Abschlussarbeit
zum
Bachelor of Science im Fach Informatik

Thema:

Einsatz von Java-Frameworks in verteilten Systemen

(Umfrage in Softwarehäusern und IT-Abteilungen)

Fachbereich Programmiersysteme
Prof. Dr. Friedrich Steimann

erstellt von Ute Stegmann

(c) 04/2005 - 09/2005

Inhaltsverzeichnis

1 Einleitung 4
1.1 Aufgabenstellung und Ziele 4
1.2 Aufbau der Arbeit 4
1.3 Anmerkungen der Autorin 4
2 Begriffsbestimmungen 6
2.1 Die Rolle der Programmiersprache Java 6
2.2 Der Begriff des Frameworks 6
2.3 Verteilte Systeme 7
2.3.1 Eigenschaften Verteilter Systeme 7
2.3.2 Architekturen 8
2.4 Persistenz 9
2.5 Modellierung (UML, MDA, MDSD) 10
3 Framework- und Java-Technologien 11
3.1 Die J2EE-Spezifikation 11
3.1.1 J2EE Infrastruktur 12
3.1.2 Wichtige APIs 13
3.1.3 Probleme der EJB-Technologie 16
3.2 Zum Vergleich: Die Microsoft .NET-Technologie 17
3.2.1 „Konzept 17
3.3 Das Java-Applikations Spring 18
3.3.1 Aufbau des Spring-Frameworks 18
3.3.2 Anwendungsszenarien des Spring-Framework (Quelle [Spring1]): 20
3.4 Eclipse (Open-Source IDE und Framework) 22
3.5 Das MVC-Web-Framework Struts (Open-Source) 23
3.5.1 Aufbau 23
3.5.2 Theoretischer Ansatz 23
3.5.3 Praktische Umsetzung 24
3.5.4 Kritik 25
3.6 Java Server Faces 26
3.7 Java Data Objects (JDO) 27
3.8 Das Persistenzframework Hibernate (Open-Source) 28
3.9 Das Publishing Framework Cocoon (Open-Source) 29
3.10 Weitere Open-Source-Frameworks 30
3.10.1 Turbine 30
3.10.2 Velocity 30
3.11 Test-Framework (JUnit) 31
4 Vorbereitung der Umfrage 32
4.1 Vorgehensweise 32
4.2 Aufbau der Umfrage 33
4.3 Technische Umsetzung der Umfrage (Internet/Online) 34
4.4 Ermittlung von und Kontakt zu Teilnehmern (Firmen) 36
4.4.1 Kriterien zur Auswahl der Teilnehmer 37
4.4.2 Strategien und Quellen zur Teilnehmerauswahl 37
4.4.3 Vorgehensweise 38
5.1 Teil 1 Architekturen für verteilte Anwendungen 39
5.1.1 Motivation der gestellten Architekturfragen 39
5.1.2 Fragen und Ergebnisse zu Teil 1 im Detail 39

5.2 Teil 2 Strategien zur Frameworkentwicklung 41
5.2.1 Motivation der gestellten Strategiefragen zur Frameworkentwicklung 41
5.2.2 Fragen und Ergebnisse zu Teil 2 im Detail 41
5.3 Teil 3 Framework- und Java-Technologien 43
5.3.1 Motivation der gestellten Fragen zu den Framework- und Java-Technologien 43
5.3.2 Fragen und Ergebnisse zu Teil 3 im Detail 43
5.4 Teil 4 Fragen zur Modellierungspraxis 55
5.4.1 Motivation der gestellten Fragen zur Modellierung 55
5.4.2 Fragen und Ergebnisse zu Teil 4 im Detail 55
5.5 Teil 5 Framework-Bewertung 56
5.5.1 Motivation der Bewertungsfragen zum Frameworkeinsatz 56
5.5.2 Fragen und Ergebnisse zu Teil 5 im Detail 56
5.6 Teil 6 Unternehmensangaben 57
5.6.1 Motivation der Frage zum Unternehmen 57
5.6.2 Fragen und Ergebnisse zu Teil 6 im Detail 57
6 Summary 59
7 Literaturverzeichnis 60
8 Abbildungsverzeichnis 61

1 Einleitung

1.1 Aufgabenstellung und Ziele

Verschiedenen Firmen entwickeln eigene Java-Frameworks bzw. habe in der Vergangenheit entsprechende Entwicklungen durchgeführt. Teilweise basieren diese auf reinem Java und Swing, teilweise auf J2EE oder es werden gewisse Aspekte mit bestehenden Frameworks aus der Open-Source-Welt wie z.B. Struts abgebildet.

Ganz allgemein gesprochen ist ein Framework eine Menge kooperierender Klassen, die ein wieder verwendbares Design für einen spezifischen Anwendungsbereich vorgeben (abstrakte Klassen, Interfaces und Kollaborationen).

Ziel dieser Arbeit ist es näher zu untersuchen, was ein Java-Framework für verteilte Systeme nach Meinung verschiedener Unternehmen (genauer IT-Entscheider) leisten sollte und wie die jeweiligen konkreten Umsetzungen aussehen.

Hierzu soll in verschiedenen Softwarehäusern und IT-Abteilungen von größeren Unternehmen eine Umfrage durchgeführt werden, die Java-Frameworks nach bestimmten Kriterien klassifiziert. Ergänzend dazu sollen existierende Java-Frameworks aus der Open-Source-Welt und ggf. öffentlich zugängliche Informationen von gewerblichen Frameworks untersucht werden.

1.2 Aufbau der Arbeit

In Kapitel 2 werden zunächst die Grundlagen zu den Begriffen Framework und verteilte Systeme skizziert.

Kapitel 3 gibt dann einen Überblick über gängige Framework- und Java-Tochnologien. Außerdem werden einige bekannte Open-Source-Frameworks kurz vorgestellt.

Zur Vorbereitung der Umfrage wurde zunächst Aufbau, Inhalt und technische sowie organisatorische Umsetzung der Umfrage geplant. In Kapitel 4 werden diese Aktivitäten sowie das Vorgehen bei der Auswahl der Teilnehmer beschrieben.

Die Umfrage selber und die Ergebnisse sind schließlich in Kapitel 5 umfassend dargestellt.

Den Abschluss bilden in Kapitel 6 die Schlussfolgerungen aus den Umfrageergebnissen sowie ein Ausblick auf mögliche zukünftige Entwicklungen.

Das Literatur- und das Abbildungsverzeichnis befinden sich am Ende der Arbeit.

Ein Ausdruck der als Internet-Befragung durchführten Umfrage-Seiten ist als Anlage beigefügt.

1.3 Anmerkungen der Autorin

Die Idee zu der vorliegenden Arbeit entstand aus meiner beruflichen Praxis als DV-Kauffrau und zuletzt Projektleiterin mit dem späten Einstieg in ein Informatik-Studium.

Ein kleiner historischer Rückblick aus meinem natürlich subjektiven Erleben:

Der Einsatz von Frameworks in Client-/Server-Architekturen ist kein neues Thema.
Schon in den 80er Jahren begegneten mir in einem Softwarehaus Rahmenwerke zur Softwareentwicklung von Client-/Server-Anwendungen mit damals noch ausschließlich Fat-Clients. Man entwickelte noch nicht objektorientiert und es gab Bezeichnungen wie

Dialogsteuerung für solche Programmiergerüste, die individuell z.B. in Softwarehäusern entwickelt wurden.

Schon damals habe ich z.B. gelernt, dass Anwendungsentwicklung um effektiv zu sein, möglichst wenig mit systemtechnischen Problemen belastet werden sollte und dass Plattformunabhängigkeit insbesondere wenn Software für verschiedene Kunden mit individueller Hardware entwickelt wird, ein erstrebenswertes Ziel ist.

Mit dem Einzug der Objektorientierten Softwareentwicklung zunächst unter C++ erlebte ich anfangs dann eher mehr „Wildwuchs“ bei der Softwareentwicklung. Vielleicht auch weil die aus der C-Programmierung kommenden Softwareentwickler ein wenig mehr Interesse an der „Selbstverwirklichung“ hatten als daran, was im Interesse der Software-Anwender bzw. Kunden war. Selbst die vorher schon bekannte klare Trennung der Softwareschichten schien sich insbesondere zwischen Präsentationsschicht und Anwendungslogik bei den ersten Windows-Programmen plötzlich aufzulösen.

Ich persönlich hatte mir immer gewünscht, dass sich eine Art Standard Entwicklungsrahmen etabliert (den Begriff Framework kannte ich damals noch nicht), auf den man mittelfristig setzen kann.

Mit dem Einzug der moderneren OO-Sprache Java und den Standards wie J2EE von Sun Microsystems, die sich weltweit durchsetzten, schien dieser Wunsch in Erfüllung zu gehen.

Allerdings gab es – glaubte man der Open-Source-Gemeinde- auch bald Kritik insbesondere an den sogenannten schwergewichtigen EJB-Containern und an den Entity-Beans.

Nun kamen also aus der Open-Source-Welt Ergänzungen, Verbesserungen, eigene Frameworks, Microsoft ging mit C# und .NET gar einen eigenen Weg. Und der gemeine IT-Entscheider wusste wieder nicht, worauf er eigentlich langfristig setzen soll.

Klar ist, dass es heute viele Möglichkeiten gäbe effektiv und Dank Java und Open-Source sogar kostengünstig Software für verteilte Systeme zu entwickeln.

Die Frage die mich bewegt ist, welche Standards werden heute tatsächlich eingesetzt und welche Qualität und Leistungsfähigkeit haben die eingesetzten Frameworks.

Aus beruflichen Kontakten erfuhr ich vor einiger Zeit, dass sehr große Unternehmen teilweise komplett eigene Java-Frameworks für verteilte Systeme über mehrere Jahre entwickelt haben. Und hier stellt sich die Frage, was die Gründe dafür waren und ob dies tatsächlich zu besserer Qualität geführt hat.

Was lag also näher, als über eine Umfrage diesen Fragen einmal nachzugehen.

Dankenswerter Weise hat Prof. Friedrich Steimann mir die Möglichkeit gegeben, dies in meiner Bachelor-Abschlussarbeit an der FernUniversität in Hagen zu tun.

(Ute Stegmann)

Hamburg, den 06.09.2005

2 Begriffsbestimmungen

Dieses Kapitel definiert in Kurzform die für gibt das Thema der Umfrage relevanten Begriffe Java-Frameworks und Verteilte Systeme.

2.1 Die Rolle der Programmiersprache Java

[wikipedia1]: „Java ist eine objektorientierte Programmiersprache und als solche ein eingetragenes Warenzeichen der Firma Sun Microsystems.

Java-Programme benötigen zur Ausführung eine spezielle Umgebung, die Java-Laufzeitumgebung oder Java-Plattform genannt wird und als wichtigsten Bestandteil die Java Virtual Machine enthält.

Der Vorteil ist, dass nur diese Umgebung an verschiedene Computer und Betriebssysteme angepasst werden muss. Sobald dies geschehen ist, laufen auf der Plattform alle Java-Programme ohne Anpassungsarbeiten. Die Programmiersprache Java ist eine Komponente der Java-Technologie."

Da für fast jedes heute relevante Betriebssystem eine Java Virtual Machine existiert, kann man quasi von einer Plattformunabhängigkeit sprechen. Zudem ist Java kostenlos zu beziehen. Was beides sicher zur rasanten Verbreitung von Java beigetragen hat.

Die Java-Technologien haben sich als wichtigster Standard für die Entwicklung von Verteilten Systemen etabliert. Weitere Informationen zu Java-Technologien finden Sie in Kapitel 3.

2.2 Der Begriff des Frameworks

Ein Rahmenwerk oder (Programm-)Gerüst, englisch Framework, ist formal eine Menge kooperierender Klassen, die ein wieder verwendbares Design für eine spezifische Anwendungsarchitektur vorgeben (abstrakte Klassen, Interfaces und Kollaborationen).

Definition nach [john]:
„A Framework is a reusable design of all or part of a system that is represented by a set of abstract classes and the way their instances interact."

Frameworks geben die Basis der Anwendungsarchitektur vor, lassen dem Entwickler aber bestimmte Freiheiten bei der Ausgestaltung der fertigen Anwendungssoftware.
Für größere Applikationen wird häufig eine Verbindung von mehreren Frameworks für jeweils bestimmte Schichten der Anwendungsarchitektur oder für bestimmte Teilaspekte der Entwicklung eingesetzt.
Bezogen auf die Art der Benutzung kann man folgende Klassifizierung vornehmen:
[wikipedia1]:„

- **White-Box-Frameworks** bestehen u.a. aus einer Reihe von abstrakten Klassen. Die Aufgabe des Entwicklers besteht hier darin, die abstrakten Methoden dieser Klassen in den abgeleiteten Unterklassen der konkreten Applikation zu überschreiben. Um dies tun zu können, ist eine gute Kenntnis der Framework-Klassen erforderlich - daher leitet sich die Bezeichnung White-Box-Framework ab.
- Das Gegenstück bilden die **Black-Box-Frameworks**. Diese bieten bereits fertige Klassen an, die der Entwickler instanziieren und so zu einer Applikation zusammensetzen kann. Verfügbare Frameworks sind zumeist Mischformen zwischen White-Box- und Black-Box-Frameworks.

Mit steigendem Reifegrad tendieren Frameworks zum Black-Box-Ansatz."

Neben dem allgemeinen Oberbegriff des Frameworks existieren noch folgende Unterteilungen:

- Der Begriff des **Application-Framework** (Anwendungsrahmenwerk), das einen Rahmen für eine bestimmte Klasse von Anwendungen vorgibt, wird verwendet, wenn ein kompletter Anwendungsrahmen vorgegeben wird und das Prinzip „don't call us we call you" gilt, d.h. das Rahmenwerk steuert die Programmabläufe immer auf eine vorgegebene Art und Weise.

weitere Typen nach [wikipedia1]:„

- **Domain Frameworks** hingegen bilden den Programmrahmen für einen bestimmten Problembereich (vertical slice), d.h. Funktionen und Strukturen die zur Lösung dieses Problembereichs im Allgemeinen benötigt werden.
- **Class Frameworks** fassen Klassen und Methoden zusammen, die Unterstützung auf einer bestimmten Abstraktionsebene für ein breites Anwendungsfeld bieten.
- **Komponenten-Frameworks** abstrahieren von der objektorientierten Ebene und bieten eine Umgebung zur Entwicklung und Integration von Software-Komponenten an. Software-Komponenten werden dabei meist als Bündel von Klassen mit eindeutig definierten Schnittstellen betrachtet."

Im weiteren Verlauf dieser Arbeit verwendete ich im allgemeinen den Oberbegriff Framework; da sich dieser auch für Frameworks, die nur für Teilaspekte eingesetzt werden, .durchgesetzt hat.

2.3 Verteilte Systeme

Nach [vertsys2] ist ein verteiltes System ein System, aus Hard- und Softwarekomponenten auf vernetzten Computern, die miteinander über den Austausch von Nachrichten kommunizieren. Der Hauptgrund und Vorteil eines verteilten Systems gegenüber nicht verteilten Systemen ist die gemeinsame Nutzung von Ressourcen wie Hardwarekomponenten, Softwarekomponenten, Daten, Informationen und Funktionalität.
Das größte Verteilte System ist das weltweite Internet (world wide web).

2.3.1 Eigenschaften Verteilter Systeme

Die beiden wichtigsten Eigenschaften eines Verteilten Systems sind:

- **Transparenz**: Das Ziel ist, vor dem Benutzer (z.B. Anwendungsentwickler) eines verteilten Systems bestimmte Aspekte der Verteilung zu verbergen, da die Art der Verteilung für ihn nicht relevant und idealerweise auch nicht ersichtlich sein soll. Das System verhält sich transparent, als hätte der Nutzer es mit einem Gesamtsystem zu tun. (Quelle [vertsys1])

- **Nebenläufigkeit**: Verteilte Systeme werden von vielen Benutzer gleichzeitig genutzt. Jede Ressource ist dabei so auszulegen, dass Sie in der nebenläufigen Umgebung sicher ist. (Quelle [vertsys1]) [vertsys1]

Weitere Kriterien finden sich in dem Fragen zu verteilten Systemen innerhalb der Umfrage in Kapitel 5.

2.3.2 Architekturen

Zwei Architekturmodelle für verteilte Systeme finden heutzutage die größte Verbreitung:

- **Client-Server-Architekturen**: Ein einfaches Client-Server-System besteht aus einem Client, der eine Verbindung mit einem Server aufbaut. Der Client bietet die Benutzeroberfläche oder die Benutzerschnittstelle der Anwendung an. Der Server stellt die Funktionalität zur Verfügung. In n-Tier-Architekturen werden verschiedene Arten von Servern für die verschiedenen Schichten der Software eingesetzt (z.B. Präsentation, Anwendungslogik, Datenhaltung). (Quelle [vertsys2])
- **Peer-to-Peer-Architekurmodell**: Dieses Architekturmodell arbeitet mit gleichberechtigten Prozessen. Peer-to-Peer-Prozesse laufen lokal und tauschen nur bei Bedarf Informationen aus. (Quelle [vertsys2])

Im Bereich der Java-Frameworkentwicklung für Verteilte Systeme spielen überwiegend Client-Server-Architekturen eine entscheidende Rolle.

Die einzelnen Schichten der Anwendungsarchitekturen, die zu verteilen sind, haben folgende Aufgaben:

- Die **Präsentationsschicht** bildet die Schnittstelle für den Anwender.
- Die **Anwendungslogik** bildet den Kern der Anwendung und beinhaltet die eigentliche Funktionalität.
  Sie stellt die Verbindung zwischen Präsentationsschicht und Datenhaltung her.
- Die **Datenhaltungsschicht** stellt insbesondere bei datenzentrierten Anwendungen die Persistenz der Daten her und ist für die Sicherung der Datenkonsistenz zuständig.

Auf der Client-Seite unterscheidet man zwischen

- **Thin Clients** (engl. dünn), die nur die reine Präsentationsschicht der Anwendung enthalten (Beispiel Darstellung im Webbrowser), und
- **Fat Clients** (engl. dick), die neben der Präsentationsschicht auch Anwendungslogik ausführen.

Die Art der Client-Server-Architekturen wird im wesentlichen durch die Aufteilung bzw. Anzahl der Tiers (Schichten) bestimmt.

Angefangen vom einfachen **2-Tier**-Client-Server-Modell mit einer Serverschicht (meist Datenbankserver) und Fat-Clients, über die **3-Tier-Architektur** mit Client-Tier (Präsentation), Middle-Tier (Anwendungslogik) und Server-Tier(Datenhaltung), enthalten **4- und mehr-Tier-**Architekturen weitere Tiers zwischen Client und Server getrennt nach ihren Aufgaben (z.B. Webserver, Applicationserver, Datenbankserver).

2.4 Persistenz

Mit Persistenz bezeichnet man in der objektorientierten Programmierung die Fähigkeit Objekte bzw. Datenstrukturen dauerhaft, d.h. beispielsweise in Datenbanken oder Dateisystemen, zu speichern.

Im Gegensatz dazu bezeichnet man „flüchtige“ Objekte als transient.

Als Speichermedium haben sich heutzutage Relationale Datenbanken mit SQL-Syntax wie z.B. Oracle, DB2, oder MySql durchgesetzt.

In dem „Stilbruch“ zwischen Objektorientierter Sicht der Anwendungsprogrammierung und der relationalen Sicht auf die Datenhaltung liegt eine Ursache für die vergleichsweise aufwendige Persistenzlogik, die in der OOP notwendig ist.

Denn es ist das Problem der Abbildung von Objekten auf relationale DB-Tabellen zu lösen.

Und insbesondere die Performance bei großen Datenmengen kann hierbei ein Problem werden.

In Kapitel 3 werden verschiedene Technologien (z.B. Entity Beans der J2EE-Technologie, O/R-Mapping mit dem Hibernate-Framework, Java Data Objects) skizziert.

2.5 Modellierung (UML, MDA, MDSD)

Die moderne objektorientierte Softwareentwicklung ist ohne Modellierung nicht mehr vorstellbar. Die allseits bekannte *Unified Modeling Language* (UML) bzw. entsprechende Tools zur UML-Modellierung sind dabei ein unverzichtbares Instrument.

Ein weiterer junger Standard der Object Management Group (OMG) ist MDA, die Model Driven Architecture, und darauf basierend die MDSD, das Model Driven Software Developement. Nach MDA soll für die Konstruktion eines Softwaresystems ein ausführliches, formal genaues Modell gebaut werden soll, aus dem Softwarekomponenten generiert werden können. Sowohl MDA als auch MDSD haben das Ziel einen einheitlichen Standard zur Anwendung modellgetriebener Softwareentwicklung zu etablieren und durch Formalisierung der abzubildenden Geschäftslogik eine Automatisierung der Softwareentwicklung zu ermöglichen.(Quelle: [wikipedia1])

Insoweit stellt sich für den Einsatz von Frameworks bei der Softwareentwicklung die Frage, wie umfangreich die Modellierung betrieben wird bzw. in die Framework-Technologie integriert ist und ob die neuen Standards in den Unternehmen eine Rolle spielen.

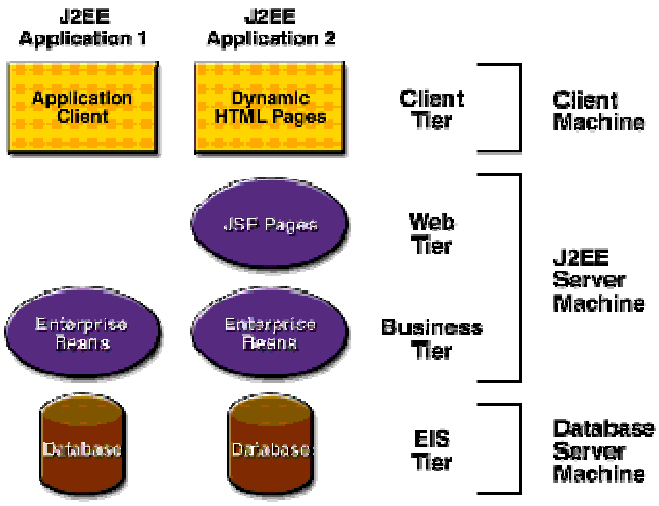
Eine ausführliche Behandlung dieses Themas hätte den Rahmen der Arbeit gesprengt, deshalb ist in der Umfrage auch nur eine Frage zum Umfang der Modellierung enthalten und es wird nach dem Einsatz von Codegeneratoren gefragt. Ergebnisse dazu finden Sie in Kapitel 5.

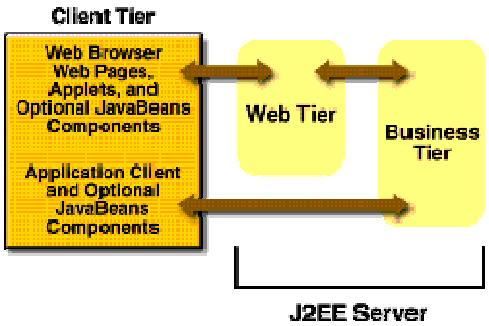
3 Framework- und Java-Technologien

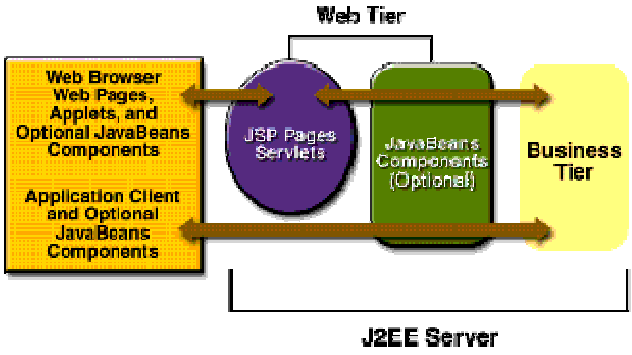
3.1 Die J2EE-Spezifikation

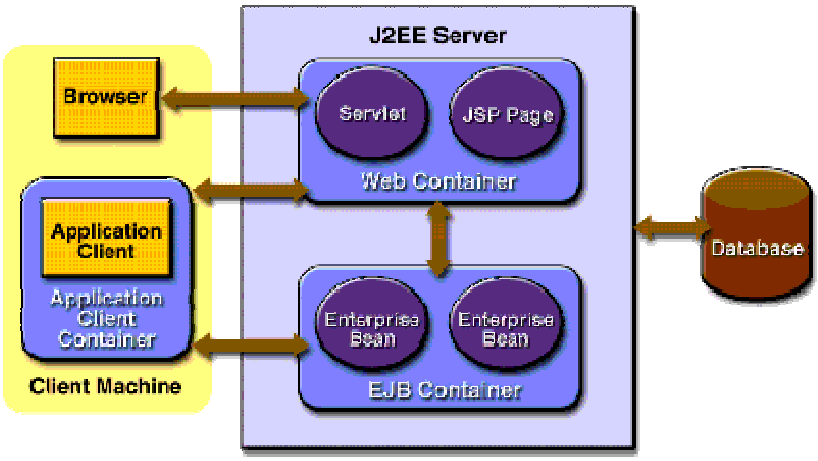
Die Java 2 Platform Enterprise Edition (J2EE) ist die Erweiterung der Java 2 Plattform Standard Edition (J2SE) zur Middleware-Plattform für verteilte Java-basierte Enterprise-Anwendungen. Kernkonzept der Plattform ist das Komponentenmodell mit Komponentenbegriff und Container. J2EE ist genauer eine Spezifikation die dazu dient, einen allgemein akzeptierten Rahmen zur Verfügung zu stellen, um mit modularen *Komponenten verteilte*, *mehrschichtige* Anwendungen zu entwickeln. Die Schnittstellen zwischen den Komponenten und Schichten sind klar definiert. Damit sind Komponenten verschiedener Software-Hersteller interoperabel, sofern sich die Hersteller an die Spezifikation halten.
Die Spezifikation stellt ferner sicher, dass eine innerhalb der Spezifikation erstellte verteilte Anwendung gut *skalierbar* ist.
In einer J2EE-Umgebung wird Geschäftslogik oft in der Form einer Enterprise JavaBeans Komponente implementiert. Diese Komponenten laufen innerhalb eines EJB-Containers, welcher für die Bereitstellung von infrastrukturellen Diensten, wie z.B. Persistenz, Transaktionssteuerung oder Lebensdauerverwaltung, verantwortlich ist. Enterprise Beans erlauben dem Entwickler, sich auf die Geschäftslogik zu konzentrieren, während die infrastrukturellen Dienste von dem Container gesteuert werden.

Folgende Grafiken aus [SunTutorial1] der Spezifikation 1.4 geben einen Überblick der Technologie, danach folgt eine Zusammenfassung die auszugsweise [Wikipedia1] entnommen ist:









Die nachfolgende Zusammenfassung der J2EE-Technologie ist auszugsweise [wikipedia1] entnommen:

„

3.1.1 J2EE Infrastruktur

J2EE-Komponenten erfordern als Laufzeitumgebung eine spezielle Infrastruktur, einen sogenannten J2EE Application Server. Dieser Server stellt technische Funktionalitäten wie

- Sicherheit (Security)
- Transaktionsmanagement
- Namens- und Verzeichnisdienste
- Kommunikation zwischen J2EE-Komponenten
- Management der Komponenten über den gesamten Lebenszyklus (inklusive Instanziierung)
- Unterstützung für die Installation (Deployment)

zur Verfügung. Des weiteren kapselt der Server den Zugriff auf die Ressourcen des zugrundeliegenden Betriebssystems (Dateisystem, Netzwerk ...). J2EE Server werden häufig auch mit dem Begriff Application Server bezeichnet.

Wesentliche Bestandteile eines J2EE Servers sind

- ein EJB-Container als Laufzeitumgebung für Enterprise Java Beans
- ein Web-Container als Laufzeitumgebung für Servlets und Java Server Pages (JSP)
- ein Client-Container als Laufzeitumgebung für einen Application-Client

J2EE-Server unterstützen die nachfolgend genannten J2EE Programmierschnittstellen ("APIs"). Sie sind meist selbst in der Programmiersprache Java entwickelt.

Es sind zahlreiche Implementierungen für J2EE-Server verfügbar, teils kommerziell, teils in Form frei verfügbarer Open Source-Lösungen. Eine Referenzimplementierung wird von Sun Microsystems zur Verfügung gestellt. Zu beachten ist, dass nicht alle Server die Spezifikation von J2EE vollständig abdecken - viele Server bieten nur Teilaspekte des Standards.

Als weitere Infrastrukturkomponente kommt für die persistente Speicherung von Daten ein Datenbank Managementsystem (DBMS) zum Einsatz.

Der client-seitige Zugriff auf eine J2EE-Anwendung schließlich erfolgt meist über einen Browser, daneben sind aber auch Applikations-Clients (Java-Applikationen, CORBA-Komponenten, Web Service-Clients über XML) möglich.

3.1.2 Wichtige APIs

Die J2EE APIs beinhalten verschiedene Technologien, die die Funktionalität des Basis-J2SE-APIs erweitern. Alle APIs der aktuellen Version sind im folgenden aufgelistet.

Java Naming and Directory Interface (JNDI)

ist eine gemeinsame Schnittstelle mit der alle Java Klassen auf Namens- und Verzeichnisdienste zugreifen können. Über JNDI wird insbesondere der Zugriff auf J2EE Komponenten sichergestellt.

Enterprise Java Beans (EJB)

beinhalten die Geschäftslogik einer Enterprise Anwendung oder gestatten Zugriff auf persistente Daten. Die Beans laufen in einem EJB-Container ab. Es gibt drei unterschiedliche Typen von EJBs:

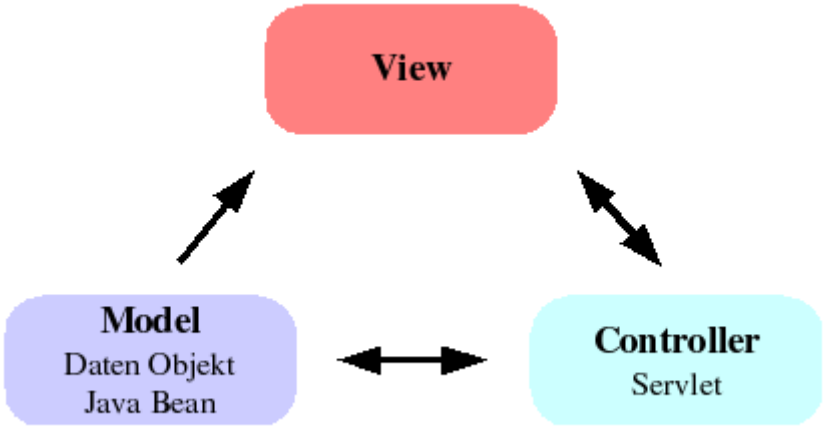
- Session-Beans, sowohl statusbehaftet als auch ohne internen Status, welche die Geschäftslogik implementieren und meistens vom Client zugreifbar sind
- Message-Driven-Beans für die Verarbeitung von JMS Nachrichten
- Entity-Beans für die Abbildung von persistenten Datenobjekten“(man unterscheidet zwischen Container Managed Persistenz CMP, bei der sich der EJB-Container um die Persistierung der Daten kümmert, und Bean Managed Persistenz BMP, bei der die Entity Bean selber für die Persistenz verantwortlich ist) „

Java Servlet API (Servlet)

Im Allgemeinen erlaubt das Java Servlet API die Erweiterung von Servern, deren Protokoll auf Anfragen und Antworten basiert. Primär werden Servlets im Zusammenhang mit dem Hypertext Transfer Protocol (HTTP) verwendet wo sie in einem Web-Container leben und Anfragen von Webbrowsern beantworten.“

„Als **Servlet** bezeichnet man im Rahmen der Java 2 Platform Enterprise Edition (J2EE) ein Java-Objekt, an das ein Webserver Anfragen seiner Clients delegiert, um die Antwort an den Client zu erzeugen. Der Inhalt dieser Antwort wird dabei erst im Moment der Anfrage generiert, und ist nicht bereits statisch (etwa in Form einer HTML-Seite) für den Webserver verfügbar.

Servlets stellen damit im Rahmen der J2EE-Spezifikation das Pendant zu einem CGI-Skript oder anderen Konzepten, mit denen dynamisch Web-Inhalte erstellt werden können (PHP etc.) dar. Sie sind Instanzen von Java-Klassen, die von der durch die Spezifikation definierten Klasse `javax.servlet.HttpServlet` abgeleitet wurden. Diese Instanzen werden bei Bedarf von einer Laufzeitumgebung, dem sogenannten *Web-Container* erzeugt und von ihm aus angesprochen. Der Webcontainer seinerseits kommuniziert mit dem Webserver."



Modell-View-Controller (MVC)

„Servlets werden oft nach dem Model-View-Controller (MVC)-Pattern zusammen mit JSP verwendet, wobei das Servlet den Controller und die JSP-Seite einen View repräsentiert.

Bei Verwendung der Servlet-Spezifikation und einer entsprechenden Webcontainer-Umgebung (z.B. Jakarta Tomcat) besteht die Implementierung einer dynamischen Webseite in folgendem: Es wird eine von `javax.servlet.HttpServlet` abgeleitete Klasse erstellt. Zwei Methoden der Superklasse werden überschrieben, die für die Verarbeitung der beiden Kernmethoden der HTTP-Spezifikation zuständig sind (*GET* und *POST*). In einer XML-Datei werden Metainformationen über das Servlet hinterlegt. Diese XML-Datei wird zusammen mit der kompilierten Klasse (sowie ggf. weiteren benötigten Klassen) in eine einzige Archiv-Datei zusammengeführt (sogenanntes *Web-Archiv*), dieses wiederum wird dem Webcontainer über eine von ihm bereitgestellte Funktionalität zur Verfügung gestellt (sog. *Deployment*).

Java Server Pages (JSP)"

„JavaServer Pages, abgekürzt **JSP**, ist eine von Sun Microsystems entwickelte Technologie, die im Wesentlichen zur einfachen dynamischen Erzeugung von HTML- und XML-Ausgaben eines Webservers dient. Sie erlaubt es Java-Code und spezielle JSP-Aktionen in statischen Inhalt einzubetten. Dies hat den Vorteil, dass die Logik unabhängig vom Design implementiert werden kann.

Die JSP-Syntax erlaubt es mittels spezieller XML-Tags (JSP-Aktionen) vordefinierte Funktionalität einzubinden. Diese JSP-Aktionen werden in so genannten Tag-Bibliotheken (Tag Libraries) als Erweiterung der HTML bzw. XML Tags definiert.

JavaServer Pages werden unter Verwendung eines speziellen JSP-Compilers in Java-Quellcode umgewandelt. Dieser Quellcode, der einem Java-Servlet entspricht, wird im Anschluss durch

den Java-Compiler in Bytecode umgewandelt. Die so erzeugten Java-Klassen können dann von einem Webserver mit entsprechender Servlet-Engine ausgeführt bzw. interpretiert werden.“

„Java Message Service (JMS)

ist eine API für die asynchrone Nachrichtenverarbeitung.

Java Database Connectivity (JDBC)

lässt die Anwendung auf eine relationale Datenbank mittels SQL zugreifen.

Java Transaction API (JTA)

erlaubt der Anwendung die Steuerung der Transaktionsverwaltung. JTA ist die Java-Schnittstelle zu Transaktionsmonitoren. Standardmäßig wird diese Schnittselle implementiert vom Java Transaction Service (JTS), welcher eine Schnittstelle zu CORBA Object Transaction Service (OTS) bietet.

Java Authentication and Authorization Service (JAAS)

Der "Java Authentication and Authorization Service" (JAAS) ist eine Java-API, die es ermöglicht, Dienste zur Authentifikation und Zugriffrechte in Java-Programmen bereitzustellen. JAAS implementiert ein Standard-"Pluggable Authentication Module" (PAM) und unterstützt durch dieses Modul einfach eine benutzerbasierte Autorisation.

JavaMail

erlaubt den Zugriff auf Mail Services, wie z.B. SMTP oder POP3.

Java API for XML Processing (JAXP)

hilft dem Entwickler bei der Bearbeitung von XML Dokumenten.

J2EE Connector Architecture (JCA)

dient dazu andere Systeme transparent zu integrieren.

Java Beans Activation Framework

bietet die Möglichkeit verschiedene Daten anhand des MIME-Headers zu erkennen.

“

Einige Implementierungen des J2EE-Standards

JBoss
Open Source J2EE Container
BEA WebLogic
IBM WebSphere
Sun Application Server 1.4
Oracle Application Server 10g
SAP Web Application Server

Weitere Informationen findet man auf der Original-Homepage http://java.sun.com/j2ee/

3.1.3 Probleme der EJB-Technologie

Aus vielen Berichten in Zeitschriften und im Internet wird klar, dass sich zwar J2EE als Standard durchgesetzt hat, aber die Entwicklung der EJB-Komponenten nicht so unkompliziert ist, wie die J2EE-Spezifikation verspricht.

Insbesondere der Einsatz der Entity Beans – speziell auch CMP – wird in der Praxis oft umgangen und nach Alternativen gesucht, die besser zu handhaben und performanter sind.
Für die passende Persistenzstrategie gibt es inzwischen eine Reihe von Alternativen neben Entity Beans, z.B. Standard-JDBC, Java Data Objects (JDO) oder das Hibernate-Framework. Letzteres hat inzwischen einen Status als Quasi-Standard neben EJB.

Auch in den Bereichen der Anwendungslogik und Web-Oberflächen werden Alternativen gesucht. Beispielsweise Das Spring-Framework mit seinem sogenannten „leichtgewichtigen“ Container oder Struts als Web-Framework, dass eine starke Verbreitung hat.

Das positive an diesen Entwicklungen ist, dass Alternativen zu teilen der J2EE von Sun teilweise aufgegriffen und die zukünftigen Standards beeinflussen.
Beispielsweise definiert EJB 3.0 eine Leightweight-Persistierung für sogenannte Plain Old Java Objects (POJO) – also einfache Java-Objekte – , die auch außerhalb von EJB-Containern nutzbar ist.
Und die Technologie der JavaServer Faces (s.u.) ist unter dem Einfluss des verbreiteten Open-Source-Framework Struts festgelegt worden. Hier wirkt sogar der Struts-Entwickler bei der Spezifikation führend mit.

Im weiteren Verlauf dieses Kapitels werden überblicksartig noch einige weitere Frameworks und Technologien vorgestellt.

Diese Umfrage soll unter anderem auch hinterfragen, welche Technologien tatsächlich verbreitet sind.

3.2 Zum Vergleich: Die Microsoft .NET-Technologie

Die Zusammenfassung aus [Wikipedia1] gibt einen Überblick der .NET-Technologie:

„**.NET** ist ein Satz von Softwaretechnologien des Softwareherstellers Microsoft, die als Gegenpol zu Sun Microsystems' ONE (Java) eingeführt wurden und neben einer virtuellen Laufzeitumgebung aus einem Rahmenwerk (Framework) von Klassenbibliotheken (API) und Diensten besteht, die als Basis für Eigenentwicklungen dienen. Die umfangreiche Programmierumgebung .NET soll eine neue Generation von Programmen und damit auch der Programmierung einläuten. Keine der verwendeten Technologien ist völlig neu, aber in der Gesamtheit ist **.NET** eine Innovation."

3.2.1 „Konzept

Die .NET-Plattform stellt mit der **Common Language Infrastructure** (CLI) eine Basis zur Ausführung von Programmen, die mit unterschiedlichen Programmiersprachen erstellt wurden, her. Dies wird durch die Verwendung einer (objektorientierten) virtuellen Maschine und die **Framework Class Library** (FCL) – einer gemeinsamen Klassenbibliothek – erreicht.

CLR, CIL, Garbage-Collection und Reflection

Die **Common Language Runtime** (CLR) ist die virtuelle Maschine (**VM**) von .NET und stellt somit die Laufzeitumgebung für verschiedene an .NET angepasste Hochsprachen zur Verfügung und kann als Pendant zur Java Virtual Machine bei Java gesehen werden. Die VM führt den standardisierten Zwischencode der **Common Intermediate Language** (CIL) aus, dem Pendant zum Bytecode bei Java. Die CIL hieß früher **Microsoft Intermediate Language** (MSIL), wurde aber im Rahmen der Standardisierung durch die ECMA umbenannt. Für die CIL wurde ein sprachübergreifendes System von objektbasierten Datentypen definiert, so dass auch in unterschiedlichen Sprachen geschriebene Programmteile auf gemeinsame Ressourcen zugreifen können. Das von Vorneherein im Gesamtkonzept berücksichtigte Nebeneinander mehrerer Programmiersprachen unterscheidet .NET von anderen auf Zwischencode basierenden Laufzeitumgebungen (z.B. *Java Virtual Machine*, siehe Java).

Das Besondere an der CLR ist weniger die technische Innovation als vielmehr die strategische (und langfristig vielleicht folgenreichste) Entscheidung des Marktführers Microsoft *für* ein laufzeit-basiertes System, das u.a. Programmierfehler vermindern helfen soll und *gegen* die traditionelle direkte Kompilierung in den Maschinencode des Zielsystems. Zusammen mit der Marktmacht von Java und dem Erfolg der sogenannten Skriptsprachen ist hiermit ein Trend zu identifizieren. Dieser stellt einen als historisch zu bezeichnenden Bruch mit der noch wenige Jahre zuvor, zumindest in der kommerziellen Softwareentwicklung, vorherrschenden Dominanz der direkt in Maschinencode kompilierenden Programmiersprachen (insbesondere C++ und C) dar. Anders ausgedrückt ist dies ein Schritt weg von der Performance zur Programmlaufzeit um jeden Preis und hin zur Erhöhung der Effizienz in der Programmentwicklung.

Mittels Reflection ist es insbesondere mit dem Emit()-Mechanismus möglich, zur Laufzeit Programmcode über ein Objektmodell zu generieren und es in-Memory in eine lauffähige Assembly zu überführen. Diese Technik wird unter anderem von ASP.NET genutzt sowie einigen anderen Mechanismen wie z. B. der Geschwindigkeitsoptimierung regulärer Ausdrücke. Zudem ist es außerordentlich einfach, zur Laufzeit Skriptcode dynamisch zu kompilieren und einzubinden.

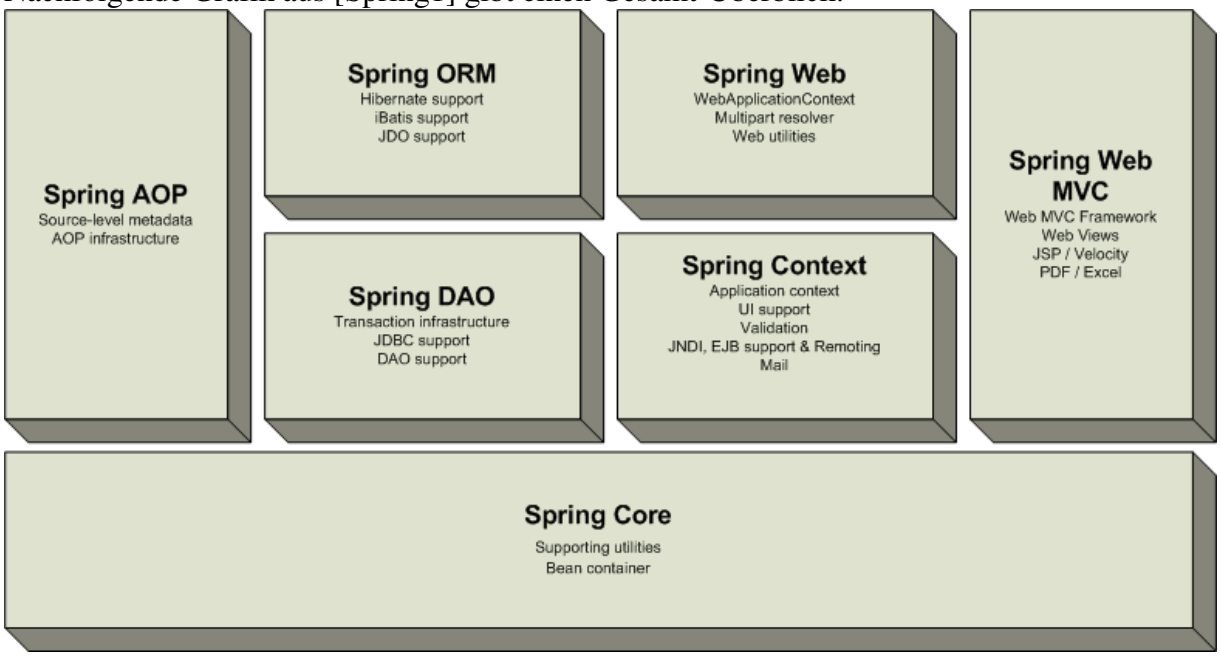
3.3 Das Java-Applikations Spring

Wie bereits oben erläutert gab es in der J2EE Praxis durchaus Probleme mit der EJB-Entwicklung z.B. mit einem komplexen Programmiermodell mit einem monolithischen Container oder mit der Tatsache, dass die Dienste des Containers laut Spezifikation nicht angepasst oder erweitert werden können.

Das Spring Framework bietet zur Lösung einen leichtgewichtigen Containers und stellt einen Mechanismus zur Verfügung, um Geschäftsapplikationen in eine J2EE-Umgebung zu integrieren.

3.3.1 Aufbau des Spring-Frameworks

Nachfolgende Grafik aus [Spring1] gibt einen Gesamt-Überblick:

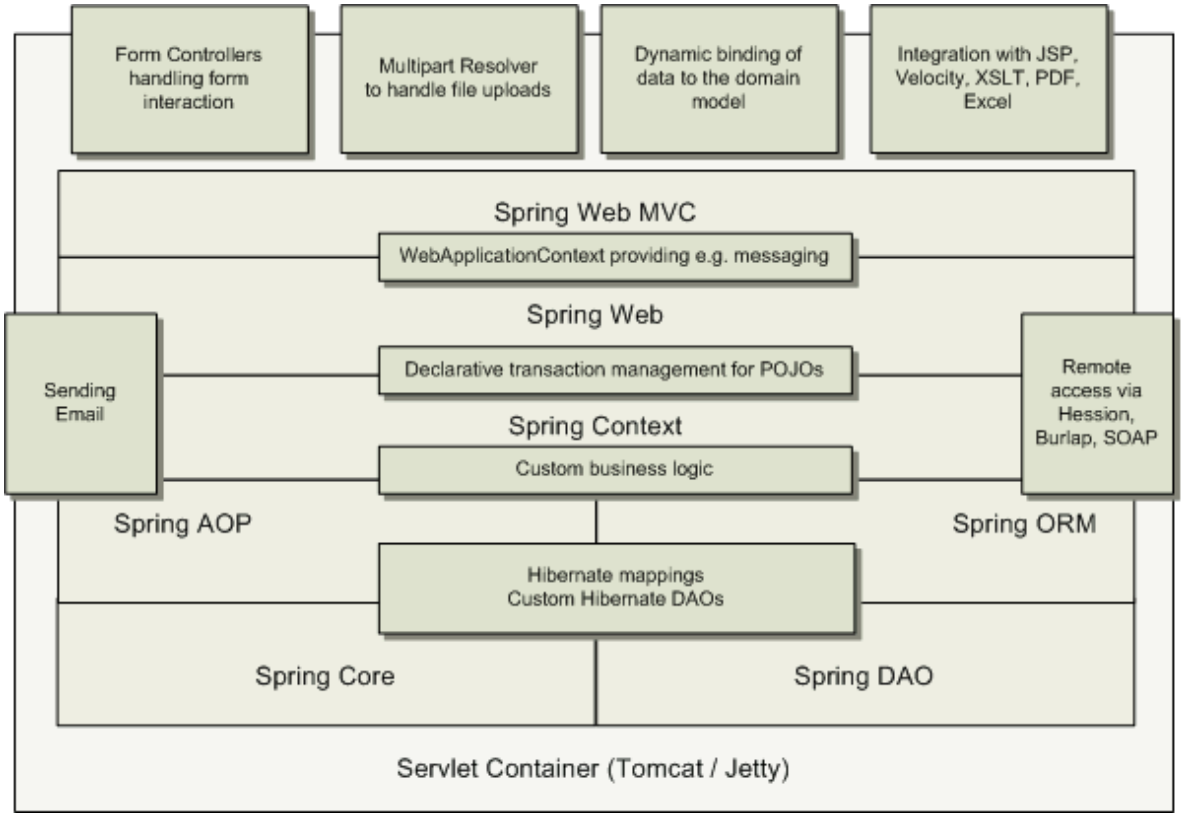


Erläuterungen zu den einzelnen Teilen von Spring finden Sie auf der nächsten Seite.

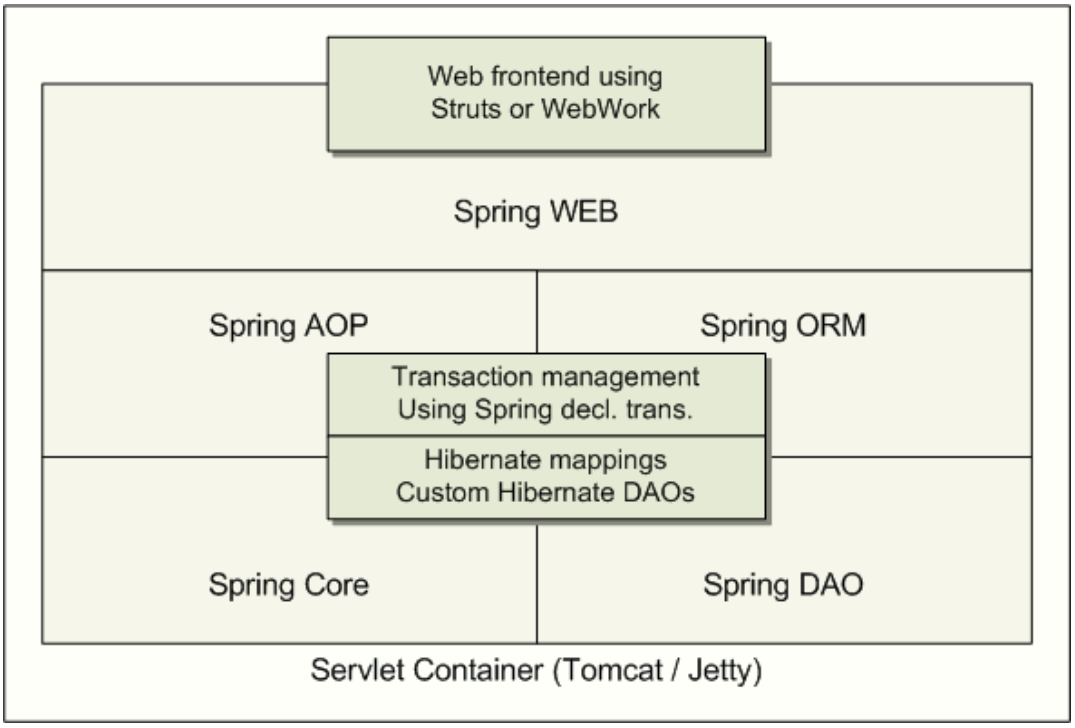
Quelle [Spring1]: „

- The *Core* package is the most fundamental part of the framework and provides the Dependency Injection features allowing you to manage bean container functionality. The basic concept here is the BeanFactory, which provides a factory pattern removing the need for programmatic singletons and allowing you to decouple the configuration and specification of dependencies from your actual program logic.
- On top of the *Core* package sits the *Context* package, providing a way to access beans in a framework-style manner, somewhat resembling a JNDI-registry. The context package inherits its features from the beans package and adds support for text messaging using e.g. resource bundles, event-propagation, resource-loading and transparent creation of contexts by, for example, a servlet container.
- The *DAO* package provides a JDBC-abstraction layer that removes the need to do tedious JDBC coding and parsing of database-vendor specific error codes. Also, the JDBC package provides a way to do programmatic as well as declarative transaction management, not only for classes implementing special interfaces, but for *all your POJOs (plain old java objects)*.
- The *ORM* package provides integration layers for popular object-relational mapping APIs, including JDO, Hibernate and iBatis. Using the ORM package you can use all those O/R-mappers in combination with all the other features Spring offers, like simple declarative transaction management mentioned before.
- Spring's *AOP* package provides an *AOP Alliance* compliant aspect-oriented programming implementation allowing you to define, for example, method-interceptors and pointcuts to cleanly decouple code implementing functionality that should logically speaking be separated. Using source-level metadata functionality you can incorporate all kinds of behavioral information into your code, a little like .NET attributes.
- Spring's *Web* package provides basic web-oriented integration features, such as multipart functionality, initialization of contexts using servlet listeners and a web-oriented application context. When using Spring together with WebWork or Struts, this is the package to integrate with.
- Spring's *Web MVC* package provides a Model-View-Controller implementation for web-applications. Spring's MVC implementation is not just any implementation, it provides a clean separation between domain model code and web forms and allows you to use all the other features of the Spring Framework like validation.

3.3.2 Anwendungsszenarien des Spring-Framework (Quelle [Spring1]):

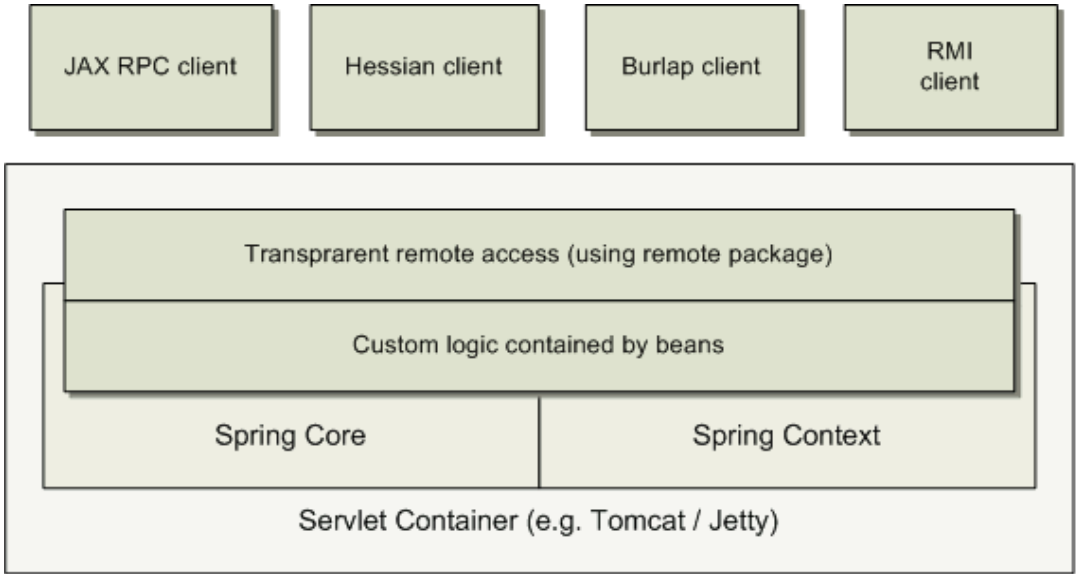


Typical full-fledged Spring web application



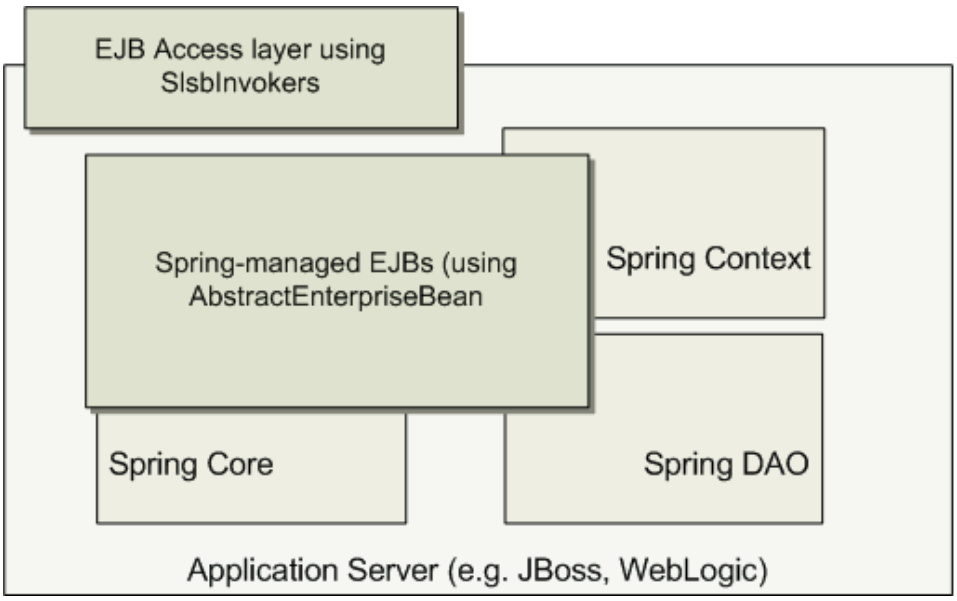
Spring middle-tier using a third-party web framework

.



Remoting usage scenario

.



EJBs - Wrapping existing POJOs

3.4 Eclipse (Open-Source IDE und Framework)

Eclipse als Open-Source IDE und zugleich Framework für die Entwicklung von Verteilten Systemen in Java hat in letzter Zeit rasante Verbreitung in der Softwareentwicklung gefunden. Insbesondere das Konzept der flexiblen Erweiterbarkeit durch Plugins, die von diversen Herstellern sowohl als Open-Source als auch als kommerzielle Plugins bereitgestellt wurden, hat die Verbreitung beschleunigt.

[wikipedia1] gibt – hier auszugswiese dargestellt – folgende Zusammenfassung zu Eclipse:

„Eclipse ist ein Framework, das meist als freie Entwicklungsumgebung (IDE) genutzt wird.

Bis einschließlich zur Version 2.1 war Eclipse als erweiterbare IDE konzipiert. Seit Version 3.0 ist Eclipse selbst nur der Kern, der die einzelnen **Plugins** lädt, die dann die eigentliche Funktionalität zur Verfügung stellen.

Diese Funktionalität nennt sich **Rich Client Platform (kurz RCP)** und basiert auf dem OSGi-Standard.

Sowohl Eclipse als auch die Plugins sind vollständig in Java implementiert.

Als GUI-Framework zur Erstellung der grafischen Oberfläche wurde **SWT**“ (Standard Widget Toolkit) „verwendet. Eclipse ist daher nicht plattformunabhängig, wird aber für 11 verschiedene Systeme und Architekturen bereitgestellt. Zur Darstellung der GUI-Komponenten basiert SWT ähnlich wie AWT auf den nativen GUI-Komponenten des jeweiligen Betriebssystems.

Die Plugins lassen sich durch den Download direkt in Eclipse von einem Update-Server oder durch einfaches Entpacken installieren.

Die Bezeichnung Eclipse ist der englische Begriff für eine Sonnenfinsternis (solar eclipse). Laut Erich Gamma, einem der Entwickler, soll der Name darauf hinweisen, dass mit Eclipse proprietäre Entwicklungsumgebungen in den Schatten gestellt werden.

Eclipse stellte den Nachfolger für IBM Visual Age dar. Die Entwicklung soll mehr als 40 Millionen Dollar gekostet haben. Der Quellcode für Eclipse wurde dann von IBM freigegeben. Die Verwaltung und Entwicklung von Eclipse wurde von der Eclipse-Foundation übernommen.“

„Standard Widget Toolkit (SWT)
„Das Standard Widget Toolkit (SWT) ist eine Bibliothek für die Erstellung grafischer Oberflächen mit Java.
SWT wurde im Jahr 2001 von IBM für die Entwicklungsumgebung Eclipse entwickelt und wird kontinuierlich gepflegt. SWT nutzt dabei im Gegensatz zu Swing die nativen grafischen Elemente des Betriebssystems - wie das AWT von Sun - und ermöglich somit die Erstellung von Programmen welche eine Optik vergleichbar mit "nativen" Programmen aufweisen. Allerdings leidet SWT auf einigen nicht-Windows Platformen unter Effizienzproblemen, da es viele Features eines Basistoolkits voraussetzt, welche wenn nicht vorhanden emuliert werden müssen (z.B. Z-Ordering auf GTK).“

„JFace
Das UI-Toolkit JFace setzt aus den von SWT gelieferten Basiskomponenten komplexere Widgets zusammen und läuft überall, wo SWT zur Verfügung steht. Eclipse ist die wohl bekannteste Anwendung, die JFace einsetzt.“

3.5 Das MVC-Web-Framework Struts (Open-Source)

Struts ist ein Open Source Framework zur Erstellung von Webapplikationen mit Standard Technologien wie JSPs, Servlets und Java Beans. Struts ermöglicht eine einfache Umsetzung des Model View Controller Paradigmas. Dies ist die offizielle Seite zum Struts-Framework : http://struts.apache.org/

Folgende sehr gute Zusammenfassung ist auszugsweise [wikipedia1] entnommen:

„**Struts** ist ein Open-Source-Framework für die Präsentationsschicht von Java-Web-Anwendungen. Es ist eines der bekanntesten Jakarta-Projekte und beschleunigt die Entwicklung von Web-Applikationen wesentlich, indem es HTTP-Anfragen in einen standardisierten Prozess verarbeitet. Dabei bedient es sich standardisierter Technologien wie JavaServlets, Java Beans, `Resource Bundles` und XML sowie verschiedener Jakarta-Commons-Pakete.

Für den Entwickler bedeutet das, dass viele applikationsrelevante Funktionen bereits implementiert und einsatzbereit sind. Struts wurde von versierten Entwicklern geschaffen und wird bereits in hunderten Applikationen eingesetzt. Das garantiert ein solides Framework mit den besten existierenden Softwareentwicklungsstrategien. Struts wurde von Craig McClanahan im Jahr 2000 entwickelt. Seitdem arbeitet eine ständig wachsende Entwicklergemeinde an der Verbesserung des Frameworks. Aktuelle Version ist 1.2.

3.5.1 Aufbau

Das Struts-Framework basiert auf dem Paradigma des Model-View-Controllers. Dabei sind die Komponenten von Struts in der View und dem Controller anzusiedeln. Typische Model-Funktionalitäten bietet das Framework nicht. Das Framework beinhaltet zur Zeit ca. 300 Java-Klassen, die in 8 Kernpakete eingeteilt sind.

3.5.2 Theoretischer Ansatz

Model-1-Architektur mit JavaServlets/JSP

Nachdem die Technologie von Java-Servlets eingeführt wurde, erkannte man schnell, dass das Erstellen von HTML-Seiten mit Servlets zeitaufwendig und die Weiterentwicklung von Views sehr mühsam sein kann. Hinzu kam, dass alle drei Komponenten des MVC in einem Servlet realisiert wurden. Daraufhin wurde die JSP-Technologie eingeführt, die vor allem dem Webdesigner die Arbeit leichter machte und sich in kurzer Zeit durchsetzte. Daraus entwickelte sich die Model-1-Architektur, die beschreibt, dass Datenhaltung, Geschäftslogik und View in einer JSP anzusiedeln sind, was jedoch bei komplexer Anwendungslogik zu neuen Problemen führt: Die JSP wird in kürzester Zeit unüberschaubar und lässt sich nur durch Entwickler warten, die sowohl HTML- als auch Java-Kenntnisse besitzen. Zwar mildert die Verwendung von Tag-Bibliotheken wie z. B. JSTL, dieses Problem etwas, weil kein Java-Code mehr benötigt wird, das konzeptionelle Grund-Problem der Vermischung von Präsentation mit fachlicher Funktionalität bleibt jedoch bestehen.

Model-2-Architektur mit Struts

Die aufgeführten Problematiken haben dazu beigetragen, dass man ein neues Konzept entwickeln musste, das die einzelnen Komponenten klarer voneinander abgrenzt und Java-Servlets und JSP gleichermaßen einbindet. Es entstand die Model-2-Architektur für Webanwendungen, die ihren Einsatz im Struts-Framework gefunden hat. Eine wichtige Rolle spielt hierbei das *ActionFirst Pattern*. Das bedeutet, dass View-Komponenten, die durch JSP abgebildet werden, ausschließlich über Struts-Actions (Servlets) referenziert werden, niemals über einen HTTP-Request. Dabei wird jeder HTTP-Request von einer Controller- Komponente (Servlet) entgegengenommen, bei Struts dem so genannten ActionServlet. Nach einem vom Benutzer initiierten HTTP-Request wird eine Struts-Action ausgeführt, die eine Verbindung zur Geschäftslogik herstellt und diese aktualisiert. Die Ergebnisse der Ausführung werden anschließend in dem HTTP-Request als Attribute abgelegt. Danach wird über einen Action-Forward die View aufgerufen und mit den im HTTP-Request abgelegten Daten gefüllt. Befindet sich in einer View-Komponente ein HTML-Formular, so wird dieses über ein Java-Bean mit Werten aus dem Model gefüllt. Die Model-2-Architektur bietet sich hier als Regelwerk an (u.a. ist die Kommunikation von Views untereinander, wie auch die feste Kodierung von Flusssteuerung in JSP, verboten).

3.5.3 Praktische Umsetzung

Ziel: Trennung von Präsentation, Datenhaltung und Anwendungslogik. Dies erhöht die Übersicht und die Wartbarkeit. Deshalb definiert man in Struts 3 Hauptkomponenten:

JSP: Präsentation
Action: Anwendungslogik (Holen und Aktualisieren der Daten)
FormBean: Datenhaltung (Inhalte von Textfeldern, Kollektionen für Listen usw.)

Alle 3 Komponenten werden in der zentralen Konfigurationsdatei von Struts (`struts-config.xml`) miteinander verknüpft und können somit miteinander kommunizieren.

Beispiel:

```
<struts-config>
    <form-beans type="org.apache.struts.action.ActionFormBean">
        <form-bean name="RegisterForm" type="bar.foo.web.RegisterForm" />
    </form-beans>
    <action-mappings type="org.apache.struts.action.ActionMapping">
        <action path="register"
                name="RegisterForm"
                type="bar.foo.web.RegisterAction"
                input="/web/register.jsp"
                scope="request"
                validate="true">
            <forward name="success" path="/home.do" />
            <forward name="failure" path="/error.do" />
        </action>
    </action-mappings>
</struts-config>
```

Hier wurden register.jsp, RegisterAction, RegisterForm verknüpft.

Die Präsentation

Ein Paradigma ist, in der JSP so wenig Java-Code wie möglich zu verwenden. Um dies zu ermöglichen, sollte man die von Struts mitgelieferten Tag-Libs einsetzen, die einem beim

Auslesen der Form-Bean helfen. Geschrieben und gelesen wird hier jeweils nur aus der im Request liegenden Form-Bean.

Beispiel:

```
<html:text name="RegisterForm" property="emailAddress" size="20"/>
```

Hier wird das Attribut "emailAddress" der FormBean "RegisterForm" über seine Setter-Methode beim Submit der Form mit dem Inhalt des Text-Formulars gefüllt.

Tiles

Struts besitzt ebenfalls mächtige Template-Erweiterungen, Tiles genannt. Sie ermöglichen dem Entwickler, seine Webseiten komponentenbasiert aufzubauen (Header, Footer, Content usw.). Tiles geht dabei weit über das statische Inkludieren von JSP-Inhalten hinaus.

Die Datenhaltung

Die Form-Bean ist eine normale Java Bean, die alle benötigten Daten für die JSP und die Action besitzt. Sie ist die einzige Schnittstelle zwischen diesen beiden Komponenten. Die Form-Bean wird unter dem in der struts-config festgelegten Namen im Request gespeichert. Das Struts-Framework liest die Daten über ihre Getter- und Settermethoden aus.

Die FormBean enthält auch eine validate-Methode, die (wenn in der config aktiviert) die Daten der FormBean prüft, bevor sie zur Action geschickt werden. Desweitern enthält sie auch noch eine reset-Methode.

Die Action

Die Action ist diejenige Komponente, die mit dem Backend kommuniziert und von ihr Daten holt und sie auch wieder dahin schreibt. Sie enthält meist weitere Prüf-und Auswertungsmechanismen. Sie speichert die Daten in der Form-Bean ab und liest sie nach dem Submit wieder aus. Daher gilt es als Unsitte, die Daten hier direkt in den Request zu speichern; dazu ist schließlich die Form-Bean da.

3.5.4 Kritik

Neben den Vorteilen von Struts existieren auch folgende Nachteile.

Die Programmierenden müssen selbst darauf achten, dass ihre Actions sicher bezüglich Nebenläufigkeit sind. Dies ist deshalb riskant, weil eine Action gleichzeitig durch mehrere parallele Threads des Web-Servers aufgerufen werden kann.
Actions sind nur als abstrakte Klassen definiert, nicht als Schnittstellen. Dies ist von Nachteil, weil beim Programmieren Implementierungsdetails beachtet werden müssen, was unübersichtlich und unflexibel ist. (Siehe hierzu auch das Entwurfsmuster Kommando.)
ActionForms bilden häufig große Teile der fachlichen Klassenstruktur nach. Dies erhöht einerseits den Aufwand beim Erstellen der Anwendung, andererseits aber auch die Wartungskosten.
“

3.6 Java Server Faces

Die neueste Technologie von Sun für Web-Darstellung wurde bei Heise-Online im Frühjahr 2004 folgendermaßen angekündigt:

„Die vom Java Community Process verabschiedete Spezifikation Java Server Faces soll die Entwicklung von Anwendungen für das Web vereinfachen. Mit dem neuen Framework lassen sich Bedienoberflächen für Web-Anwendungen aus vorgefertigten Komponenten zusammenbauen, ähnlich wie Desktop-Applikationen mit dem GUI-Framework Swing.

Die Java Server Faces stellen Entwicklern eine Sammlung von Standard-Bedienelementen und Mechanismen für deren Kopplung mit der eigenen Server-Software zur Verfügung. Den HTML-Code für den Browser generiert das Framework dabei selbstständig; alternativ kann man die Oberfläche auch in Form von WML-Seiten fürs WAP-Handy oder in anderen Formaten ausgeben lassen.

In neuen Entwicklungsumgebungen wie Suns Java Studio Creator (vormals Project Rave) sollen sich Web-Applikationen in Zukunft mit grafischen Editoren weitgehend zusammenklicken lassen. Die Java Server Faces konkurrieren dabei mit Microsofts ASP.NET Web Forms.“

Der Entwickler des Struts-Frameworks Craig McClanahan ist gleichzeitig der „Co-Spezification Lead“ der JavaServer Faces“. Konzeptionell kann man JavaServer Faces als eine Mischung aus Struts und Java Swing bezeichnen. Auch hier geht Sun also den Weg positive Einflüsse aus der Open-Source-Welt in neuen Standards zu berücksichtigen.

Bei [SunJSF] findet sich folgender Technologie-Überblick:

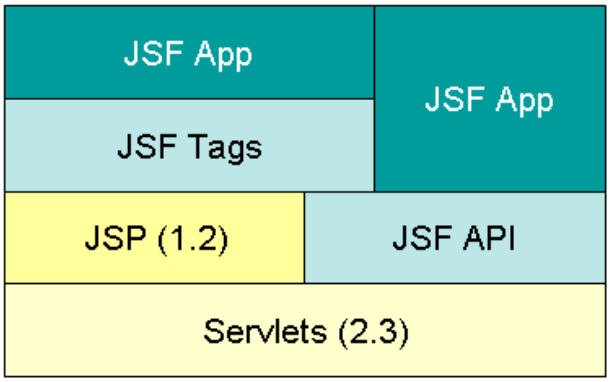


Figure 1: High-level architecture of JSF

„The JSF technology consists of two main components:

1. Java APIs to represent UI components, manage state, handle events, and validate input. The API has support for internationalization and accessibility.
2. Two JSP custom tag libraries for expressing user interface (UI) components within a JSP page, and for wiring components to server-side objects. Page authors can easily add UI components to their pages.

Figure 2 shows the relationship between the client, the server, and JSF.”

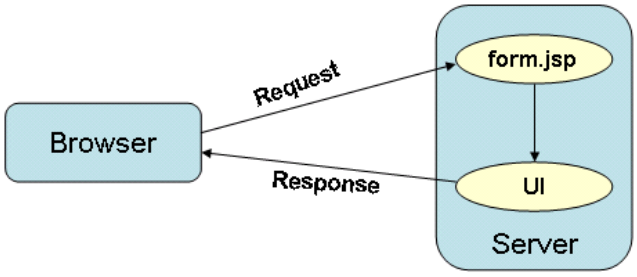


Figure 2: The UI runs on the server

3.7 Java Data Objects (JDO)

Die durch den Java Community Process verabschiedete Java Data Objects-Spezifikation (JDO) verspricht die transparente Persistierung von einfachen Java-Objekten abstrahiert vom zugrunde liegenden Speichermedium, ist also nicht festgelegt auf relationale Datenbanken.
JDO bietet eine eigene Query Language, woraus allerdings eine schlechte SQL-Unterstützung, resultiert, was ggf. nachteilig sein kann. .

Da Sun mit EJB 3.0 auch einen Mechanismus etabliert, der die Persistierung von einfachen Java Objekten (Plain Old Java Objects POJO) außerhalb von EJB-Containern unterstützt, und EJB 3.0 vor JDO favorisiert, ist fraglich, ob JDO in der Zukunft weiter genutzt wird.

Die Umfrage soll Anhaltspunkte über die Verbreitung von JDO liefern.

3.8 Das Persistenzframework Hibernate (Open-Source)

Das Hibernate-Framework bietet mittels objektrelationem Mapping (O/R-Mapping) die Möglichkeit einfache Java Objekte persistent in relationalen Datenbanken zu speichern.

Hibernate bietet ebenfalls eine eigene Query Language.

Hibernate ist auf SQL-Datenbanken beschränkt, zu denen ein JDBC-Treiber verfügbar ist. Dies sind allerdings im wesentlichen alle heute relevanten SQL-Datenbanken, insoweit stellt dies keinen Nachteil dar.

Im Gegensatz zu JDO bietet Hibernate eine bessere Unterstützung der relationalen Datenbanken. Natürlich wird auch ein Transaktionsmanagement geboten.

Das O/R-Mapping ist mittels XML sehr flexibel gestaltet.

Dies sind vermutlich auch die Gründe für die starke Verbreitung von Hibernate.

In der Umfrage soll die tatsächliche Verbreitung erfragt werden.

3.9 Das Publishing Framework Cocoon (Open-Source)

Ein weiteres Framework, das von Teilnehmern der Umfrage eingesetzt wird, ist das Publishing-Framework Cocoon.

Die folgende Zusammenfassung ist wieder [Wikipedia1] entnommen:

„Als Cocoon wird das XML Publishing System der Apache Foundation bezeichnet. Dieses Framework wurde geschaffen, um Daten in XML-Form zu speichern und mittels XSL formatiert auszugeben. Als Ausgabeprodukte von XML-Daten können XHTML, PDF oder RTF stehen.

„Cocoon ist ein „Publishing Framework Servlet“, dass mittels seines modularen Reaktorkonzepts ein XML-Quelldokument je nach anfragendem Client in ein beliebiges Zielformat transformiert“ (Übersetzung: cocoon.apache.org)

Cocoon wurde 1998 von dem italienischen Student Stefano Mazzocchi geschrieben (angefangen), während er den Science Fiction-Film Cocoon sah – daher der Name für das System.

Technisch basiert Cocoon auf der Servlet-Technologie.“

„Der Quellcode von Cocoon fällt unter die Apache License und ist somit als freie Software zu betrachten.

Mittels Cocoon lassen sich nicht nur dynamische Web-Anwendungen erstellen, auch die lokale Benutzung ist möglich. Apache Lenya ist ein Content Management System, welches auf Cocoon aufbaut.
Cocoon baut außerdem auf dem Konzept der strikten Trennung von Inhalt und Logik auf, daher sind diese auch in unterschiedlichen Dateien gespeichert.
“

3.10 Weitere Open-Source-Frameworks

Nachfolgende Kurbeschreibungen liefert [Objectone1]:

3.10.1 Turbine

„Turbine ist ein Servlet-basiertes MVC Framework zum Entwickeln von sicheren Webapplikationen. Dieses, ebenfalls von der Jakarta Group entwickelte Framework, läßt im Vergleich zu Struts auch andere Technologien als View-Komponenten zu (z.B. Velocity, WebMacro, JSP).

Besondere Merkmale

- MVC 2 Ansatz
- Benutzerverwaltung
- Sessionverwaltung
- Internationalisierung
- automatische Validierung von Formularen
- Implentierung mit anderen Template Systemen wie JSPs, Velocity oder WebMacro
- Persistenzschicht mit Apache Torque (OR Mapping mit XML Datei)
- globales Caching
- J2EE Integration

Turbine verfügt über Services, welche die Entwicklung von Web-Applikationen im Bezug auf alle drei Komponenten der MVC-Architektur unterstützen. So ist beispielsweise ein OR-Mapping-Service zur Modellerzeugung und ein Service zur Template-basierten Erzeugung von Views verfügbar. Ergänzend stehen ein ausgeklügeltes Benutzer-Konzept, Security-Services, Logging-Services, Pool-Services und weitere Integrationsmöglichkeiten zur Verfügung. Das Turbine-Framework liefert damit die wichtigsten Komponenten für viele Web Applikationen.“

3.10.2 Velocity

„Velocity ist eine Java-basierte Template Engine. Sie ist Teil des Apache Jakarta Projektes. Velocity wurde entwickelt, um eine strikte Trennung zwischen Geschäftslogik und Darstellung zu erzeugen. Die Template Sprache von Velocity beinhaltet mächtige, aber dennoch leicht verwendbare Sprachelemente.

Besondere Merkmale

- Trennung von Logik und Darstellung
- klare und sehr einfache Syntax für Template Entwickler
- einfache Schnittstelle für Programmentwickler
- da Template und Code separat gehalten werden, kann unabhängig entwickelt und gewartet werden
- die Template Engine lässt sich einfach in eine Java-Anwendung integrieren

“ …

„Der wichtigste Einsatzbereich von Velocity ist die Servlet-basierte Webprogrammierung. Weiter wird es in der SQL- und Java-Code-Erzeugung eingesetzt, oder in der XML Verarbeitung bzw. Transformation. Velocity gilt immer dann als hilfreich, sobald der Einsatz von Datenformatierung oder Präsentation ansteht.“

3.11 Test-Framework (JUnit)

Nicht unerwähnt bleiben soll an dieser Stelle auch der Quasi-Standard JUnit zur Erstellung von Testklassen.

Das JUnit Framework bietet – wie den meisten Entwicklern bekannt sein sollte – die Möglichkeit besonders für den automatisierten Unit Test einheitliche Testklassen auf Basis des JUnit-Frameworks zu erstellen.

Ziel ist die eher testgetriebene Softwareentwicklung, d.h. die Tests sollten gemäß den Anforderungen zuerst festgelegt werden, so dass von Anfang an systematisch jede Klassegetestet werden kann.

Wieweit die Realität und der Stress der Entwickler dem entgegenstehen, soll die Umfrage hinterfragen.

Anmerkung:
Die hier angesprochenen Test können keinesfalls den echten Test der Anwendung durch den fachlichen Anwender ersetzen. Denn Testklassen decken keine konzeptionellen Fehler des Entwicklers auf, da sie vom Entwickler geschrieben sind.

4 Vorbereitung der Umfrage

4.1 Vorgehensweise

Zur konkreten Durchführung der Umfrage mussten einerseits Aufbau und Inhalt der Umfrage festgelegt werden. Wobei folgende Schritte erforderlich waren:

1. Umfang der Umfrage festlegen,
2. Strukturierung und Aufbau der Umfrage
3. Sammeln der Fragen
4. Typisierung der Fragen (Einfachauswahl, Mehrfachauswahl, Eingaben, Bewertungsfragen)
5. konkrete Ausformulierung der Fragen

Und andererseits war die technische und organisatorische Umsetzung zu klären.

Konkret waren dies folgende Fragen:

- Wie soll die Umfrage durchgeführt werden Online- oder in Papierform?
- Damit verbunden, wie werden die Ergebnisse ausgewertet?
- Welche Teilnehmer kommen in Frage und wie werden diese kontaktiert (Brief, email, ..)
- Wieviel Zeit nimmt die Umfrage in Anspruch

Nachfolgende Kapitel 4.2 bis 4.4 beschreiben die Details zum Vorgehen.

4.2 Aufbau der Umfrage

Im Kapitel 2 wurden bereits diverse Fragenstellungen angerissen, die sich beim Einsatz von Java-Frameworks in verteilten Systemen stellen.
Grob lassen sich die Inhalte der Umfrage in folgende Kategorien zusammenfassen:

1. Fragen zu Gründen und Einflussfaktoren für den Frameworkeinsatz:

- Fragen zur technischen Infrastruktur der verteilten Systeme:
  Teilweise steht diese bereits vor dem Einsatz des Framework fest (z.B. gegebene Netztwerktopologie, bestehende Hardware etc.) und teilweise wird sie im Zusammenhang mit dem Einsatz neuer Softwaretechnologien neu geplant wird.
- Fragen zu den Einflussfaktoren bezüglich der Entscheidung für Frameworktechnologien

2. Fragen zum konkreten Einsatz von Java-Frameworks in verteilten Systemen:

- Was leistet das Framework?
- Welche Technologien werden eingesetzt?

3. Welche Softwareentwicklungsstandards werden umgesetzt:

- Testverfahren
- Standardisierung der Entwicklung der Architekturschichten
- OO Modellierung

4. Wie wird die Qualität der fertigen bzw. bereits eingesetzten Frameworks bewertet

- Stabilität und Sicherheit
- Verteilungs- und Netzwerkaspekte
- Benutzerakzeptanz
- Wirtschaftlichkeit

Den genauen Inhalt der Umfrage finden Sie in Kapitel 5.

Ein Ausdruck der Online-Version finden Sie in Anlage 1.

4.3 Technische Umsetzung der Umfrage (Internet/Online)

Bei der Frage, wie die Umfrage durchgeführt werden soll, entschied ich mich für eine Internet-Umfrage, da hier eine systematische Auswertung ohne große „Handarbeit“ möglich ist.
Im Vorwege habe ich mich kurz in die Technik von Online-Fragebögen eingearbeitet und die technischen Möglichkeiten an der FernUniversität Hagen zur Umsetzung geprüft.

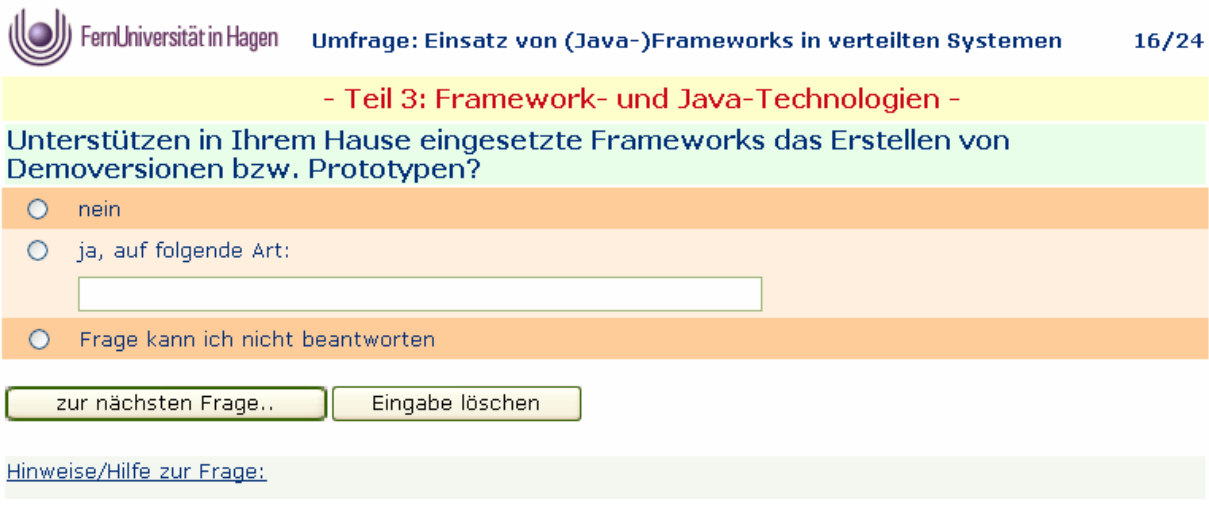
Als einfachste und dort am besten unterstützte und zugleich sichere Methode erwies sich die Nutzung von HTML-Forms und PHP mit einer Speicherung der Daten in einer selbst definierten MySql-Datenbank. Die Speicherung der Daten in Dateien wurde aus Sicherheitsgründen und wegen der besseren Bedienbarkeit und Auswertung zugunsten einer Online-Datenbank verworfen. Als Datenbank kam nur MySql in Frage, da diese als einzige an der Fernuni kostenlos verfügbar ist. Als Scriptsprachen wurden Perl und PHP getestet.

Die Entscheidung für PHP fiel auf Grund von Problemen im Zusammenspiel von Perl und MySql auf dem Fernuni-Server.

Folgende Anforderungen waren aus meiner Sicht zu erfüllen:

- Die Fragen sollten in einer Datenbanktabelle abgelegt werden und daraus mittels eines PHP-Programms ausgelesen und in HTML dargestellt werden.
- Die Anwendung der Online-Umfrage sollte selbstverständlich Multi-User-fähig sein.
- Die Ergebnisse/Antworten je Benutzer sollten ebenfalls in einer Datenbank-Tabelle gespeichert werden, um die spätere programmgesteuerte Auswertung zu ermöglichen.
- Den Teilnehmern sollte einerseits Annonymität zugesichert werden können, andererseits sollte aber auch ein gewisses Maß an Seriosität durch eine Online-Registrierung gegeben sein.
- Die Ergebnisse sollten jederzeit ausgewertet werden können.
- Die Anwender sollen als Anreiz an einer Verlosung von einigen Amazon-Gutscheinen teilnehmen können und ggf. später das Umfrage-Ergebnis erhalten können. Dazu war die optionale Erfassung der email-Adresse vorzusehen.
- Die Bearbeitungszeit sollte im Normalfall 15 bis max. 30 Minuten betragen.
- Es soll ein optisch ansprechendes Layout verwendet werden (bessere Rückläuferquote).

Die obigen Anforderungen wurden mittels eines PHP-Programms umgesetzt
Die Umfrage ist als Anlage 1 dieser Arbeit beigefügt. Anbei ein Beispiel für das Layout:



Die Anforderungen wurden mit Hilfe folgender Klassen bzw. Datenbanktabellen umgesetzt:

Tabelle 1: Fragentexte zur Umfrage

Feld	Typ	Attribute	Null	Standard	Extra
TB01_ID	int(5)	UNSIGNED	Nein		auto_increment
TB01_UmfNr	int(3)	UNSIGNED	Nein	1	
TB01_Kategorie	char(2)		Nein	AA	
TB01_FrageNr	int(5)	UNSIGNED	Nein	1	
TB01_Fragetext	varchar(255)		Nein		
TB01_ Kommentar	varchar(255)		Ja	NULL	
TB01_Zusatzinfo	varchar(255)		Ja	*NULL*	
TB01_Type	char(1)		Nein		
TB01_Verzweigung	char(3)		Nein	001	
TB01_Layout	char(1)		Nein		
TB01_AnzFelder	char(2)		Nein	01	
TB01_Antwort1	varchar(255)		Ja	*NULL*	
TB01_Antwort2	varchar(255)		Ja	*NULL*	
TB01_Antwort3	varchar(255)		Ja	*NULL*	
TB01_Antwort4	varchar(255)		Ja	*NULL*	
TB01_Antwort5	varchar(255)		Ja	*NULL*	
TB01_Antwort6	varchar(255)		Ja	*NULL*	
TB01_Antwort7	varchar(255)		Ja	*NULL*	
TB01_Antwort8	varchar(255)		Ja	*NULL*	
TB01_Antwort9	varchar(255)		Ja	*NULL*	
TB01_Antwort10	varchar(255)		Ja	*NULL*	
TB01_Antwort11	varchar(255)		Ja	*NULL*	
TB01_Antwort12	varchar(255)		Ja	*NULL*	

Tabelle2: Umfrageergebnisse

Feld	Typ	Attribute	Null	Standard	Extra
TB02_Key_	int(5)		Nein		auto_increment
TB02_UmfrNr	int(3)		Nein	1	
TB02_FrageNr	int(5)		Nein	1	
TB02_AntwNr	int(2)		Nein	1	
TB02_TeilnNr	varchar(12)		Nein	000000000001	
TB02_Auswahl	int(2)		Ja	*NULL*	
TB02_Eingabe	varchar(255)		Ja	*NULL*	
TB02_Zeitstempel	datetime		Nein	0000-00-00 00:00:00	

Tabelle 3: Umfrageteilnehmer - Statistische Unternehmensdaten (annonym)

Feld	Typ	Attribute	Null	Standard	Extra
TB03_Key	int(5)		Nein		auto_increment
TB03_TeilnNr	varchar(12)		Nein	000000000001	
TB03_STANDORTE	varchar(4)		Nein		
TB03_MITARBEITER	varchar(5)		Nein		
TB03_ENTWICKLER	varchar(5)		Nein		
TB03_TAETIGKEIT	varchar(60)		Nein		
TB03_SOFTWAREART	varchar(255)		Nein		
TB03_BRANCHE	varchar(255)		Nein		
TB03_PROGRSPR	varchar(255)		Nein		
TB03_TOOLS	varchar(255)		Nein		
TB03_APLSRV	varchar(255)		Nein		

Tabelle 4: Liste der Emails (Angabe optional) für Verlosung und Ergebnisversand

Feld	Typ	Attribute	Null	Standard	Extra
TB04_Key	int(5)		Nein		auto_increment
TB04_EMAIL	varchar(255)		Nein		
TB04_STUDIE	char(1)		Nein		
TB04_VERLOSUNG	char(1)		Nein		

Tabelle 5: Registrierung Unternehmen, Adresse, etc.
(nur zur Kontrolle, keine Datenauswertung)

Feld	Typ	Attribute	Null	Standard	Extra
TB05_Key	int(5)		Nein		auto_increment
TB05_Reg1	varchar(12)		Nein		
TB05_Reg2	varchar(255)		Nein		
TB05_Reg3	varchar(255)		Nein		
TB05_Reg4	varchar(255)		Nein		
TB05_Reg5	varchar(255)		Nein		
TB05_Reg6	varchar(255)		Ja	NULL	

4.4 Ermittlung von und Kontakt zu Teilnehmern (Firmen)

Die Ermittlung von Mitarbeitern aus Softwarehäusern und IT-Abteilungen in großen Unternehmen, die als Teilnehmer in Frage kamen, war recht aufwendig.

4.4.1 Kriterien zur Auswahl der Teilnehmer

Folgende Anforderungen an die Auswahl der Teilnehmer sollten realisiert werden:

- In den ausgewählten Unternehmen sollten Java-Frameworks eingesetzt oder selbst entwickelt worden sein und Java eine signifikante Rolle spielen.
- Es sollte eine möglichst Breite Streuung der Branchen (bei Softwarehäusern, Branchen der Kundenkreise) geben.
- Es sollte eine möglichst Breite örtliche Streuung der Unternehmen/Kunden geben.
- Es sollten unterschiedlich Große Unternehmen beteiligt sein
- Es sollte mindestens 25 Unternehmen und davon mindestens 50% Softwarehäuser mit großen Kundenkreisen als Teilnehmer erreicht werden. Wunschziel waren mehr als 50 Teilnehmer.

4.4.2 Strategien und Quellen zur Teilnehmerauswahl

Folgende Quellen zur Auswahl der Teilnehmer wurden verwendet:

- Bekannte große Softwarehäuser wurden angeschrieben, von denen vermutet wurde, dass Java-Framework-Entwicklung für verteilte Systeme betrieben wird.
- Bekannte große Unternehmen mit IT-Abteilungen aus Versicherungen, Banken, Krankenkassen wurden angeschrieben in der Annahme, dass hier in jedem Fall das Thema Java eine Rolle spielen sollten.
- Große Firmen wurden aus Jobguides z.B. für Hamburg, Stuttgart, München ermittelt.
- In Jobbörsen wurden Stellenangebote für Java-,J2EE-Architekten und Entwickler gesucht und per Internet-Recherche versucht zu prüfen, ob die Firmen als Teilnehmer in Frage kommen
- Firmen mit Frameworkangeboten (Softwarehäuser) wurden per Internetrecherche auf deren Homepages ermittelt.
- Partner und Kunden von Frameworkherstellern wird ebenfalls per Internet recherchiert.

Insgesamt war der Rücklauf zunächst schleppend. Viele Teilnehmer wurden zur Erinnerung zweimal kontaktiert (per email).

Der Zeitraum der Umfrage wurde aus diesem Grund auf Antrag entsprechend verlängert.

4.4.3 Vorgehensweise

Um eine seriöse Zusammenstellung der Teilnehmer zu erhalten, sollten diese nach bestimmten Kriterien vorher ausgewählt und möglichst persönlich angeschrieben werden.
Es wurden insgesamt 320 Firmen teilweise zunächst per Brief (ca. 150) und später per email an spezielle Ansprechpartner angeschrieben. Als Anreiz wurde die Möglichkeit geschaffen, den Teilnehmern später die Umfrage-Ergebnisse per email zuzusenden und es wurden kleinere Gewinne (Amazon-Gutscheine im Wert von 20€ per Losverfahren) angeboten.
Es wurden insgesamt **35 Unternehmen bis Ende August** als Teilnehmer gewonnen, was einer Rücklaufquote von etwas über 10% entspricht und angesichts der schwierigen Ermittlung der Teilnehmer als sehr gut angesehen werden kann. Die obigen Kriterien wurden erfüllt.

Hinweis: Da den Teilnehmern Anonymität zugesichert wurde, ist eine Liste der Teilnehmer nicht Bestandteil dieser Arbeit, kann aber ggf. vom Prüfer bei mir eingesehen werden.

Nachfolgender Text des Anschreiben wurde - teilweise individualisiert - verwendet:

Umfrage zum Einsatz von Java-Frameworks in Verteilten Systemen

---

FernUniversität in Hagen U.Stegmann, Am Hünenstein 10, 20535 Hamburg

[Adresskopf des Unternehmens]

Ihre IT-Experten sind gefordert, profitieren Sie später vom Umfrage-Ergebnis.

Hagen/Hamburg, 18.07.2005

Guten Tag Herr [Ansprechpartner, falls ermittelt]…,

im Rahmen einer studentischen Abschlussarbeit führe ich eine Umfrage zum Thema „Einsatz von Java-Frameworks in Verteilten Systemen" in ausgewählten Softwarehäusern und IT-Abteilungen von größeren Unternehmen durch.

Ziel der Umfrage ist es, näher zu untersuchen, was ein Java-Framework für Verteilte Systeme nach Meinung von IT-Experten und IT-Entscheidern verschiedener Unternehmen leisten sollte und wie die jeweiligen konkreten Umsetzungen aussehen.

Hierfür habe ich Ihr Unternehmen als interessanten Teilnehmer ausgewählt und würde mich freuen, wenn Sie an der Online-Umfrage teilnehmen.

Als teilnehmendes Unternehmen können Sie später das Ergebnis der Studie gesondert zugesendet bekommen und haben So die Gelegenheit von den Erfahrungen anderer zu profitieren. Darüber hinaus verlosen wir unter den Teilnehmern einige kleinere Gewinne.

Die 24 Fragen umfassende Umfrage wird zur Wahrung der Anonymität der Teilnehmer als Internet-Umfrage unter nachfolgender Adresse durchgeführt.

http://www.stud.Fernuni-Hagen.de/q5254752/Umfrage.html

Bitte geben Sie diese Einladung innerhalb Ihres Hauses an einen IT-Experten für Java-Frameworks bzw. entsprechenden IT-Entscheider weiter.

Als Teilnehmer starten Sie die Umfrage einfach durch Aufruf der obigen Internet-Adresse.

Weitere Fragen beantworte ich Ihnen gerne per email: Ute.Stegmann@Fernuni-Hagen.de

Ich hoffe ich habe nun Ihr Interesse geweckt und bedanke mich von dieser Stelle für Ihre Unterstützung.

Mit freundlichen Grüßen

[Ute Stegmann persönlich unterschrieben]

(c)2005 Ute Stegmann DV-Kaufmann, Informatik-Studentin an der FernUniversität in Hagen
Fachbereich Informatik - Lehrgebiet Programmiersysteme - Prof. Dr. Friedrich Steimann

5 Umfrageergebnisse

5.1 Teil 1 Architekturen für verteilte Anwendungen

5.1.1 Motivation der gestellten Architekturfragen

In Frage 1 soll zunächst abgeklärt werden, welche Architekturen das jeweilige Unternehmen einsetzt bzw. bei Software- und Beratungshäusern deren jeweilige Kunden einsetzen.

Hintergrund hierfür ist, zu klären welche Architekturen die Frameworks unterstützen oder integrieren müssen. Es ist davon auszugehen, dass die Architekturentscheidung und die Entscheidung für bzw. gegen bestimmte Frameworks sich gegenseitig stark beeinflussen.

In Frage 2 geht es darum, die Relevanz von zu integrierenden sonstigen Systemen wie z.B. Großrechner zu erfragen. Im später folgenden Teil der Fragen zu Java- und Frameworktechnologien wird dann noch genauer auf die Integration von z.B. Host-Anwendungen (Legacy-Systeme) eingegangen.

Mit Frage 3 soll untersucht werden, inwieweit fertige Komponenten in den Unternehmen heute eine Rolle spielen. Denn eine Motivation des EJB-Konzeptes war, dass Interoperabilität zwischen EBJ-Komponenten verschiedener Softwarehersteller –vorbehaltlich sauberer Schnittstellen- möglich ist, da alle dem gleichen Modell folgen.

5.1.2 Fragen und Ergebnisse zu Teil 1 im Detail

Frage Nr. 1:
Welche Form von Architekturen setzen Sie in Ihrem Unternehmen aktuell für Verteilte Systeme ein bzw. planen Sie in naher Zukunft einzusetzen?

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
2-Tier Client/Server (Fat-Client, Datenhaltung)	13	37%
3-Tier Client/Server (Präsentation, Anwendungslogik, Datenhaltung)	23	66%
4-Tier Client/Server (Präsentation, Geschäftslogik, Geschäftsobjekte, Datenhaltung)	23	66%
5-Tier Client/Server (Client, Web, Geschäftslogik, Geschäftsobjekte, Datenhaltung)	13	37%
Andere:	1	3%
Gültige Antworten insgesamt	35	100%

Eingaben im ergänzenden Feld (Andere etc.):

- 2,3,4 je nach Anforderungen, Skills der Projektmitarbeiter, Komplexität der Aufgabe,...

Fragen und Ergebnisse zu Teil 1 im Detail – Fortsetzung-

Frage Nr. 2:
Welche Systeme sind in Ihrem Unternehmen neben Verteilten Systemen noch vorhanden?

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
Host-Systeme (z.B. IBM-Großrechner)	17	49%
Einzelplatzsysteme (z.B. in externen Geschäftsstellen)	25	71%
mobile Systeme (z.B Notebooks oder PDA)	25	71%
Andere:	3	9%
Gültige Antworten insgesamt	35	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Wir sind IT-Dienstleister und führen im Auftrag unserer Kunden Individualentwicklungen durch, d.h. die Systemlandschaft bei uns im Hause ist nicht die Zielumgebung. Unser Schwerpunkt liegt aber in der Entwicklung vert. Systeme
- Serversysteme für J2EE-Appliaktionen
- IBM iSeries

Frage Nr. 3:
Werden in Ihrem Unternehmen komponentenbasierte Systeme entwickelt?

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
nein	1	3%
auf Basis von eingekauften Komponenten	0	0%
auf Basis von Open-Source Komponenten	14	40%
nur für einige Teilsysteme	4	11%
bei allen Systemen, die neu entwickelt werden	10	29%
bei manchen Systemen komplett	6	17%
Gültige Antworten insgesamt	35	100%

5.2 Teil 2 Strategien zur Frameworkentwicklung

5.2.1 Motivation der gestellten Strategiefragen zur Frameworkentwicklung

In diesem Teil des Fragenkomplexes geht es darum zu ermitteln, welche Faktoren welchen Einfluss auf die Wahl der Architektur und den Frameworkeinsatz haben.

Bei den Einflussfaktoren zum Frameworkeinsatz wird unterschieden zwischen der generellen Entscheidung für einen Frameworkeinsatz und der späteren Entscheidung für bestimmte Frameworks oder eine Eigenentwicklung.

Ziel dieses Fragenkomplexes ist es, allgemein herauszufinden, welche Kriterien bei der Neuentwicklung bzw. den Einsatz von Frameworks in Unternehmen am wichtigsten sind.

5.2.2 Fragen und Ergebnisse zu Teil 2 im Detail

Frage Nr. 4:
Bitte bewerten Sie die Wichtigkeit folgender Einflussfaktoren für die Wahl der Architektur von Verteilten Systemen.

Antwort (nur eine Antwort war möglich)	1	%	2	%	3	%	4	%	5	%
Bestehende IT-Strategien in Ihrem Unternehmen	13	37%	11	31%	8	23%	2	6%	1	3%
Anzahl Benutzer (Nebenläufigkeit)	8	23%	14	40%	9	26%	3	9%	1	3%
Technische Aspekte der Netzinfrastruktur (z.B. Bandbreite)	4	12%	14	41%	12	35%	3	9%	1	3%
bestehende heterogene Hardware	2	6%	9	27%	12	36%	6	18%	4	12%
Integration bestehender Infrastruktur (z.B. Anwendungs- oder DB-Server)	17	49%	11	31%	6	17%	1	3%	0	0%
Leistungsfähigkeit der Rechnersysteme	3	9%	9	26%	16	46%	4	11%	3	9%
Skalierbarkeit	16	46%	11	31%	7	20%	1	3%	0	0%
Sicherheit / Zuverlässigkeit	17	49%	14	40%	4	11%	0	0%	0	0%
Organisation der Datensicherung	8	24%	4	12%	8	24%	9	27%	4	12%
Zentrale Pflege der Anwendungen bei Thin Clients	7	21%	10	29%	10	29%	3	9%	4	12%

Fragen und Ergebnisse zu Teil 2 im Detail -Fortsetzung-

Frage Nr. 5:
Bewerten Sie nun die Wichtigkeit, die folgende Kriterien bei der generellen Entscheidung für den Einsatz von Frameworks in Ihrem Unternehmen haben.

Antwort (nur eine Antwort war möglich)	1	%	2	%	3	%	4	%	5	%
Wirtschaftlichkeit (z.B Entwicklungsaufwand, Kosten der Anwendung, Aufwand für Einarbeitung)	18	53%	15	44%	1	3%	0	0%	0	0%
Wartbarkeit bezüglich Fehlerbereinigung, Änderung, Erweiterung, Modularität und Test	17	50%	16	47%	1	3%	0	0%	0	0%
Wiederverwendbarkeit	11	32%	13	38%	9	26%	1	3%	0	0%
Entwicklungsrisiko	5	15%	19	56%	9	26%	1	3%	0	0%
Zukunftssicherheit	12	35%	14	41%	6	18%	2	6%	0	0%
Qualität, Sicherheit (z.B. bekannte Stabilität des Frameworkcodes)	11	32%	19	56%	3	9%	1	3%	0	0%
Standardisierung des Softwareentwicklungsprozesses	11	32%	15	44%	5	15%	3	9%	0	0%
Möglichkeit des Einsatzes von Codegeneratoren	2	6%	10	30%	8	24%	7	21%	6	18%

Frage Nr. 6:
Bewerten Sie die Wichtigkeit, die folgende Kriterien bei der Entscheidung für bestimmte Frameworks oder ggf. auch für eine Eigenentwicklung haben.

Antwort (nur eine Antwort war möglich)	1	%	2	%	3	%	4	%	5	%
Portabilität der eigenen Entwicklung	5	15%	17	50%	10	29%	0	0%	2	6%
Interoperabilität mit anderen Produkten	10	29%	20	59%	3	9%	0	0%	1	3%
Zukunftssicherheit bezüglich technischer Entwicklungen	11	32%	16	47%	5	15%	2	6%	0	0%
Zukunftssicherheit bezüglich Anwenderanforderungen	10	29%	19	56%	5	15%	0	0%	0	0%
bekannte Skalierbarkeit des Rahmencodes	6	18%	13	38%	13	38%	2	6%	0	0%
betriebliche Randbedingungen	4	13%	10	32%	12	39%	5	16%	0	0%
Anforderungen der Anwender-Fachbereiche, Funktionalität / Komponentenvielfalt	7	22%	13	41%	11	34%	1	3%	0	0%
Integration von sogenannten Legacy-Systemen, d.h. bestehende Alt-Systeme (z.B. Host-Anwendungen)	7	21%	10	30%	9	27%	6	18%	1	3%
Akzeptanz durch die Softwareentwickler inkl. Handhabbarkeit, Zuverlässigkeit, Erlernbarkeit und damit verbundenem Schulungsaufwand	9	26%	19	56%	4	12%	1	3%	1	3%

5.3 Teil 3 Framework- und Java-Technologien

5.3.1 Motivation der gestellten Fragen zu den Framework- und Java-Technologien

Im Teil 3 der Umfrage geht es nun um den konkreten Einsatz von Frameworks in den befragten Unternehmen bzw. bei Softwarehäusern deren Kunden.

Dieser Fragenkomplex deckt 3 Bereiche ab.

- Zunächst wird gefragt, welche Art(en) von Java-Frameworks für verteilte Systeme eingesetzt werden. Dabei wird auch nach Eigenentwicklung und Open Source-Einsatz gefragt.
- Ein zweiter Bereich sind Fragen zur konkreten Leistungsfähigkeit der eingesetzten Frameworks. Kriterien sind hierbei unter anderem die Standardisierung des Entwicklungsprozesses bis hin zum Einsatz von Codegeneratoren für die einzelnen Schichten (Präsentation, Anwendungslogik, Persistenz). Ebenfalls wird nach Testverfahren und Möglichkeiten des Fehlersupports gefragt. Zwei weitere wichtige Punkte sind die Migration aus Vorgängersystemen und das Bereitstellen von Prototypen.
- Schließlich wird den konkret zum Einsatz kommenden Java-Technologien für die einzelnen Schichten der Anwendungsarchitektur sowie dem Bereich Test ein weiterer Fragenkomplex gewidmet. Hier geht es vor allem darum, herauszufinden welche Entscheidungen gemäß den Technologie-Fragestellungen (siehe Kapitel 2) getroffen wurden.

5.3.2 Fragen und Ergebnisse zu Teil 3 im Detail

Frage Nr. 7:
Welche Frameworks setzen Sie bereits zur Softwareentwicklung von Verteilten Anwendungen ein?

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
selbst entwickelte Frameworks	27	79%
Open-Source Frameworks	31	91%
folgende kommerzielle Frameworks (gekauft):	5	15%
Gültige Antworten insgesamt	34	100%

Eingaben im ergänzenden Feld (Andere etc.):

- .NET
- cocoon, struts
- JCoffee
- Common-Controls
- Toplink
- abaxx

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 8:
Welche Teile der Entwicklung werden durch bei Ihnen eingesetzte Frameworks (Rahmenwerke, Programmiergerüste) standardisiert?

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
Oberflächenlayout der Anwendung (GUI)	28	82%
Anwendungslogik	20	59%
Datenzugriffslogik bzw. Persistenzschicht	32	94%
Testverfahren	25	74%
Weitere (bitte aufzählen):	3	9%
Gültige Antworten insgesamt	34	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Autentifizierung, Loggin
- Interprozesskommunikation
- Komponenteninteraktion, Transaktionsmanagement, Metrikanalyse, Logging, Konfiguration, Exceptionhandling
- DV-Basisfunktionalitäten, Einhaltung der Codier- und Modellierrichtlinien

Frage Nr. 9:
Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Allgemein:

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
Java Standard Edition (J2SE)	16	48%
Java Enterprise Edition (J2EE)	30	91%
Open-Source-Frameworks	24	73%
Andere:	2	6%
Gültige Antworten insgesamt	33	100%

Eingaben im ergänzenden Feld (Andere etc.):

- JDO
- Struts, Spring, OpenJades

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 10:
Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Für die Präsentationsschicht:

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
Struts (Open-Source-Framework)	20	59%
ein anderes Web-Framework (ggf. unter "Andere:" genauer bezeichnen)	8	24%
eigenes Web-Frontend (JSP,Servlets,..) mit Zugriff auf EJB Session Beans	14	41%
Eclipse SWT	14	41%
Java-Swing (Thin-Client in Verbindung mit Anwendungslogik auf Basis von EJB)	12	35%
Java-Swing (Fat-Client)	11	32%
Andere:	10	29%
Gültige Antworten insgesamt	34	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Cocoon
- Java Server Faces
- JSF, teilweise Eigenentwicklung
- JSF u.a.
- Velocity
- SpringMVC
- Spring
- Cameleon OSP
- Spring MVC

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 11:
Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Für die Anwendungslogikschicht (Business-Tier):

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
EJB Session Beans, Message-Driven Beans	22	65%
eigene Logik / eigene Frameworks	16	47%
Spring (Open-Source Framework)	15	44%
Eclipse Framework im Rahmen der Rich Client Plattform (RCP)	9	26%
Andere:	4	12%
Gültige Antworten insgesamt	34	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Cocoon
- Hibernate
- Hivemind
- Hivemind

Frage Nr. 12:
Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Für die Persistenzschicht (Datenhaltung):

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
	0	0%
EJB: Container Managed Entity Beans (CMP)	11	32%
EJB: Bean Managed Entity Beans (BMP)	3	9%
Java Data Objects (JDO)	10	29%
Eigenlösung für die Persistenz	12	35%
Hibernate (Open-Source Framework)	16	47%
Gültige Antworten insgesamt	34	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Toplink (im Oracle-Umfeld)
- IBM Lotus Domino
- Oracle Toplink
- EJB3 Persistence
- Toplink (kommerziell)
- Cameleon OSP / DataObject
- Spring JDBC

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 13:
Verwenden Sie für bestimmte Teile Ihrer Softwareentwicklung sogenannte Code-Generatoren, mit denen Programmcode erzeugt werden kann?

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
nein	14	44%
ja, folgende Arten von Code-Generatoren:	18	56%
Gültige Antworten insgesamt	32	100%

Eingaben im ergänzenden Feld (Andere etc.):

- objectiveX
- Persistenzlayer, Together (fachliche Zugriffsschichten)
- Lomboz (Eclipse-Plugin) f.d. Generierung von Interfaces+Deployment-Descriptoren im J2EE-Umfeld
- JET Templates, eigene Generatoren, XDoclet
- AndroMDA, openArchitectureWare
- Component Generator (Persistenz, Finder, EBs, Deployment Desc., DB-Scripte), View-Description Generator
- XDoclet, IBM WSAD
- AndroMDA OpenArchitectureWare
- eigener in Verbindung mit AndroMDA & XDoclet
- Ruby on Rails
- GUI aus DB-Tabellen generieren
- Eingabevalidierung, Zugriff auf Datenbanken / Legacy Systeme
- xdoclet
- Eigenenteicklungen XML Spec --> Java,...
- Persistenz aus DB-Schema

Frage Nr. 14:
Basieren Ihre auf Basis von Frameworks erstellten Verteilten Anwendungen auf offenen Schnittstellen und Protokollen und sind damit erweiterbar?

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
ja	30	94%
nein	2	6%
Gültige Antworten insgesamt	32	100%

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 15 a:
In welcher Form unterstützen Ihre Frameworks ggf. die Migration aus Vorgängersystemen?
Migration der Programmquellen:

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
Keine Unterstützung, die Programme müssen komplett neu erstellt werden.	17	55%
Einige Teile der Programmquellen können automatisch migriert werden, der Rest wird neu erstellt.	10	32%
Große Teile der Programmquellen können automatisch migriert werden.	3	10%
Sonstige Lösung:	0	0%
Gültige Antworten insgesamt	31	100%

Eingaben im ergänzenden Feld (Andere etc.):

Frage Nr. 15 b:
Migration der Daten:

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
Keine Migration, die Daten müssen durch Mitarbeiter neu erfasst werden.	0	0%
Datenmigration aus den Datenbankquellen des Vorgändersystems in die des neuen Systems wird unterstützt (z.B. mit Zuweisungstabellen).	24	77%
Sonstige Lösung:	3	10%
Gültige Antworten insgesamt	31	100%

Eingaben im ergänzenden Feld (Andere etc.):

- i.d.R. direktes Aufsetzen auf bestehenden Daten
- Keine Migration, die Daten werden benutzt wie sie sind
- Keine Migration, die Daten

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 16:
Unterstützen in Ihrem Hause eingesetzte Frameworks das Erstellen von Demoversionen bzw. Prototypen?

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
nein	19	66%
ja, auf folgende Art:	10	34%
Gültige Antworten insgesamt	29	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Anbieten einer Offlineversion mit Demodaten
- Webbasiertes Onlinedemo
- GUI-Komponentenframework für GUI-Prototypen und Usability-Storyboards
- Über vordefinierte Deployment-Prozesse können anwendungsspez. Systemausprägungen auf Basis der bestehenden Komponenten (u.a. auch mit Demodaten) generiert werden
- Schnelle Entwicklung von Dummy-Oberflächen mit Hilfe von Struts/Common-Controls
- durch Codegenerator große Zeitersparnis, dadurch schnell eine Prototyp-Version erstellbar
- Begrenzung der Datensätze u.a.

Frage Nr. 17:
Welche Art von Testverfahren führen Sie durch?

Antwort (mehrere Antworten waren möglich)	Häufigkeit	Anteil
Entwickeln mit Unit Tests (White Box Tests)	29	85%
Black Box Tests	16	47%
Integrationstests	28	82%
Lasttests	25	74%
Performance Tests	29	85%
Usability Tests	24	71%
regelmäßige Codereviews / Code Walkthroughs	26	76%
Gültige Antworten insgesamt	34	100%

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 18 a:
Unterstützen Ihre Frameworks Sie bei den von Ihnen durchgeführten Testverfahren bzw. setzen Sie ein Test-Framework wie z.B. JUnit für die Erzeugung von Testklassen ein?

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
nein	10	30%
ja, auf folgende Art:	23	70%
Gültige Antworten insgesamt	33	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Continous Integration auf einem Extra Rechner
- Einsatz von JUnit für automatisierbare Modultests
- JUnit
- JUnit-Tests zum Testen von Komponenten-Schnittstellen und -Interna
- für Unit Tests
- Integrationin Eclipse, Anpassung von Generatoren für Use-Cases (Eigenbau)
- JUnit, StrutsTestCase, MockObjects
- JUnit
- Timer für Programmabschnitte
- JUnit
- Wie empfohlen
- JUnit mit JCoverage eingebettet im Buildprozess (Maven)

Frage Nr. 18 b:
Welche besonderen Probleme traten beim Test der auf Basis von Frameworks entwickelten Applikation(en) auf?

Antwort	Häufigkeit	Anteil
Gültige Antworten insgesamt	33	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Teilweise erschwertes "Erstes Aufsetzen"
- Gute test tools sind zu teuer ;-)
- Keine
- Bereitstellung hinreichender nicht trivialer Testdaten
- Eine der primären Herausforderungen bestand darin, den besonderen Testanforderungen bestimmter Industriezweige zu genügen. Hier sind vor allem die Anforderungen der pharmazeutischen Industrie im Hinblick auf den Entwicklungsprozess
- - Abhängikeiten zwischen Komponeniten sind oft zu hart im Code verdrahtet
  - Anwendung wurde nicht auf Testbarkeit hin entwickelt
- Entwicklergestützte Tests, wie JUnit, haben das Problem, dass sie in stressigen Projektphasen nicht mehr konsequent durchgeführt werden. Metricen wie JCoverage decken dies zwar auf, lösen das Problem aber nicht.
- Die Qualität der Testdaten ist schlecht.
- Probleme mit inkonsistenten Testdaten

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 19:
Unterstützen Ihre Frameworks die Verwaltung von Fehlermeldungen, Support-Anfragen, etc. von Kunden bzw. Anwendern?

Antwort (nur eine Antwort war möglich)	Häufigkeit	Anteil
nein	17	55%
ja, durch:	14	45%
Gültige Antworten insgesamt	31	100%

Eingaben im ergänzenden Feld (Andere etc.):

- Nutzung von Bugzilla zur Verwaltung von Fehlermeldungen, würde ich aber nicht als Framework bezeichnen.
- integriertes Sammeln und Weiterleiten von Prozessprotokollen und Fehlerzuständen
- Exceptions werden kategorisiert und an ein Trackingsystem übergeben.
- Erfassung von Sitzungszuständen und Logfiles zum Einfügen in Bugzilla
- ein entsprechendes GUI und die Anschlussmöglichkeit eines PVCS (Tracker, Tracer etc.)
- Email
- Eigenentwicklung
- Protokollierung, Kategorisierung, Tracking
- zentrale Fehlerbehandlung, Logging, ggf. Tracing, etc
- Issue und Bugtracking

Frage Nr. 20:
Bitte beschreiben Sie in einigen kurzen Sätzen die typischen Merkmale und Besonderheiten Ihres Frameworks soweit Sie auf den vorherigen Seiten noch nicht dazu befragt wurden.

Eingaben :

- Da wir ein IT-Beratungshaus sind und daher unterschiedliche Projekte bearbeiten, ist auch der Framework-Einsatz abhängig vom konkreten Projekt. Im Hause gibt es ein selbstentwickeltes Framework (vergleichbar mit J2EE + Application-Server), bei neuen Proje
- das Framework erlaubt vo allen die Entwicklung von direkt synchronisierten und verteilten Modulen durch ein verteiltes Event-Handling (Messaging) zwischen allen Komponenten (z.B. auch von Server zum Client / Push-Prinzip)
- Stark standardisierte Anwendungsarchitektur (Objekt-Aktion Paradigma). Es gibt nur 3 Fenstertypen. Flexible Benutzerführung. Eigenes Protokoll für Kommunikation zwischen Client und Server (View-Agent). Gemeinsame Datenquellen (IMS, DB2)von Legacy (Hos
- sehr einfache API auf Basis von POJO Classes Darauf aufbauend musterimplementierungen mit J2EE und IBM Lotus Domino Wir auchten bei unserem Framework in jedem Fal afu ein einfaches Klassen Design und vermeiden unötig aufgeblähte Klassenhirarchien
- Wir haben eigene Frameworks für XML-orientierte Persistenzschicht und für GUI-Komponenten im Einsatz, planen aber ggf. auf JSF oder ähnliche umzusteigen.

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 20 –Fortsetzung-

- Das Komponentenframework JCoffee® liefert eine Möglichkeit, wie der Konflikt zwischen "J2EE ist eine leistungsfähige und zukunftorientierte Plattformtechnologie" und "Die technische Komplexität von J2EE beeinträchtigt häufig die Produktivität und Qualität
- Mehrsprachigkeit ist ein wichtiger Punkt, der ein Grund für die Einführung von Struts war.
- Wir arbeiten mit einheitlichen Architketuren und einheitlichen Technologiestacks wie z.B. Struts / Hivemind / Hibernate / JBoss ... Dadurch ist es möglich neben Architekturvorgaben auch Programmierrichtlinien und Referenzimplementierungen zur Verfügung zu
- Haupt-Framework OpenJades: Business Layer für EclipseRCP, basierend auf J2EE, CMP EJBs, Codegenerator AndroMDA/XDoclet, s. www.openjades.com
- Der Anwendungsentwickler muss sich mit den zum Einsatz kommenden Technologien nicht auskennen. Anwendungen können durch reine Framework-Änderungen auf eine andere Technologie portiert werden.
- Standardisierte Oberflächen. Eigenes Protokoll zur Kommunikation Client/Server (view/agent). Eigener Persistenzdienst mit Datenhaltung in DB2 und IMS.
- Unser sogenanntes I*NET-Framework erzwingt ein einheitliches Programmiermodell auf Basis der MVC-Architektur. Alle technischen Schnittstellen (z.B. Security, Backend-Zugriff und sonstiges Services) werden durch das Framework gekapselt, damit die eigentlic
- CUBA (Component Unification Base) ist die ultimative Plattform für die Entwicklung komponentenbasierter Java-Anwendungen, die flexibel als EJB-Applikationen, AXIS WebServices oder stand-alone J2SE-Programme eingesetzt werden können.

Frage Nr. 21:
Wie aufwendig war ggf. Ihre Frameworkentwicklung oder Frameworkevaluation?
Antworten:

Dauer in Monaten : 5
Anzahl Entwickler : 4

Dauer in Monaten : 1
Anzahl Entwickler : 2
Evaluierte Framew: Struts, Spring, Cocoon, VisualStudio, Eclipse

Dauer in Monaten : Evaluierung: 0,5 bis 1 Monat
Anzahl Entwickler : 1
Evaluierte Framew: intelliBO, Castor, Struts, Jasper Reports

Dauer in Monaten : parallel zur Projektrealisierung, laufende Weiterentwicklung
Anzahl Entwickler : 8

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 20 –Fortsetzung-

Dauer in Monaten : 6
Anzahl Entwickler : 8

Dauer in Monaten : 60
Anzahl Entwickler : 12
Evaluierte Framew: keine

Dauer in Monaten : 12
Anzahl Entwickler : 1

Dauer in Monaten : gering - "on the job" und nicht explizit als Evaluation
Anzahl Entwickler : 3-12

Dauer in Monaten : 3
Anzahl Entwickler : 2
Evaluierte Framew: Struts, JSF, Hibernate, Spring, Cocoon, J2EE, Prevailer

---

Dauer in Monaten : 6
Anzahl Entwickler : 3
Evaluierte Framew: J2EE, EJBs, Spring, Hivemind, Struts, Velocity

---

Dauer in Monaten : 120

Dauer in Monaten : 12
Anzahl Entwickler : 1
Evaluierte Framew: Struts, Cocoon, EJB, Hibernate, JPOX, Spring, Jakarta Commons, JSF, JDO, iBatis ...

Dauer in Monaten : 48
Anzahl Entwickler : 50

Dauer in Monaten : 12
Anzahl Entwickler : 6
Evaluierte Framew: Struts, Hivemind, Spring, Tapestry, Maverick, Hibernate, Toplink

---

Dauer in Monaten : 1
Anzahl Entwickler : 3
Evaluierte Framew: Struts, Spring, OpenJades (Eigenentwicklung)

Dauer in Monaten : 2
Anzahl Entwickler : 20

Fragen und Ergebnisse zu Teil 3 im Detail -Fortsetzung-

Frage Nr. 20 –Fortsetzung-

Dauer in Monaten : 1/2 Monat Evaluation
Anzahl Entwickler : 1
Evaluierte Framew: Cameleon OSP, IBM-Komponenten für Java-Rich-Client

---

Dauer in Monaten : 1
Anzahl Entwickler : 2
Evaluierte Framew: struts, tapestry, turbine

Dauer in Monaten : 48
Anzahl Entwickler : 20
Evaluierte Framew: Zu Beginn der Arbeiten gab es keine

Dauer in Monaten : 3
Anzahl Entwickler : 5
Evaluierte Framew: spring

Dauer in Monaten : 60
Anzahl Entwickler : 8
Evaluierte Framew: I*NET

Dauer in Monaten : *keine Angaben*
Anzahl Entwickler : *keine Angaben*
Evaluierte Framew: Struts, Hibernate, Spring, Maven, Luntbuild, JUnit, DBUnit, Velocity, Jakarta Commons

---

5.4 Teil 4 Fragen zur Modellierungspraxis

5.4.1 Motivation der gestellten Fragen zur Modellierung

In diesem Fragenkomplex wird der im Zusammenhang mit dem Entwicklungsprozess wichtigen Frage nach dem Umfang der Objektorientierten Modellierung nachgegangen. Eine umfassende Analyse dieses Bereiches hätte den Rahmen der Umfrage gesprengt. Allerdings sollte gerade auf Grund der Diskussion um die MDA (siehe Kapitel 2) nicht ganz auf diesen Fragenkomplex verzichtet werden. Ziel dieser Frage ist, herauszufinden inwieweit eine überwiegend von der Modellierung getriebene / gesteuerte Entwicklung in der Realität vorkommt.

5.4.2 Fragen und Ergebnisse zu Teil 4 im Detail

Frage Nr. 22:
In welchem Umfang betreiben Sie die objektorientierte Modellierung mit UML?

Antwort (nur eine Antwort war möglich)	Häufigkeit	%
1. Gar nicht bzw. es existieren keine Vorgaben, jeder Softwareentwickler kann, muss aber nicht UML nutzen.	5	16%
2. In der Analysephase werden auf Basis der Anwendungsfalldiagramme (Use Cases) oder anderer Beschreibungen Klassendiagramme erstellt. Weitere Modellierung wird nicht betrieben.	7	22%
3.(umfasst 2.) Es werden im weiteren Verlauf durch Grob- und Feinkonzept die Klassendiagramme verfeinert und ggf. Interaktionsdiagramme zur genaueren Beschreibung erstellt. In der Realisierungs- bzw. reinen Entwicklungsphase wird nicht modelliert.	10	31%
4.(umfasst 3.) In allen Phasen der Softwareentwicklung werden UML-Diagramme erstellt, insbesondere jedes Programm im Vorwege modelliert. Die Programme können allerdings später vom Modell abweichen.	6	19%
5.(umfasst 4.) Es wird eine komplett modellgetriebene Softwareentwicklung betrieben, d.h. es existieren insbesondere auch zu allen Programmen ggf. durch Reverse-Engineering aktuelle Modelle/Diagramme.	4	13%
Gültige Antworten insgesamt	32	100%

Eingaben im ergänzenden Feld (Andere etc.):

- In der Designphase werden Neu- und Änderungskonzepte zunächst modelliert, das wird bis zum Ende der Entwicklung Iterativ wiederholt. Das bedeutet jedoch nicht, dass jedes Programmmodul modeliiert wird
- Orientierung am MOF standard,nicht nur UML
- Modellierung der Konzepte und Anforderungen, Design und Dokumentation auf der Basis des Sourcecodes (Reverse-Engineering) - kein MDA
- Vom Projekt abhängig
- Für Kundenprojekte verwenden wir den RUP
- Die Wahrheit liegt ZWISCHEN 3 und 4, da in 4 "JEDES" verwendet wird
- UML hat sich beim Kunden noch nicht durchgesetzt.
- Einsatz überwiegend zur Architekturbeschreibung, Klassen, Komponenten, Paket und Deployment-Diagramme
- Codegenerator für OpenJades geht vom UML Diagramm aus (z.B. erstellt in Gentleware Poseidon)
- stark von den Kundenvorgaben abhängig

5.5 Teil 5 Framework-Bewertung

5.5.1 Motivation der Bewertungsfragen zum Frameworkeinsatz

Im abschließenden Teil der Umfrage soll nun eine Bewertung des Frameworkeinsatzes zu den Oberbegriffen Stabilität und Sicherheit, Verteilungs- und Netzwerkaspekte, Benutzerakzeptanz und Wirtschaftlichkeit vorgenommen werden.
Hier schließt sich also der Kreis hin zu den am Anfang genannten Prioritäten der Einflussfaktoren, über die konkrete Umsetzung hin zum Erreichen der Ziele.

5.5.2 Fragen und Ergebnisse zu Teil 5 im Detail

Frage Nr. 23:
Bitte beantworten Sie bezogen auf den bisherigen Einsatz von Frameworks in Ihrem Unternehmen folgende Fragen:

Antwort (nur eine Antwort war möglich)	ja	%	nein	%
Stabilität und Sicherheit: Hat sich die Stabilität und Sicherheit in den durch Ihre Frameworks abgedeckten Bereichen verbessert?	28	93%	2	7%
Funktionieren die eingesetzten Frameworks fehlerfrei?	17	57%	13	43%
Gibt es Interoperabilitätsprobleme mit anderen Produkten (Protokolle, Datenformate)?	12	38%	20	63%
Verteilungs- / Netzwerkaspekte: Sind die mit Ihren Frameworks erstellten Applikationen im erforderlichen Rahmen skalierbar?	31	100%	0	0%
War eine Lastverteilung für Sie relevant und wird sie durch den Framework-Einsatz / die Verteilungsplattform unterstützt?	19	61%	12	39%
Benutzerakzeptanz: Haben ihre Anwender Nachteile wahrgenommen etwa durch Umstellung auf Internetanwendungen?	10	36%	18	64%
Wirtschaftlichkeit: Haben sich ihre Erwartungen bezüglich Kostenersparnis durch den Framework-Einsatz bisher erfüllt?	24	86%	4	14%
Hat der Frameworkeinsatz eine Wiederverwendung von Programmen, Komponenten etc. bewirkt? .	26	84%	5	16%

5.6 Teil 6 Unternehmensangaben

5.6.1 Motivation der Frage zum Unternehmen

Ergänzend zu den inhaltlichen Fragen werden hier Fragen zur Art und Größe des Unternehmens, der entwickelten Software und der Nutzung von Entwicklungswerkzeugen und Dienstprogrammen gestellt, um die Leistungsfähigkeit des Unternehmens einschätzen zu können.

5.6.2 Fragen und Ergebnisse zu Teil 6 im Detail

Frage Nr. 24:
Unternehmensdaten

Unternehmen mit ... Mitarbeiter:	
... bis zu 10 ... :	4
... bis zu 20 ... :	8
... bis zu 50 ... :	9
... bis zu 100 ... :	3
... bis zu 500 ... :	5
... über 500 ... :	5

Unternehmen mit ... Standorten:	
... einem ... :	18
... bis zu 10 ... :	13
... bis zu 50 ... :	0
... bis zu 100 ... :	1
... bis zu 500 ... :	1
... über 500 ... :	0

Häufigkeit verschiedener Arten entwickelter Software in den Unternehmen:	
Betriebswirtschaftliche SW :	27
Multimedia/Internet SW :	18
Technische SW :	12
Systemsoftware :	10
Software-Komponenten :	19
Embedded-Systems :	6
HW/SW Systemlösungen :	4
Sonstige Software :	3

Fragen und Ergebnisse zu Teil 6 im Detail -Fortsetzung-

Frage Nr. 24 -Fortsetzung-

Häufigkeit verwendeter Programmiersprachen :	
Java	33
C#	16
C++(C)	16
Cobol	6
PL/I	3
VB/Basic	9
Delphi/Pascal	5
Andere	2

Häufigkeit verwendeter Entwicklungsumgebungen:	
IBM Websphere	13
Borland JBuilder	8
IDEeclipse	27
Borland Together Edition for Eclipse	7
JDeveloper	3
Andere	5

Häufigkeit verwendeter UML-Modellierungstools:	
Rational Rose	13
Eclipse UML	8
Andere	13

Häufigkeit verwendeter Aplication Server :	
IBM Websphere	20
Bea Weblogic	17
Oracle	4
JBOSS	20
Andere	7
Micrsoft .NET	11

6 Summary

Neben den einzelnen Umfrageergebnissen, die ja für sich sprechen, lässt sich folgendes Fazit aus der Umfrage ziehen:

Bei den eingesetzten Framework-Technologien für die einzelnen Architekturschichten lassen sich keine eindeutigen Tendenzen für jeweils den Trend der Zukunft erkennen. Alle Technologien haben eine mehr oder weniger häufige Verbreitung, sieht man einmal von den ganze 3mal genannten Bean Managed Entity Beans ab.

Bezogen auf die verschiedenen Unternehmen lässt sich klar ausmachen, dass nur große Unternehmen wie große Versicherungen, Krankenkassen oder Banken sich eine eigene Framework-Entwicklung nur für das eigene Unternehmen leisten (können?). Interessant ist, dass bei großen Eigenentwicklungen immer auch Legacy-Systeme zu integrieren sind.

Dabei werden dann aber auch gleich Entwicklungszeiträume von 3-5 Jahren mit mehreren Entwicklern angesetzt. Ob sich dieser Aufwand lohnt, wenn man bedenkt wie rasant die Standards sich weiter entwickeln, bleibt fraglich.

Softwarehäuser setzen tendenziell – so ist zumindest ein Trend erkennbar – mehr auf eine Kombination aus Open-Source und J2EE-Standards und erweitern diese ggf. um ein eigenes Framework.

Was die Qualität der Frameworks angeht, so ist die Situation allerdings bedenklich. Immerhin 43% der Teilnehmer geben an, dass die Frameworks nicht fehlerfrei funktionieren. Und 36% geben an, dass die Anwender Nachteile wahrgenommen haben bei der Umstellung z.B. auf Internetanwendungen.

Sieht man sich die Bereiche Modellierung, Test und Fehlerverfolgung an, die doch recht wenig durchgängig umgesetzt werden, so scheint es insgesamt noch viel Verbesserungspotential bei der Softwareentwicklung mit Frameworks zu geben.

7 Literaturverzeichnis

[bea1] Bea WebLogic, http://www.bea.com/products/weblogic/server/index.shtml

[esoz1] Empirische Sozialforschung, Andreas Diekmann,Mai 2003, 10. Aufl.,ISBN:3499555514

[expo1] Expert One-on-One J2EE Development without EJB, Rod Johnson/Jürgen Höller, Juni 2004, ISBN:0764558315

[hibn1] Hibernate-Framework Homepage, http://www.hibernate.org/

[hibn2] Hibernate in Action,Christian Bauer/Gavin King, Oktober 2004, ISBN:193239415X

[ibm1] IBM Redbook Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework, http://http:/www.ibm.com/redbooks

[ibm2] IBM WebSphere, http://www-3.ibm.com/software/webservers/appserv/was/

[jain1] Java ist auch eine Insel..,Christian Ullenboom,September 2004, ISBN: 3898425266

[john1] R. E. Johnson: Components, Frameworks, Patterns, in Proceedings Sym-posium on Software Reusability, Boston 1997

[objectone1] objectone (JBoss, Hintergrundwissen zu J2EE,Frameworks), http://www.objectone.de/

[orac1] Oracle9i AS, http://www.oracle.com/ip/deploy/ias/index.html?oracle9ias_features.html

[Spring1] Spring Framework, http://springframework.org/

[struts] Struts Framework, http://jakarta.apache.org/struts

[sun1] Sun Enterprise Java Beans Technologie, http://java.sun.com/products/ejb/

[sun2] Java 2 Platform, Enterprise Edition (J2EE), http://java.sun.com/j2ee/

[SunJSF] J2EE JavaServer Faces, http://java.sun.com/j2ee/javaserverfaces/

[SunTutorial1] The J2EE 1.4 Tutorial, http://java.sun.com/j2ee/1.4/docs/tutorial/doc/

[vertsys1] Verteilte Systeme - Konzepte und Design, PEarson Studium, ISBN 3-8273-7022-1

[vertsys2] Verteilte Systeme und Anwendungen,Ulrike Hammerschall, Januar 2005,ISBN:3827370965

[wikipedia1] Wikipedia Die freie Enzyklopädie: http://de.wikipedia.org/wiki/

8 Abbildungsverzeichnis

J2EE-Technologie ……11
J2EE Client Tier ……11
J2EE Web Tier ……12
J2EE Server ……12
Modell-View-Controller (MVC) ..……...14
Spring Framework ……18
Spring Anwendungsszenarien - Typical full-fledged Spring web application ……..20
Spring Anwendungsszenarien - Spring middle-tier using a third-party web framework 20
Spring Anwendungsszenarien - Remoting usage scenario ……21
Spring Anwendungsszenarien - EJBs - Wrapping existing POJOs ……21
JavaServer Faces - Figure 1 High-level architecture of JSF ……..26
JavaServer Faces - Figure 2: The UI runs on the server ……26
Umfrage: Layout der Online-Umfrage ……...34
Tabellen zur Online-Umfrage ……35

FernUniversität in Hagen

Umfrage zum Einsatz von (Java-)Frameworks in Verteilten Systemen

Guten Tag!

Vielen Dank, dass Sie an unserer Umfrage teilnehmen, die im Rahmen einer studentischen Abschlussarbeit durchgeführt wird.

Auf den nachfolgenden Seiten werden wir Ihnen verschiedene Fragen zur Softwareentwicklung und speziell zum Einsatz von Frameworks (J2EE, Open-Source etc.) in Ihrem Unternehmen stellen.

Wichtiger Hinweise für IT-Beratungs- und Softwarehäuser:
Bitte beantworten Sie einige der nachfolgenden Fragen ggf. aus Sicht eines typischen Kunden Ihres Hauses.
Dies gilt, falls Sie (Normalfall) die von Ihnen entwickelten (oder beratenen) Frameworks/Softwarekomponenten für Verteilte Systeme in Ihrem eigenen Hause nicht umfassend einsetzen.

Als Teilnehmer an dieser Umfrage können Sie sich später das Ergebnis dieser Studie direkt zusenden lassen.
Darüber hinaus verlosen wir unter den Teilnehmern einige kleinere Gewinne.

Wir wünschen Ihnen nun viel Spaß und ein paar interessante Minuten bei der Beantwortung der Fragen.

Umfrage Starten

(c) 2005 Ute Stegmann DV-Kauffrau, Informatik-Studentin an der FernUniversität in Hagen
am Fachbereich Informatik - Lehrgebiet Programmiersysteme - Prof. Dr. Friedrich Steimann
Details (Lehrgebiet Programiersysteme)

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen**

Registrierung:

Die Umfrage richtet sich an IT-Experten (Java-Frameworks) und
IT-Entscheider aus Unternehmen mit entsprechender Kompetenz.
Um die Seriosität zu waren, bitten wir Sie sich vorab kurz hier zu registrieren.
Die Registrierung ist natürlich freiwillig.

Die Registrierungsdaten werden nicht zusammen mit Ihren Antworten gespeichert.
Sie brauchen also keinerlei Bedenken bezüglich der Annonymität Ihrer Eingaben haben.

Unternehmen: annonym
PLZ: 00000 Ort: Deutschland
Ihr Name: annonym
Ihre Tätigkeit: IT

Umfrage Starten

(c) 2005 Ute Stegmann DV-Kauffrau, Informatik-Studentin an der FernUniversität in Hagen
am Fachbereich Informatik - Lehrgebiet Programmiersysteme - Prof. Dr. Friedrich Steimann

Hinweise: Die Ausdrucke der Umfrageseiten wurden mit Microsoft Word erstellt. Das Layout kann von der Internet-Darstellung etwas abweichen.

- Teil 1: Architekturen für Verteilte Systeme -

Welche Form von Architekturen setzen Sie in Ihrem Unternehmen aktuell für Verteilte Systeme ein bzw. planen Sie in naher Zukunft einzusetzen?

- [ ] 2-Tier Client/Server (Fat-Client, Datenhaltung)
- [ ] 3-Tier Client/Server (Präsentation, Anwendungslogik, Datenhaltung)
- [ ] 4-Tier Client/Server (Präsentation, Geschäftslogik, Geschäftsobjekte, Datenhaltung)
- [ ] 5-Tier Client/Server (Client, Web, Geschäftslogik, Geschäftsobjekte, Datenhaltung)
- [ ] Andere:
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Bitte geben Sie ggf. mehrere Antworten, wenn mehrere Architekturen gleichberechtigt nebeneinander bestehen. Architekturen, die in der Vergangenheit eingesetzt wurden, aber aktuell oder in naher Zukunft nicht mehr verwendet werden, bitte nicht angeben.

- Teil 1: Architekturen für Verteilte Systeme -

Welche Systeme sind in Ihrem Unternehmen neben Verteilten Systemen noch vorhanden?

- [ ] Host-Systeme (z.B. IBM-Großrechner)
- [ ] Einzelplatzsysteme (z.B. in externen Geschäftsstellen)
- [ ] mobile Systeme (z.B Notebooks oder PDA)
- [ ] Andere:
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Insbesondere von Interesse sind Systeme, die im Rahmen eines Framework-Einsatzes für Verteilte Systeme zu unterstützen sind, z.B. auf denen noch Alt-Systeme weiter bestehen.

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen** **03/24**

- Teil 1: Architekturen für Verteilte Systeme -

Werden in Ihrem Unternehmen komponentenbasierte Systeme entwickelt?

- ☐ nein
- ☐ auf Basis von eingekauften Komponenten
- ☐ auf Basis von Open-Source Komponenten
- ☐ nur für einige Teilsysteme
- ☐ bei allen Systemen, die neu entwickelt werden
- ☐ bei manchen Systemen komplett
- ☐ Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen** **04/24**

- Teil 2: Strategien zur Frameworkentwicklung -

Bitte bewerten Sie die Wichtigkeit folgender Einflussfaktoren für die Wahl der Architektur von Verteilten Systemen.

Bewertung von 1 bis 5 sehr wichtig bis unwichtig:	sehr wichtig 1	2	3	4	unwichtig 5	weiß ich nicht
Bestehende IT-Strategien in Ihrem Unternehmen	☐	☐	☐	☐	☐	☐
Anzahl Benutzer (Nebenläufigkeit)	☐	☐	☐	☐	☐	☐
Technische Aspekte der Netzinfrastruktur (z.B. Bandbreite)	☐	☐	☐	☐	☐	☐
bestehende heterogene Hardware	☐	☐	☐	☐	☐	☐
Integration bestehender Infrastruktur (z.B. Anwendungs- oder DB-Server)	☐	☐	☐	☐	☐	☐
Leistungsfähigkeit der Rechnersysteme	☐	☐	☐	☐	☐	☐
Skalierbarkeit	☐	☐	☐	☐	☐	☐
Sicherheit / Zuverlässigkeit	☐	☐	☐	☐	☐	☐
Organisation der Datensicherung	☐	☐	☐	☐	☐	☐
Zentrale Pflege der Anwendungen bei Thin Clients	☐	☐	☐	☐	☐	☐

zur nächsten Frage.. | Eingabe löschen

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen** **05/24**

- Teil 2: Strategien zur Frameworkentwicklung -

Bewerten Sie nun die Wichtigkeit, die folgende Kriterien bei der generellen Entscheidung für den Einsatz von Frameworks in Ihrem Unternehmen haben.

Bewertung von 1 bis 5 sehr wichtig bis unwichtig:	sehr wichtig 1	2	3	4	unwichtig 5	weiß ich nicht
Wirtschaftlichkeit (z.B Entwicklungsaufwand, Kosten der Anwendung, Aufwand für Einarbeitung)	☐	☐	☐	☐	☐	☐
Wartbarkeit bezüglich Fehlerbereinigung, Änderung, Erweiterung, Modularität und Test	☐	☐	☐	☐	☐	☐
Wiederverwendbarkeit	☐	☐	☐	☐	☐	☐
Entwicklungsrisiko	☐	☐	☐	☐	☐	☐
Zukunftssicherheit	☐	☐	☐	☐	☐	☐
Qualität, Sicherheit (z.B. bekannte Stabilität des Frameworkcodes)	☐	☐	☐	☐	☐	☐
Standardisierung des Softwareentwicklungsprozesses	☐	☐	☐	☐	☐	☐
Möglichkeit des Einsatzes von Codegeneratoren	☐	☐	☐	☐	☐	☐

zur nächsten Frage.. | Eingabe löschen

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen** **06/24**

- Teil 2: Strategien zur Frameworkentwicklung -

Bewerten Sie die Wichtigkeit, die folgende Kriterien bei der Entscheidung für bestimmte Frameworks oder ggf. auch für eine Eigenentwicklung haben.

Bewertung von 1 bis 5 sehr wichtig bis unwichtig:	sehr wichtig 1	2	3	4	unwichtig 5	weiß ich nicht
Portabilität der eigenen Entwicklung	☐	☐	☐	☐	☐	☐
Interoperabilität mit anderen Produkten	☐	☐	☐	☐	☐	☐
Zukunftssicherheit bezüglich technischer Entwicklungen	☐	☐	☐	☐	☐	☐
Zukunftssicherheit bezüglich Anwenderanforderungen	☐	☐	☐	☐	☐	☐
bekannte Skalierbarkeit des Rahmencodes	☐	☐	☐	☐	☐	☐
betriebliche Randbedingungen	☐	☐	☐	☐	☐	☐
Anforderungen der Anwender-Fachbereiche, Funktionalität / Komponentenvielfalt	☐	☐	☐	☐	☐	☐
Integration von sogenannten Legacy-Systemen, d.h. bestehende Alt-Systeme (z.B. Host-Anwendungen)	☐	☐	☐	☐	☐	☐
Akzeptanz durch die Softwareentwickler inkl. Handhabbarkeit, Zuverlässigkeit, Erlernbarkeit und damit verbundenem Schulungsaufwand	☐	☐	☐	☐	☐	☐

zur nächsten Frage.. | Eingabe löschen

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen 07/24**

- Teil 3: Framework- und Java-Technologien -

Welche Frameworks setzen Sie bereits zur Softwareentwicklung von Verteilten Anwendungen ein?

- [ ] selbst entwickelte Frameworks
- [ ] Open-Source Frameworks
- [ ] folgende kommerzielle Frameworks (gekauft):
  - [ ]
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen 08/24**

- Teil 3: Framework- und Java-Technologien -

Welche Teile der Entwicklung werden durch bei Ihnen eingesetzte Frameworks (Rahmenwerke, Programmiergerüste) standardisiert?

- [ ] Oberflächenlayout der Anwendung (GUI)
- [ ] Anwendungslogik
- [ ] Datenzugriffslogik bzw. Persistenzschicht
- [ ] Testverfahren
- [ ] Weitere (bitte aufzählen):
  - [ ]
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

- Teil 3: Framework- und Java-Technologien -

Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Allgemein:

- [ ] Java Standard Edition (J2SE)
- [ ] Java Enterprise Edition (J2EE)
- [ ] Open-Source-Frameworks
- [ ] Andere:
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Bitte kreuzen Sie ggf. mehrere Antworten an.
Speziellere Fragen zur Präsentations-, Anwendungslogik- und Persistenz-Schicht folgen auf den nächsten Seiten.

- Teil 3: Framework- und Java-Technologien -

Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Für die Präsentationsschicht:

- [ ] Struts (Open-Source-Framework)
- [ ] ein anderes Web-Framework (ggf. unter "Andere:" genauer bezeichnen)
- [ ] eigenes Web-Frontend (JSP,Servlets,..) mit Zugriff auf EJB Session Beans
- [ ] Eclipse SWT
- [ ] Java-Swing (Thin-Client in Verbindung mit Anwendungslogik auf Basis von EJB)
- [ ] Java-Swing (Fat-Client)
- [ ] Andere:
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Bitte kreuzen Sie ggf. mehrere Antworten an.
Weitere Fragen zu Technologien/Frameworks für die Anwendungslogik- und Persistenz-Schicht folgen auf der nächsten Seiten.

- Teil 3: Framework- und Java-Technologien -

Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Für die Anwendungslogikschicht (Business-Tier):

- [ ] EJB Session Beans, Message-Driven Beans
- [ ] eigene Logik / eigene Frameworks
- [ ] Spring (Open-Source Framework)
- [ ] Eclipse Framework im Rahmen der Rich Client Plattform (RCP)
- [ ] Andere:
  - [ ]
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Bitte kreuzen Sie ggf. mehrere Antworten an. Eine weitere Frage zu Java-Technologien/-Frameworks für die Persistenzschicht folgt auf der nächsten Seite.

- Teil 3: Framework- und Java-Technologien -

Welche der folgenden Java-Technologien bzw. Arten von Java-Frameworks für die Entwicklung von Verteilten Anwendungen bevorzugen Sie?
Für die Persistenzschicht (Datenhaltung):

- [ ] EJB: Container Managed Entity Beans (CMP)
- [ ] EJB: Bean Managed Entity Beans (BMP)
- [ ] Java Data Objects (JDO)
- [ ] Eigenlösung für die Persistenz
- [ ] Hibernate (Open-Source Framework)
- [ ] Andere:
  - [ ]
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Bitte kreuzen Sie ggf. mehrere Antworten an.

- Teil 3: Framework- und Java-Technologien -

Verwenden Sie für bestimmte Teile Ihrer Softwareentwicklung sogenannte Code-Generatoren, mit denen Programmcode erzeugt werden kann?

- ☐ nein
- ☐ ja, folgende Arten von Code-Generatoren:
  [ ]
- ☐ Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Sie sollten hier mit ja antworten, wenn Sie mit Hilfe von Tools bzw. auf Basis von UML-Diagrammen oder aus einer Entwicklungsumgebung heraus bestimmte Programmteile oder auch ganze Programme automatisch erzeugen lassen. Dies kann die Nutzung von Container Managed Entity Beans ebenso sein wie das Erzeugen von Java-Programmen aus UML-Sequenzdiagrammen. Bitte nennen Sie uns in kurzen Stichworten, auf welche Arten Sie Programmcode generieren.

- Teil 3: Framework- und Java-Technologien -

Basieren Ihre auf Basis von Frameworks erstellten Verteilten Anwendungen auf offenen Schnittstellen und Protokollen und sind damit erweiterbar?

- ☐ ja
- ☐ nein
- ☐ Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

- Teil 3: Framework- und Java-Technologien -

In welcher Form unterstützen Ihre Frameworks ggf. die Migration aus Vorgängersystemen?

Migration der Programmquellen:

- ☐ Keine Unterstützung, die Programme müssen komplett neu erstellt werden.
- ☐ Einige Teile der Programmquellen können automatisch migriert werden, der Rest wird neu erstellt.
- ☐ Große Teile der Programmquellen können automatisch migriert werden.
- ☐ Sonstige Lösung:
  [ ]
- ☐ Frage kann ich nicht beantworten

Migration der Daten:

- ☐ Keine Migration, die Daten müssen durch Mitarbeiter neu erfasst werden.
- ☐ Datenmigration aus den Datenbankquellen des Vorgändersystems in die des neuen Systems wird unterstützt (z.B. mit Zuweisungstabellen).
- ☐ Sonstige Lösung:
  [ ]
- ☐ Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

- Teil 3: Framework- und Java-Technologien -

Unterstützen in Ihrem Hause eingesetzte Frameworks das Erstellen von Demoversionen bzw. Prototypen?

- ☐ nein
- ☐ ja, auf folgende Art:
  [ ]
- ☐ Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

- Teil 3: Framework- und Java-Technologien -

Welche Art von Testverfahren führen Sie durch?

- [ ] Entwickeln mit Unit Tests (White Box Tests)
- [ ] Black Box Tests
- [ ] Integrationstests
- [ ] Lasttests
- [ ] Performance Tests
- [ ] Usability Tests
- [ ] regelmäßige Codereviews / Code Walkthroughs
- [ ] Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

- Teil 3: Framework- und Java-Technologien -

Unterstützen Ihre Frameworks Sie bei den von Ihnen durchgeführten Testverfahren bzw. setzen Sie ein Test-Framework wie z.B. JUnit für die Erzeugung von Testklassen ein?

- ( ) nein
- ( ) ja, auf folgende Art:

  [ ]
- ( ) Frage kann ich nicht beantworten

Welche besonderen Probleme traten beim Test der auf Basis von Frameworks entwickelten Applikation(en) auf?

[ ]

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Bitte beschreiben Sie ggf. kurz die Probleme in dem Eingabetextfeld.

- Teil 3: Framework- und Java-Technologien -

Unterstützen Ihre Frameworks die Verwaltung von Fehlermeldungen, Support-Anfragen, etc. von Kunden bzw. Anwendern?

- ☐ nein
- ☐ ja, durch:
- ☐ Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Antworten Sie mit ja, wenn beispielsweise jede Fehlermeldung eine eindeutige Fehlernummer und Angaben zu den Fehlerquellen wie Programm, etc. enthält, auf die sich Anwender bzw. Kunden später beziehen können um schneller Hilfe zu erhalten. Auch Komponenten zur automatischen Protokollierung von Fehlern, automatischer Benachrichtigung von Administratoren bzw. Support-Mitarbeitern und weitergehende Tools oder komplexe Supportsysteme sind hier gemeint.

- Teil 3: Framework- und Java-Technologien -

Bitte beschreiben Sie in einigen kurzen Sätzen die typischen Merkmale und Besonderheiten Ihres Frameworks soweit Sie in den vorherigen Fragen noch nicht dazu befragt wurden.

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:

- Teil 3: Framework- und Java-Technologien -

Wie aufwendig war ggf. Ihre Frameworkentwicklung oder Frameworkevaluation?

Dauer in Monaten:

Durchschnittliche Anzahl Entwickler:

Folgende Frameworks wurden evaluiert:

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Falls Sie kein eigenes Framework entwickelt haben, geben Sie bitte den Aufwand für die Prüfung (Evaluation) von Frameworks bis zum tatsächlichen konkreten Einsatz an.

- Teil 4: Objektorientierte Modellierung -

In welchem Umfang betreiben Sie die objektorientierte Modellierung mit UML?

- 1. Gar nicht bzw. es existieren keine Vorgaben, jeder Softwareentwickler kann, muss aber nicht UML nutzen.
- 2. In der Analysephase werden auf Basis der Anwendungsfalldiagramme (Use Cases) oder anderer Beschreibungen Klassendiagramme erstellt. Weitere Modellierung wird nicht betrieben.
- 3.(umfasst 2.) Es werden im weiteren Verlauf durch Grob- und Feinkonzept die Klassendiagramme verfeinert und ggf. Interaktionsdiagramme zur genaueren Beschreibung erstellt. In der Realisierungs- bzw. reinen Entwicklungsphase wird nicht modelliert.
- 4.(umfasst 3.) In allen Phasen der Softwareentwicklung werden UML-Diagramme erstellt, insbesondere jedes Programm im Vorwege modelliert. Die Programme können allerdings später vom Modell abweichen.
- 5.(umfasst 4.) Es wird eine komplett modellgetriebene Softwareentwicklung betrieben, d.h. es existieren insbesondere auch zu allen Programmen ggf. durch Reverse-Engineering aktuelle Modelle/Diagramme.

Anmerkungen:

- Frage kann ich nicht beantworten

zur nächsten Frage.. | Eingabe löschen

Hinweise/Hilfe zur Frage:
Für ergänzende Anmerkungen nutzen Sie ggf. bitte das untere Eingabefeld.

- Teil 5: Framework-Bewertung -

Bitte beantworten Sie bezogen auf den bisherigen Einsatz von Frameworks in Ihrem Unternehmen folgende Fragen:

	Ja	Nein	Frage kann ich nicht beantworten
Stabilität und Sicherheit: Hat sich die Stabilität und Sicherheit in den durch Ihre Frameworks abgedeckten Bereichen verbessert?	☐	☐	☐
Funktionieren die eingesetzten Frameworks fehlerfrei?	☐	☐	☐
Gibt es Interoperabilitätsprobleme mit anderen Produkten (Protokolle, Datenformate)?	☐	☐	☐
Verteilungs- / Netzwerkaspekte: Sind die mit Ihren Frameworks erstellten Applikationen im erforderlichen Rahmen skalierbar?	☐	☐	☐
War eine Lastverteilung für Sie relevant und wird sie durch den Framework-Einsatz / die Verteilungsplattform unterstützt?	☐	☐	☐
Benutzerakzeptanz: Haben ihre Anwender Nachteile wahrgenommen etwa durch Umstellung auf Internetanwendungen?	☐	☐	☐
Wirtschaftlichkeit: Haben sich ihre Erwartungen bezüglich Kostenersparnis durch den Framework-Einsatz bisher erfüllt?	☐	☐	☐
Hat der Frameworkeinsatz eine Wiederverwendung von Programmen, Komponenten etc. bewirkt? .	☐	☐	☐

zur nächsten Frage.. | Eingabe löschen

- Teil 6: Allgemeine Fragen -

Bitte beantworten Sie nun kurz noch die folgenden Fragen zum Unternehmen:

Branche: _____ | Anzahl Standorte:

Anzahl Mitarbeiter: in der IT: | Ihre Funktion/Tätigkeit: _____

Art der entwickelten Software im Unternehmen:

- ☐ Betriebswirtschaftliche SW
- ☐ Multimedia/Internet SW
- ☐ Technische SW
- ☐ Systemsoftware
- ☐ Software-Komponenten
- ☐ Embedded-Systems
- ☐ HW/SW Systemlösungen
- ☐ Sonstige SW: _____

Verwendete Programmiersprachen im Unternehmen:

- ☐ Java
- ☐ C#(.NET)
- ☐ C++ (C)
- ☐ Cobol
- ☐ PL/I
- ☐ VB (Basic)
- ☐ Delphi (Pascal)
- ☐ Andere: _____

Eingesetzte Entwicklungsumgebungen (IDE):

- ☐ IBM Websphere ☐ Borland JBuilder
- ☐ Eclipse ☐ Borland Together Edition for Eclipse
- ☐ JDeveloper ☐ Andere IDE: _____

Eingesetzte UML-Modellierungs-Tools:

- ☐ Rational Rose
- ☐ Eclipse UML
- ☐ Andere: _____

Im Unternehmen eingesetzte Application-Server:

J2EE: ☐ IBM Websphere ☐ BEA Weblogic ☐ Oracle-AS ☐ JBOSS ☐ Andere .NET: ☐ Microsoft

zur nächsten Frage.. | Eingabe löschen

Liebe Teilnehmerin/lieber Teilnehmer,

Die Befragung ist nun beendet.
Wir bedanken uns an dieser Stelle noch einmal für Ihre Unterstützung.
Sobald die Umfrageergebnisse ausgewertet sind, können Sie sich als kleines Dankeschön das Ergebnis der Studie zusenden lassen.
Bei Interesse geben Sie bitte Ihre email-Adresse an.
Darüber hinaus verlosen wir Amazon-Gutscheine im Wert von 5 X 20 €. Falls Sie an der Verlosung teilnehmen möchten, benötigen wir hierfür ebenfalls Ihre email-Adresse.
Die email wird getrennt von den Umfragedaten gespeichert.
Sie können damit sicher sein, dass von Ihrer Email-Adresse nicht auf Ihre Umfrageergebnisse rückgeschlossen werden kann!

email-Adresse:

☐ Ich möchte an der Verlosung teilnehmen.

☐ Ich möchte das Ergebnis der Studie zugesendet bekommen.

Senden

(c) 2005 Ute Stegmann DV-Kauffrau, Informatik-Studentin an der FernUniversität in Hagen
am Fachbereich Informatik - Lehrgebiet Programmiersysteme - Prof. Dr. Friedrich Steimann

FernUniversität in Hagen **Umfrage: Einsatz von (Java-)Frameworks in verteilten Systemen** **-Ende-**

Ende der Umfrage.

Sie können nun das Fenster schließen.

(c) 2005 Ute Stegmann DV-Kauffrau, Informatik-Studentin an der FernUniversität in Hagen
am Fachbereich Informatik - Lehrgebiet Programmiersysteme - Prof. Dr. Friedrich Steimann