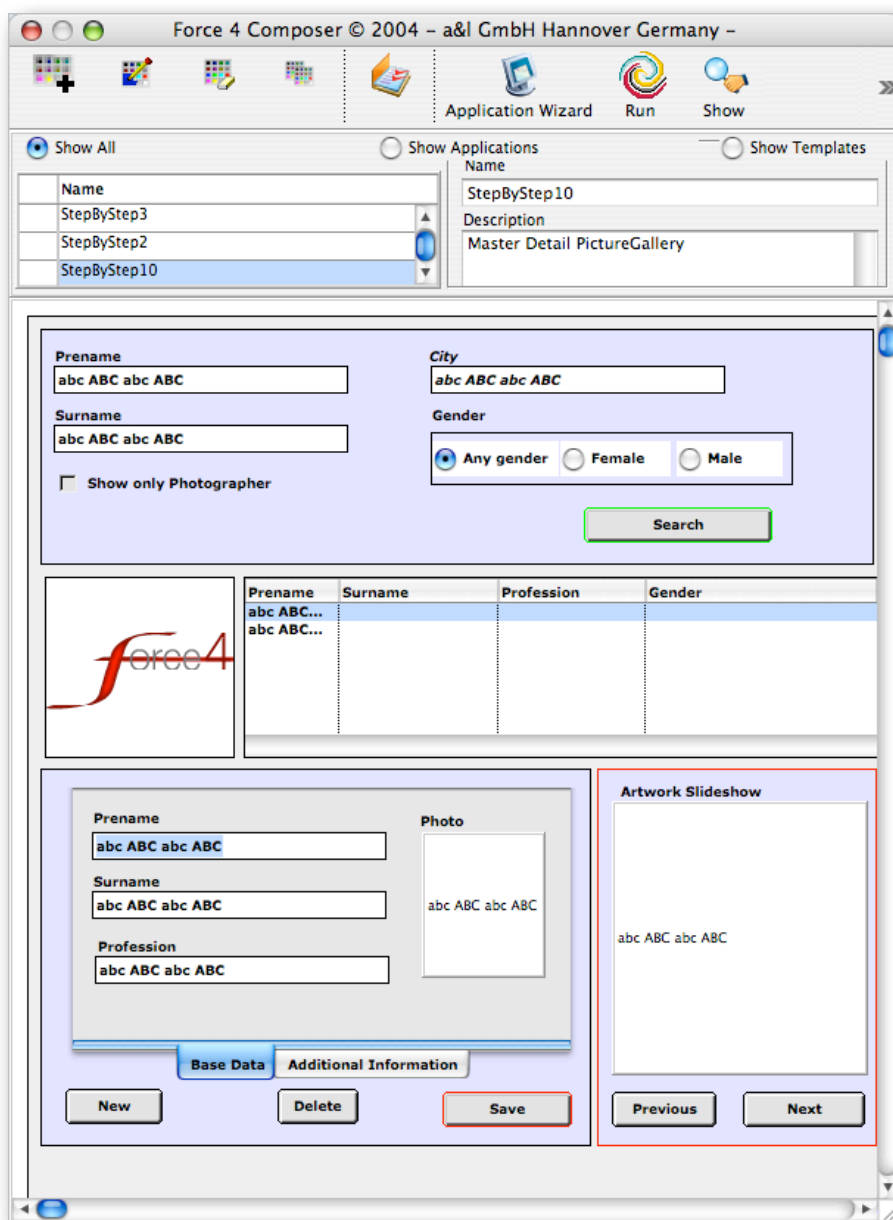
Ist ein Model entsprechend den funktionalen Vorgaben erstellt worden, so lassen sich damit zum einen Datenbanken, zum anderen aber auch direkte Schnittstellen zu JAVA, Macromedia Flex und OMNIS erstellen.

Entitäten und Attribute sind die abstrakten Gegenstücke zu Tabellen und Tabellenspalten in einer Datenbank. Zusätzlich zu den Eigenschaften, die notwendig sind, um eine Datenbank-Tabelle zu definieren, lassen sich in Entitäten und Attributen weitere Eigenschaften einstellen, um die Kompatibilität zu JAVA- und Macromedia Flex zu erzeugen.

Relationen im Force4-Modeler sind die Verbindungen zwischen den Entitäten. Durch den Aufbau dieser Relationen wird aus einer Ansammlung von Entitäten eine relationale Struktur. Die Abbildung der Relationen auf die Datenbank zeigt sich durch den Aufbau von sog. Foreign Key Constraints, Bedingungen auf Datenbank-Ebene, die die Beziehungen der Tabellen untereinander widerspiegeln (soweit die Datenbank dies unterstützt).

Composer

Der Force4 Composer ist das Tools, um schnell und einfach auf bestehenden E/R Modellen, die mit Hilfe des Force4 Modelers erzeugt worden sind, Rich Internet Applikationen zu erstellen. Dabei werden in einem WYSIWYG GUI Komponenten und Funktionen mit Hilfe von Drag&Drop



und anderen Hilfsmitteln zusammengeführt. Applikationen lassen sich auch aus Vorlagen, die ebenso mit dem Composer erstellt werden, erzeugen. Die einzelnen Bestandteile einer Applikation sind in Stil, Größen, etc. attributierbar. Zudem können diese Bestandteile, auch Container genannt, in anderen Applikationen wiederverwendet werden. Der Composer speichert die Applikationen in einer lokalen Datenbank und ist daher nicht auf die erzeugten Quelltexte angewiesen. Diese werden automatisch erzeugt und können, falls gegen einen Presentation Server gearbeitet wird, auch sofort im Echtbetrieb überprüft werden.

Ein Offline Modus ist verfügbar falls der Presentation Server nicht direkt bedient werden kann. Somit kann der Composer RIAs erstellen, die keine weitere Programmierung erfordern. Natürlich können die so erzeugten Applikationen nachträglich durch einen Entwickler im Quelltext geändert werden. Dies empfiehlt sich z.B., um zusätzlich spezifische Business Logik in die RIA einzubringen. Der Force4 Composer stellt die Basis dazu bereit. Aufgrund des integrativen Charakters von Force4 ist es zudem ein ideales Tool für Rapid Prototyping von datenbankgestützten Rich Internet Applikationen.

Server Flex und OpenLaszlo

Force4 erzeugt für Macromedia Flex und OpenLaszlo RIA Anwendungen als Quelltext, die durch den Presentation Server kompiliert werden und dann an den Client ausgeliefert werden. Dabei wird, wie schon angesprochen, das Force4 Framework benutzt. In dem Framework wird die Schnittstelle für die RemoteObjects/JavaRPCs definiert. Das Framework wird auf dem Flex/OpenLaszlo Server installiert, auf dem auch der Composer seine generierten Quelltexte zum kompilieren speichert.

Das Framework ist ein Bestandteil der Force4 Server Komponente, verlangt aber keine Force4 Lizenz und kann daher mühelos auch auf Presentation Servern genutzt werden, die selber nicht für die Force4 Datenbank Kommunikation zuständig sind. Der Kommunikationsbestandteil der Force4 Server Komponente kann daher auf gängigen J2EE kompatiblen Servern installiert werden. Die aktuelle Version erwartet eine JRE 1.4.2 oder kompatible JAVA Umgebung. Ein JDK wird nicht benötigt. Ausserdem enthält die Force4 Server Komponente alle notwendigen Packages, um mit dem Standard JDBC (java.sql) im JRE lauffähig zu sein.

Force4 benutzt kein JDBC/JNDI Name-Binding und ist somit komplett unabhängig von auf dem J2EE Server konfigurierten JAVA Datasources. Auch die für die Datenbank Kommunikation notwendigen JDBC Treiber sind in Force4 enthalten. Zusätzliche Treiber können einfach integriert werden.

Force4 Kernfunktionen

Modeler

- Anschluss aller SQL 92 fähigen Datenbanken
- z.B. IBM - DB2, ORACLE, FRONTBASE, MySQL, Microsoft SQL Server ...
- Integriertes Datenbank Modelling (Entity / Relationship)
- Reverse Engineering, Aufbau von Datenbank Modellen vorhandener Datenbanken
- Bereitstellung aller Zugriffs- und ORM Objekte (Object/Relational Mapping)
- OpenAMF Unterstützung

Composer

- Komponieren der Rich Internet Applikationen mit Drag & Drop Technik
- Bereitstellung funktionaler Komponenten für alle Datenbank Transaktionen
- Unterstützung der Canvas in Canvas Technik, verschachteln von Containern
- Aufbau eigener Templates (Vorlagen) zur einfachen Reproduktion
- Wiederverwendbarkeit aller Komponenten und Applikationen
- Unterstützung relevanter intelligenten WebClient-Systeme z.Zt. Flash und Omnis

Force4 Vorteile

Technik

- Anbindung aller, auch vorhandener, Datenbanken "on the fly".
- Die Verwendung von Datenbank Modellen statt tabellengebundener SQL Programmierung vereinfacht die Arbeit und schafft mehr Sicherheit.
- Eine Web Applikation muss nicht den realen Datenbank Aufbau kennen sondern nur deren abstrahierte Ansicht.
- Arbeitsabläufe zwischen Datenbank- und Anwendungsdesign können getrennt werden um Ressourcen effizienter zu nutzen.
- Ein einheitliches "Framework" reduziert den Pflegeaufwand und erleichtert die Einarbeitung neuer Kräfte.
- Wiederverwendbarkeit von Komponenten und Anwendungen in allen Projekten.
- Der ORM Layer sorgt für die Datenbank Integrität als auch für das Objekt Caching und garantiert dadurch optimale Performance und Speichernutzung.
- Änderung von Datenbank-Strukturen (in den SQL92 Grenzen) ohne die Applikationen ändern zu müssen
- Austausch der Zieldatenbank durch ein anderes RDBMS System, da Force4 Modelle zwischen Datenbanken konvertieren kann.
- Ausgabe von Force4 Applikationen für mehrere intelligente WebClient-Systeme (Flash, Omnis, OpenLaszlo...) erweitert die Einsatzmöglichkeiten der Anwendungen.

Ökonomie und Markt

- Wesentlich bessere Internet Anwendungen der 2. Generation bieten mehr Bedienungs -Komfort und -Sicherheit und erhöhen damit deutlich die Akzeptanz.
- Erhebliche Einsparungen von Zeit und Aufwand bei der Entwicklung von intelligenten Web Applikationen, bis zu 90% möglich.
- Sehr geringe Lernkurve und nicht erforderliches Datenbank-Spezialwissen verbreitert die Basis der Kräfte und senkt die personalgebundenen Kosten
- Schnelles Prototyping von Applikationen sichert Marktvorteile durch Überzeugung.
- Verkürzung der Markteinführung neuer Anwendungen ("time to market")
- Die Unterstützung mehrerer intelligenter WebClient-Systeme nutzt alle Möglichkeiten der Systeme und erweitert damit den Markt für Ihre Anwendungen
- Force4 Anwendungen / Modelle arbeiten mit allen SQL92 Datenbanken ohne Anpassungen