

Steffen Hiekel

**Bedeutung und Qualitätseigenschaften des
Enterprise Service Bus im Kontext von
serviceorientierten Architekturen**

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren





OTTO-VON-GUERICKE UNIVERSITÄT MAGDEBURG

FAKULTÄT FÜR INFORMATIK
INSTITUT FÜR VERTEILTE SYSTEME
ARBEITSGRUPPE SOFTWARETECHNIK



Diplomarbeit

Bedeutung und Qualitätseigenschaften des Enterprise Service Bus im Kontext von serviceorientierten Architekturen

Verfasser: **Steffen Hiekel**

Magdeburg, den 22. Januar 2007

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufbau	2
1.3	Einordnung	2
1.4	Fazit	9
2	Der ESB aus Sicht der Forschung	10
2.1	Herangehensweise	10
2.2	Definition des ESB in der Forschung	12
2.2.1	Definition nach Schulte (Gartner)	12
2.2.2	Definition nach Kischel (OBJEKTspektrum)	13
2.2.3	Definition nach Chappell	16
2.2.4	Definition nach Dostal/Jeckle/Melzer/Zengler	22
2.2.5	Definition nach Lorenzelli-Scholz (OBJEKTspektrum)	24
2.2.6	Definition nach Rieks (IM)	25
2.2.7	Definition nach Tieke (InformationWeek)	26
2.2.8	Definition nach Vollmer/Gilpin (Forrester)	27
2.3	Vergleich der Definitionen	28
2.4	Fazit	31
3	Der ESB aus Sicht der Softwareindustrie	32
3.1	Herangehensweise	32
3.2	JB1 Spezifikation	35
3.3	ESB nach Progress Software (ehemals Sonic Software)	36
3.4	ESB nach Fiorano	40

3.5	ESB nach Cape Clear	45
3.6	ESB nach BEA	51
3.7	ESB nach Oracle	56
3.8	ESB nach MuleSource	59
3.9	ESB nach Microsoft	62
3.10	ESB nach Sun Microsystems	64
3.11	ESB nach Red Hat (JBoss)	66
3.12	Vergleich der ESB-Implementierungen	67
3.13	Fazit	69
4	Qualitätseigenschaften des ESB	70
4.1	Herangehensweise	70
4.2	Allgemeine Qualitätseigenschaften	70
4.3	Spezielle Qualitätseigenschaften	72
4.4	ESB-Qualitätsmodell	72
4.4.1	Performanz	72
4.4.2	Sicherheit	73
4.4.3	Funktionsumfang/Funktionalität	74
4.4.4	Benutzbarkeit	75
4.4.5	Testbarkeit, Integrierbarkeit	77
4.4.6	Wartbarkeit	78
4.4.7	Plattformunabhängigkeit/Portierbarkeit	79
4.4.8	Skalierbarkeit	81
4.4.9	Wiederverwendbarkeit	82
4.4.10	Standardunterstützung	83
4.4.11	Routingfähigkeiten	84
4.4.12	Transformationsfähigkeiten	85
4.5	Zusammenfassung	86
5	Bewertung von verfügbaren ESB-Implementierungen	88
5.1	Herangehensweise	88
5.2	Anwendung des Qualitätsmodells	88
5.2.1	Performanz	88
5.2.2	Sicherheit	89

5.2.3	Funktionsumfang	90
5.2.4	Benutzbarkeit	91
5.2.5	Testbarkeit, Integrierbarkeit	91
5.2.6	Wartbarkeit	91
5.2.7	Plattformunabhängigkeit/Portierbarkeit	92
5.2.8	Skalierbarkeit	92
5.2.9	Wiederverwendbarkeit	93
5.2.10	Standardunterstützung	93
5.2.11	Routingfähigkeiten	94
5.2.12	Transformationsfähigkeiten	94
5.3	Vergleich	94
6	Zusammenfassung und Ausblick	96
6.1	Zusammenfassung	96
6.2	Ausblick	98
A	Transformation von Schnittstellenbeschreibungen	99
B	Standards	101
C	ESB-Anbieter	108
D	Marktstudien nach Forrester	110
E	Event Stream Processing	112
F	Anwendung des ESB-Qualitätsmodells	113

Abkürzungsverzeichnis

(UN/)EDIFACT	...	(United Nations/) <u>E</u> lectronic <u>D</u> ata <u>I</u> nterchange <u>F</u> or <u>A</u> dmistration
ACL	<u>A</u> ccess <u>C</u> ontrol <u>L</u> ist
AES	<u>A</u> dvanced <u>E</u> ncryption <u>S</u> tandard
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
B2B	<u>B</u> usiness-to- <u>B</u> usiness
BAM	<u>B</u> usiness <u>A</u> ctivity <u>M</u> onitoring
BC	<u>B</u> inding <u>C</u> omponent
BI	<u>B</u> usiness <u>I</u> ntelligence
BIOS	<u>B</u> asic <u>I</u> nput <u>O</u> utput <u>S</u> ystem
BPEL	<u>B</u> usiness <u>P</u> rocess <u>E</u> xecution <u>L</u> anguage
BPEL4WS	<u>B</u> usiness <u>P</u> rocess <u>E</u> xecution <u>L</u> anguage <u>f</u> or <u>W</u> eb <u>S</u> ervices
BPM	<u>B</u> usiness <u>P</u> rocess <u>M</u> anagement
BPMN	<u>B</u> usiness <u>P</u> rocess <u>M</u> odeling <u>N</u> otation
CASE	<u>C</u> omputer- <u>A</u> ided <u>S</u> oftware <u>E</u> ngineering
CBR	<u>C</u> ontent <u>B</u> ased <u>R</u> outing
CBSE	<u>C</u> omponent- <u>B</u> ased <u>S</u> oftware <u>E</u> ngineering
COM	<u>C</u> omponent <u>O</u> bject <u>M</u> odel
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CRM	<u>C</u> ustomer <u>R</u> elationship <u>M</u> anagement
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alues
cXML	<u>C</u> ommerce <u>X</u> ML
DCOM	<u>D</u> istributed <u>C</u> omponent <u>O</u> bject <u>M</u> odel
DIN	<u>D</u> eutsches <u>I</u> nstitut für <u>N</u> ormung e. V.
DoS	<u>D</u> enial of <u>S</u> ervice
DPF	<u>D</u> istributed <u>P</u> rocessing <u>F</u> ramework
DTD	<u>D</u> ocument <u>T</u> ype <u>D</u> efinition
EAI	<u>E</u> nterprise <u>A</u> pplication <u>I</u> ntegration
ebXML	<u>E</u> lectronic <u>B</u> usiness <u>X</u> ML
EDA	<u>E</u> vent <u>D</u> riven <u>A</u> rchitecture
EDI	<u>E</u> lectronic <u>D</u> ata <u>I</u> nterchange
EI	<u>E</u> nterprise <u>I</u> ntegration
EJB	<u>E</u> nterprise <u>J</u> ava <u>B</u> eans
EN	<u>E</u> uronorm
ERP	<u>E</u> nterprise <u>R</u> esource <u>P</u> lanning
ESB	<u>E</u> nterprise <u>S</u> ervice <u>B</u> us
ESP	<u>E</u> vent <u>S</u> tream <u>P</u> rocessing
FCM-Modell	<u>F</u> actor= <u>C</u> riteria= <u>M</u> etrics= <u>M</u> odell
FTP	<u>F</u> ile <u>T</u> ransfer <u>P</u> rotocol

GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
IIOB	Internet Inter ORB Protocol
ISO	International Organization for Standardization
IT	Informationstechnologie
J2EE	Java 2 Enterprise Edition
JAAS	Java Authentication and Authorization Service
JB	Java Business Integration
JCA	J2EE Connector Architecture
JCP	Java Community Process
JDBC	Java Database Connectivity
JMS	Java Message Service
JMX	Java Management Extensions
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JSR	Java Specification Request
JSSE	Java Secure Socket Extension
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MFL	Message Format Language
MOM	Message Oriented Middleware
NMR	Normalized Message Router
OASIS	Organization for the Advancement of Structured Information Standards
OLE	Object Linking and Embedding
OMG	Object Management Group
ORB	Object Request Broker
QoS	Quality of Service
QoP	Quality of Protection
RELAX NG	REgular LAnguage for XML Next Generation
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SCM	Supply Chain Management
SE	Service Engine
SGML	Standard Generalized Markup Language
SLA	Service Level Agreements
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SPI	Service Provider Interface
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UBL	Universal Business Language
UDDI	Universal Description Discovery and Integration
UML	Unified Modelling Language

W3C	<u>W</u> orld <u>W</u> ide <u>W</u> eb <u>C</u> onsortium
WCF	<u>W</u> indows <u>C</u> ommunication <u>F</u> oundation
WS	<u>W</u> eb <u>S</u> ervices
WS-BPEL	<u>W</u> eb <u>S</u> ervices <u>B</u> usiness <u>P</u> rocess <u>E</u> xecution <u>L</u> anguage
WSDL	<u>W</u> eb <u>S</u> ervices <u>D</u> escription <u>L</u> anguage
WSFL	<u>W</u> eb <u>S</u> ervices <u>F</u> low <u>L</u> anguage
WSIL	<u>W</u> eb <u>S</u> ervice <u>I</u> nspection <u>L</u> anguage
WSLA	<u>W</u> eb <u>S</u> ervice <u>L</u> evel <u>A</u> greement
XACML	<u>e</u> Xtensible <u>A</u> ccess <u>C</u> ontrol <u>M</u> arkup <u>L</u> anguage
xCBL	<u>X</u> ML <u>C</u> ommon <u>B</u> usiness <u>L</u> ibrary
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage
XMPP	<u>E</u> xtensible <u>M</u> essaging and <u>P</u> resence <u>P</u> rotocol
XPath	<u>X</u> ML <u>P</u> ath <u>L</u> anguage
XQuery	<u>X</u> ML <u>Q</u> uery <u>L</u> anguage
XSL	<u>E</u> xtensible <u>S</u> tylesheet <u>L</u> anguage
XSLT	<u>X</u> SL <u>T</u> ransformations

Abbildungsverzeichnis

2.1	Enterprise Service Bus nach KISCHEL	14
2.2	Enterprise Service Bus nach CHAPPELL	20
2.3	Föderierter ESB nach CHAPPELL	20
2.4	Inkrementelle Anpassung an den ESB nach CHAPPELL	20
2.5	Integrationsansätze nach CHAPPELL	22
2.6	Deployment-Szenarien des ESB nach LORENZELLI-SCHOLZ	25
2.7	ESB-Service-Container nach CHAPPELL	31
3.1	Service-Container nach JBI	35
3.2	Sonic ESB Produktfamilie	36
3.3	Sonic ESB-Architektur	38
3.4	Sonic ESB Beispiel	40
3.5	Fiorano SOA Plattform	41
3.6	Fiorano Enterprise Service Bus	43
3.7	Cape Clear ESB Plattform	45
3.8	Cape Clear Sicherheitsframework	48
3.9	BEA AquaLogic Data Services Platform	52
3.10	BEA AquaLogic Service Bus	54
3.11	Oracle Web Services Manager	57
3.12	Mule ESB	60
3.13	Mule UMO-/Service-Container	60
3.14	Mule Nachrichtentransport	61
3.15	Aufbau des Open ESB	66
D.1	Forrester Wave: Enterprise Service Bus, ESB Suites, Q4 '05	110

D.2	Forrester Wave: Enterprise Service Bus, Comprehensive ESB Suites, Q4 '05	111
D.3	Forrester Wave: Enterprise Service Bus, Q2 '06	111
E.1	Aufbau und Bestandteile einer ESP-Engine nach PALMER	112
E.2	Einbettung einer ESP-Engine nach PALMER	112
F.1	Deployment-Topologie des gewählten Szenarios.	114
F.2	Interaktion eines Aufrufs	114
F.3	Screenshot von JMeter	116
F.4	Antwortzeiten (AZ) im Vergleich	120
F.5	Anzahl paralleler Prozesse (PP) im Vergleich	121
F.6	Nachrichtendurchsätze (ND) im Vergleich	121
F.7	FU im Vergleich	122
F.8	BE im Vergleich	122
F.9	TI im Vergleich	123
F.10	PU und PUG im Vergleich	123
F.11	ST im Vergleich	124
F.12	Häufigkeitsverteilung der Qualitätsmetriken	124

Tabellenverzeichnis

3.1	Vergleich der ESB-Implementierungen.	68
4.1	Betriebssysteme	80
4.2	Zusammenfassung der Qualitätsmaße/-metriken.	87
5.1	Qualitätsmaße und -metriken der untersuchten ESB-Implementierungen.	95
B.1	Web Service Standards (WS-Standards).	102
B.2	XML-basierte Standards.	103
B.3	Protokoll Standards.	104
B.4	Java-basierte Standards.	105
B.5	Proprietäre Standards.	106
B.6	Sonstige Standards.	106
B.7	Standardkategorien.	107
C.1	ESB-Anbieter und -Produkte.	109
C.2	ESB-Anbieter Unternehmensdaten.	109
F.1	Plattformeigenschaften des gewählten Szenarios.	113

Listingverzeichnis

A.1	IDL-Datei	99
A.2	WSDL-Datei	99
F.1	WSDL-Schnittstellenbeschreibung	115
F.2	SOAP-Nachricht (Anfrage)	115
F.3	SOAP-Nachricht (Antwort)	115
F.4	Testplan des Lasttreibers	116

Kapitel 1

Einleitung

1.1 Motivation

In den letzten Jahren hat sich das Konzept der serviceorientierten Architektur (SOA) zu dem entscheidenden Ziel für moderne Geschäftsanwendungen entwickelt und glaubt man verschiedenen Softwareherstellern, so ist eine SOA bereits heute realisierbar. Den Kern einer solchen Architektur sollen Web Services und der so genannte Enterprise Service Bus (ESB) bilden. Hat sich die Web Service Technologie bereits über mehrere Jahre hinweg entwickelt und etabliert, so steckt der ESB hingegen noch in den Kinderschuhen. Dennoch sehen Analysten aber auch die Industrie selbst den ESB als zukünftige Schlüsseltechnologie für den Aufbau einer SOA.

Mittlerweile hat sich ein umfangreicher Markt an ESB-Produkten entwickelt, die alle gegeneinander konkurrieren. Gepriesen wird – neben seinem im Vergleich zu traditionellen Integrations-Frameworks¹ geringeren Preis – vor allem seine Fähigkeit, Schwachstellen bisheriger Integrationsansätze zu überwinden. Für Kunden oder potenzielle Käufer stellt sich die Frage, was ihnen ein ESB bietet bzw. leisten kann und in wie weit verschiedene ESB-Produkte in Hinblick auf ihre Qualitätseigenschaften miteinander verglichen werden können.

Im Rahmen dieser Arbeit soll daher geklärt werden, was genau unter einem ESB zu verstehen ist und welche möglichen Unterschiede dabei zwischen Forschung und Industrie bestehen. Darüber hinaus sollen in der Softwareindustrie eingesetzte ESB-Implementierungen auf ihre Qualitätseigenschaften hin untersucht werden. Basierend darauf ist ein Modell zur Beurteilung von Qualitätseigenschaften für ESB zu entwickeln und dieses für frei verfügbare ESB-Implementierungen anzuwenden.

¹ Framework bezeichnet eine Rahmenstruktur bzw. ein Gerüst, das als Grundlage für weitere Entwicklungen dient.

1.2 Aufbau

Der Aufbau dieser Arbeit gliedert sich in mehrere Kapitel, wobei dieses Kapitel dazu dient, das Thema zu motivieren und im Weiteren eine grundlegende Einordnung zu liefern. Im Zentrum stehen dabei die SOA sowie dieser vorausgegangene bzw. sich anlehrende, aktuelle Konzepte. Damit sollen Entwicklungen und Parallelen aufgezeigt werden, die helfen, das Konzept des ESB zu verstehen und gleichfalls eine Beschreibung seines Umfeldes liefern. Im zweiten Kapitel geht es um den Begriff ESB und was darunter zu verstehen ist. Dabei interessieren vor allem die Ansichten aus Kreisen der Forschung und damit verbundene Begriffsdefinitionen. Von besonderer Bedeutung sind in diesem Zusammenhang die Fragen nach Aufbau, Eigenschaften und Funktionalitäten eines ESB. Daran anlehnend erfolgt im nächsten Kapitel eine umfangreiche und detaillierte Betrachtung von angebotenen ESB-Produkten. Hierbei soll gezeigt werden, was seitens der Softwareindustrie unter dem Begriff ESB verstanden wird und welche Technologien konkret zum Einsatz kommen. Gleichfalls herauszuarbeiten sind Parallelen sowie mögliche Differenzen zwischen den Sichtweisen in der Forschung und in der Industrie. Das folgende vierte Kapitel konzentriert sich auf die Betrachtung der Qualitätseigenschaften eines ESB. Im Rahmen eines zu erarbeitenden ESB-Qualitätsmodells werden hier allgemeine wie spezielle Qualitätseigenschaften aufgestellt und Kriterien zu deren Bewertung beschrieben. Die Anwendung des ESB-Qualitätsmodells auf frei verfügbare ESB-Implementierungen erfolgt im fünften Kapitel und endet mit einem abschließenden Vergleich. Den Schluss dieser Arbeit bildet eine zusammenfassende Betrachtung der gewonnenen Erkenntnisse sowie ein Ausblick auf zukünftige Entwicklungen und Themenstellungen im Kontext des ESB.

1.3 Einordnung

Im Rahmen einer ersten Einordnung erfolgt an dieser Stelle eine kurze Einführung in das Umfeld heutiger Geschäftsanwendungen und mit ihr verbundener Technologien wie Konzepte.

Rückblickend betrachtet, kann vor allem der Schritt hin zur objektorientierten Programmierung als ein Meilenstein angesehen werden und soll als Ausgangspunkt für die weiteren Betrachtungen dienen. Nicht nur, dass die objektorientierte Programmierung andere Paradigmen vor allem die funktionsorientierte Programmierung weitestgehend abgelöst hat, sie führte auch im Rahmen des Softwareengineerings zu neuen, objektorientierten Konzepten (bspw. der UML²). Die wesentlichen Eigenschaften, die für eine objektorientierte Programmierung sprechen und letztlich zu ihrem Durchbruch führten, sind (vgl. [Hei03], S. 8; [Som01], S. 270):

- bessere Beherrschbarkeit von komplexen Softwaresystemen durch Abstraktion, Kapselung und Vererbung,
- höhere Wiederverwendung,
- Vermeidung von redundantem Programmcode (bspw. durch Vererbung),
- daraus resultierend eine bessere Wartbarkeit.

² UML – Unified Modelling Language