SCORE
Integration Suite
DELTA
software
technology

ObjectBridge EJB Edition

Enterprise JavaBeans (EJB) – ein Begriff der im Bereich unternehmenskritischer Anwendungen mehr und mehr auf Resonanz stößt. Als serverseitige Komponentenarchitektur für den unternehmensweiten Einsatz unterstützt Sie EJB bei der Steuerung erfolgsentscheidender Geschäftsprozesse. ObjectBridge EJB Edition hilft Ihnen bei der Entwicklung solch unternehmenskritischer Anwendungen, sowohl unter Berücksichtigung bereits vorhandener Anwendungen als auch im Bereich des Internet-Einsatzes.

ObjectBridge EJB Edition wird als Zusatzkomponente unter SCORE Integration Suite eingesetzt und ermöglicht einen objektbasierten Zugriff von Java- oder HTML-Clients auf beliebige Server-Komponenten. Servlets, Java Server-Pages oder auch andere Enterprise JavaBeans können die von ObjectBridge EJB Edition generierten EJB-Proxies auch über mehrere Applikationsserver hinweg ansprechen. Dabei bieten die EJB-Proxies sowohl den Zugriff auf objektorientierte Server-Anwendungen (z.B. Java- oder C++-Server), als auch auf nicht objektorientierte Server-Anwendungen (z.B. COBOL-Programme).

INHALT

PROXY GENERIERUNG

ObjectBridge EJB Edition generiert EJB-Proxies, zusammen mit den für das Deployment benötigten Deployment Descriptoren, als vollständige, lauffähige Enterprise JavaBeans.

Mit Hilfe dieser generierten EJB-Proxies ist es einer Client-Anwendung möglich, auf eine beliebige Server-Komponente zuzugreifen. Dabei müssen die Java-Programmierer nicht mehr wissen, wie eine Komponente letztendlich implementiert wurde, da der Aufruf der Server über generierte Java-Klassen (EJB-Proxies) erfolgt.

Indem die EJB-Proxies die Methodenaufrufe automatisch in entsprechende Server-Requests umsetzen, wird eine Server-Komponente so angesprochen als wäre sie selbst eine in Java geschriebene Klasse mit zugehörigen Methoden.

Die Umsetzung der Java-Datentypen in die entsprechenden Datentypen auf der Server-Seite bzw. umgekehrt erfolgt hierbei automatisch in den generierten Klassen.

VOM CLIENT ZUM SERVER

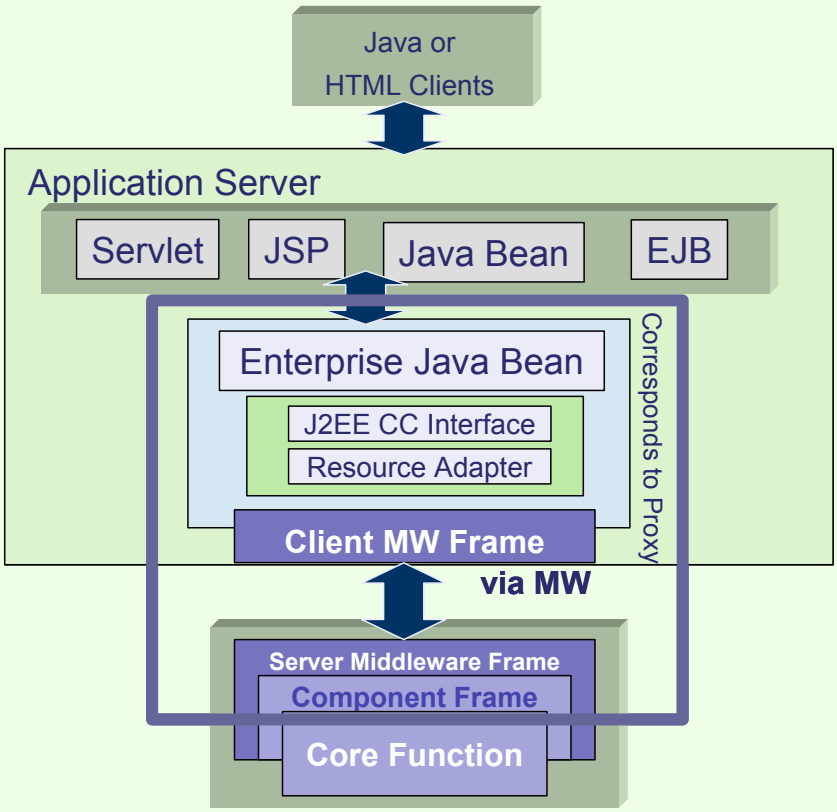
Der Zugriff auf eine Server-Komponente wird über Enterprise JavaBeans (Session Beans) realisiert, die mittels ObjectBridge EJB Edition automatisch generiert werden. Direkt nachdem die EJBs kompiliert und deployed worden sind, können sie in einem Applikationsserver verwendet werden.

Über die Methoden der in einer Client-Anwendung verwendeten Enterprise JavaBeans kann anschließend die Server-Komponente aufgerufen werden, ohne dass die Client-Anwendung die Implementierung der Server-Komponente kennen muss. Dabei werden lediglich die äußeren Schnittstellen (Interfaces) der Server-Komponente, die die generierten EJBs als Methoden zur Verfügung stellen, verwendet.

Mit Hilfe eines sogenannten Resource Adapters und eines Client Middleware Frames (CMF) wird eine Verbindung zur verwendeten Middleware aufgebaut, um von dort über den Server Middleware Frame (SMF) und den Server Component Frame (SCF) zum eigentlichen Server-Modul zu gelangen. Dabei entspricht der erwähnte Resource Adapter dem JCA Standard (J2EE Connector Architecture) von Sun Microsystems und ist mit ObjectBridge EJB Edition verfügbar. CMF, SMF und SCF werden von SCORE Integration Suite zur Verfügung gestellt bzw. von dort aus generiert.

Die strikte Schichtarchitektur in Verbindung mit der Verwendung der Generatoren erleichtert die weitere Entwicklung von Anwendungen und sichert für Client- und Server-Komponenten die Unabhängigkeit von Änderungen in der technischen Infrastruktur.

Die unten stehende Grafik verdeutlicht die Technik der Kommunikation zwischen Client und Server:



Struktur einer verteilten Anwendung mit ObjectBridge EJB Edition

PERFORMANCE STEIGERUNG

Um die Anzahl der Aufrufe zwischen Client und Server zu reduzieren, können für eine Komponente Optimierungen in Form von Request- und Response-Packaging in der Definition der Server-Schnittstellen spezifiziert werden. Dabei führen die EJB-Proxies das Packaging automatisch durch und steigern somit die Performance der Anwendung ohne dass diese verändert werden müssen.

DIE KOMPONENTEN DER OBJECTBRIDGE

ObjectBridge EJB Edition besteht aus:

- **dem Generator für die Erzeugung der Enterprise JavaBeans**
- **dem Archiv der Java Basis-Klassen**
- **dem Resource Adapter und**
- **den Middleware Interface (Agent)**

DER OBJECTBRIDGE GENERATOR

Der ObjectBridge Generator ist verantwortlich für die Erzeugung der Enterprise JavaBeans. Ausgangspunkt für die Generierung dieser Klassen ist ein Component Repository, das die Definitionen der Server-Komponente, d.h. die Beschreibung der Schnittstellen, Objekte, Methoden, Compound Operations, etc. enthält. Die Informationen des Component Repository sind im XML-Format gespeichert und können damit auch mittels anderer Werkzeuge ausgewertet werden.

Der ObjectBridge Generator interpretiert diese Informationen und erstellt die entsprechenden Java-Klassen (Proxies) als Enterprise JavaBeans.

DAS ARCHIV DER JAVA-BASISKLASSEN

ObjectBridge EJB Edition enthält bereits vorgefertigte Java-Basisklassen. Diese werden von den generierten Java-Proxies verwendet und enthalten u.a. Methoden zur Konvertierung von Datentypen sowie Exception-Klassen für die Fehlerbehandlung.

DER RESOURCE ADAPTER

Da die Enterprise JavaBeans Specification Standard keinen direkten Zugriff von EJBs auf nicht mit Java realisierte Anwendungen erlaubt, wird von ObjectBridge EJB Edition ein Resource Adapter zur Verfügung gestellt, der die Verbindung zwischen dem Applikationsserver (sprich: den EJBs) und den serverseitigen Anwendungen herstellt. Dabei entspricht der mitgelieferte Resource Adapter dem JCA Standard (J2EE Connector Architecture), der von Sun Microsystems, zusammen mit anderen namhaften Herstellern (z.B. IBM, BEA, Inprise, SAP AG, ...), entwickelt wurde.

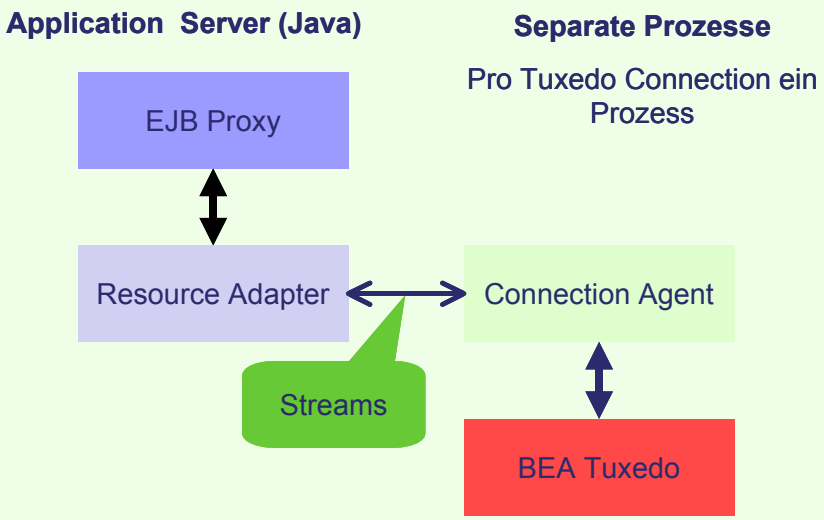
MIDDLEWARE INTERFACE (AGENT)

Der Aufruf entfernter Server ist abhängig von der verwendeten Middleware. Aus diesem Grund stellt ObjectBridge EJB Edition ein weiteres Modul – die Schnittstelle zur entsprechenden Middleware – zur Verfügung. Über diese vordefinierten Schnittstellen, den sogenannten Agents (C-DLLs), wird die Verbindung vom Resource Adapter zur Middleware hergestellt.

Sowohl für die jeweils benutzte Middleware als auch für lokale Serveraufrufe stellt die ObjectBridge EJB Edition die entsprechenden Agents bereit. Zusammen mit dem Resource Adapter bilden die Agents die eigentliche Schnittstelle zwischen den EJB-Proxies und der verwendeten Middleware.

Da also bereits die middleware-spezifischen Abläufe im Resource Adapter bzw. in den vorgefertigten Agents enthalten sind, kann eine Client-Anwendung entwickelt werden, ohne Rücksicht auf die verwendete Middleware nehmen zu müssen. Dies bedeutet auch, dass bei einem Austausch der Middleware keine Änderungen in den Java-Clients notwendig sind.

Für jede Middleware, an dieser Stelle beispielsweise BEA Tuxedo, sieht die Struktur der Kommunikation mit den generierten EJB Proxies folgendermaßen aus:



Struktur des Aufrufs der Middleware aus Sicht der generierten EJB-Proxies

VERFAHREN

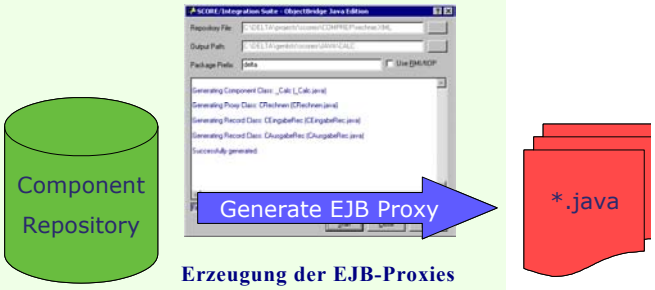
Vorgehensweise:

- **Generierung der Java Klassen**
- **Kompilierung der Enterprise JavaBeans und Archivbildung**
- **Deployment der gebildeten Packages**

Bei Verwendung von Java Development Kit (JDK) von Sun oder von Java-Entwicklungsumgebungen, die darauf basieren, kann der komplette Entwicklungsprozess einschließlich der Erstellung der kompilierten Java-Klassen innerhalb von SCOUT² durchgeführt werden.

GENERIERUNG

Als Basis für die Generierung der EJB Proxies wird ein Component Repository verwendet, das alle relevanten Informationen über eine Server-Komponente einschließlich deren Schnittstellen enthält.



Erzeugung der EJB-Proxies

Die Generierung erzeugt automatisch die entsprechenden Enterprise JavaBeans sowie einen dazugehörigen Deployment Descriptor.

COMPILE

Die so entstandenen Enterprise JavaBeans werden nun kompiliert und sodann mit dem Deployment Descriptor zu einem Package (Java-Archiv) zusammengefasst.

Erfolgt dabei die Erstellung des Package direkt unter der Development Middleware SCOUT², wird die genaue Abbildung der Package-Struktur auf die für die weitere Verwendung notwendige Verzeichnisstruktur automatisch durchgeführt, ebenso wie die automatische Ablage des Deployment Descriptors.

Verwenden Sie das Java Development Kit (JDK) von Sun, können Sie den Compile-Vorgang komfortabel innerhalb von SCOUT² direkt von SCOUT² ausführen. Bei Verwendung anderer Werkzeuge (z.B. IBM Visual Age) müssen Sie die erzeugten Java-Sourcen in ihre Entwicklungsumgebung importieren.

DEPLOYMENT

Für die Verwendung der EJBs durch Java-Clients werden die erstellten Packages anschließend in einen Applikationsserver deployed.

Das herausragende Merkmal dieses Verfahrens ist, dass die Client-Anwendungen mit Hilfe der generierten, kompilierten und deployten EJB-Proxies die Funktionen des entsprechenden Servers aufrufen als wären diese selbst in Java geschrieben.

GENERIERTE KLASSENTYPEN

Im einzelnen erzeugt der ObjectBridge Generator aus dem Component Repository die nachfolgend aufgeführten Java-Klassen:

- **die Server Klasse**
- **die Interface Klasse**
- **die Operation Parameter Klasse**

DIE SERVER-KLASSE

Für jede Server-Komponente wird zunächst anhand des Component Repository eine Java-Klasse, die sogenannte Server-Klasse, einschließlich eines zugehörigen Remote- und Home-Interfaces als Enterprise JavaBean generiert.

Diese Server-Klasse verarbeitet alle eingehenden Java Methodenaufrufe, bildet aus diesen Compound Operations und ruft die zugehörige Server-Komponente auf. Außerdem ist sie für die optimierte Ausführung der Operationen, das Request-/Response-Packaging, zuständig.

DIE INTERFACE-KLASSE

Ebenso wird für jedes Interface einer Server-Komponente eine Enterprise JavaBean, einschließlich eines zugehörigen Remote- und Home-Interfaces, erzeugt. Diese EJB enthält Java Methoden für alle Operationen, die das Server-Interface zur Verfügung stellt.

DIE OPERATION-PARAMETER-KLASSE

Für jeden in einer Operation als Struktur definierten Parameter erzeugt der Generator ebenfalls eine eigene Klasse, die Operation-Parameter-Klasse. Eine solche Java-Klasse enthält alle Datenelemente des Operationsparameters als public-Felder vom Typ `String`. Die Felder werden automatisch in die für die Server-Komponente notwendigen Datentypen konvertiert und umgekehrt.

SYSTEMVORAUSSETZUNGEN

... FÜR DIE GENERIERUNGSUMGEBUNG:

Hardware	IBM-kompatibler PC mit Prozessor der Pentium-Klasse
	128 MB Hauptspeicher (minimal)
Betriebssysteme	Microsoft Windows® NT 4.0 / 2000 / XP
Software	SCORE Integration Suite ab 1.5
	SCOUT²

... FÜR DIE LAUFZEITUMGEBUNG:

Betriebssysteme	Microsoft Windows® NT 4.0 / 2000 / XP
	Sun Solaris®
	IBM AIX
	Weitere Unix Systeme auf Anfrage
Software	Sun JDK 1.3 oder höher kompatible Java VM
	Ein auf J2EE 1.3 basierender Application Server, der die Java Connector Architecture (JCA) unterstützt.

MA 13'958.03

www.d-s-t-g.com