

Inhalt

Einleitung	11
-------------------	-----------

Teil 1 Problemstellung

1 Von All-in-one-Software zu Webservices	17
---	-----------

1.1 Enterprise Application Integration	17
1.2 SAP-Technologie und -Komponenten	19
1.2.1 SAP NetWeaver	20
1.2.2 mySAP Business Suite	21
1.3 All-in-one versus Best-of-Breed	22

2 Middleware	27
---------------------	-----------

2.1 Proxy-Dienste	27
2.2 Proxy-Dienste im Sicherheitsbereich	28
2.3 Message-Queues	29
2.4 Workflow	30
2.5 GUI-Server	30
2.6 3-Tier-Lösungen	31

3 Präsentationsserver	35
------------------------------	-----------

3.1 Universelle API für die Ausgabe über das Web	35
3.2 Casabac GUI-Server	37
3.2.1 Technologie	37
3.2.2 Beispiel zur Kalkulation der Mehrwertsteuer	39
3.3 Web Dynpro und Business Server Pages	43

4 Beispielszenarien und Voraussetzungen 45

4.1	Die nahe Zukunft	45
4.1.1	XML-Datenbanken	45
4.1.2	Entwicklung in mehreren Programmiersprachen	45
4.1.3	Grid-Computing	46
4.2	Komponenten und Protokolle	47
4.3	Ein Beispiel einer heutigen Unternehmens-IT	48
4.3.1	Modellsichten und Abstraktion	51
4.3.2	Ansatz für eine zeitgemäße Lösung: Integration mit HTTP und XML	54

Teil 2 Remote Procedure Calls und Business API

5 SAP R/3 Remote Function Call 59

5.1	Was ist RFC?	59
5.2	R/3-RFC von einer anderen R/3-Instanz	61
5.3	Windows-zu-R/3-Connectivity mit DCOM	62
5.3.1	R/3-Logon mit VBA	64
5.3.2	Anatomie der RFC- und BAPI-Aufrufe von Windows	66
5.4	Helper-Klassen für den Zugriff auf R/3 via RFC	76
5.4.1	Klasse R3LogonObj	76
5.4.2	Klasse R3Credentials	77
5.4.3	Alternative R3Credentials-Klasse	78
5.5	R/3-Java-Connectivity	80
5.5.1	SAP R/3 Java Connector JCo	80
5.5.2	SAP R/3-Logon mit Java	80
5.6	Das SAP RFC-Gateway	83
5.6.1	Registrierung eines RFC-Listeners beim SAP-Gateway	83
5.6.2	Anlegen einer RFC-Destination für einen registrierten Listener	84
5.6.3	Eigene RFC-Listener erstellen	84
5.7	Troubleshooting RFC	87
5.8	Aufruf von Remote Programs aus R/3 Via RFC	88
5.8.1	RFC via Remote Shell	88
5.8.2	RFC über Webserver	89
5.8.3	Aufruf eines Programms auf der Workstation mit RFC	90
5.8.4	Aufruf eines HTTP-Webserver von SAP R/3	94
5.8.5	Proxy-Settings in ABAP	96
5.9	RFC von SAP R/3 auf Windows	97
5.9.1	Kommunikation mit Excel via OLE/2 und HTTP-Server	97
5.9.2	Zugriff auf ein Excel-Datenblatt von R/3	98

5.10	Business-Objekte und BAPI	105
5.10.1	Business-Objekte und Modellsichten	105
5.10.2	Technische Kriterien eines BAPI	109
5.10.3	Anwendungsbeispiel: Das Sales-Order-BAPI	110
5.10.4	Beispiel: BAPI_SALESORDER_GETLIST	111
5.10.5	VBA-Beispiel zu BAPI_SALESORDER_GETLIST	119
5.11	SAP Interface Repository	121

6 Message-Queues und Message-Server 125

6.1	Anwendungen von Message-Queues	125
6.1.1	Modelle von Message-Queues	125
6.1.2	Nutzen der Message-Queues	126
6.2	Message-Server-Software	128
6.3	Kommunikation mit Message-Queues	130

7 Internes Messaging in R/3 133

7.1	Messaging mit NAST und Workflow	133
7.1.1	NAST Message Control	133
7.1.2	Entwicklung eigener NAST-Handler	134
7.1.3	Workflow-Events	138
7.1.4	Workflow versus NAST	138
7.2	Messaging mit R/3-IDocs	139
7.3	SAP R/3-Workflow programmieren	142
7.4	Workflow-Handler	145

Teil 3 Programmieren in verteilten Systemen

8 Programmieren mit mehreren Programmiersprachen 149

8.1	Von Plattformen, Frameworks und Personalities	149
8.1.1	Plattformen	149
8.1.2	Frameworks und Personalities	151
8.2	Programmiersprachen und Entwicklungsumgebung	153
8.2.1	Die Entwicklungsumgebung	154
8.2.2	Die Compile-on-Demand-Runtime	154
8.2.3	Der Debugger	154
8.2.4	Einrichten einer Entwicklungsumgebung	155
8.2.5	Plattformsentscheidung: UNIX versus Windows NT	156
8.2.6	Framework: Web AS/ABAP, Microsoft.NET oder J2EE	156
8.2.7	Windows Scripting Host versus Visual Basic	158
8.2.8	Windows Classic versus Microsoft.NET	158

8.3	Die Wahl der richtigen Programmiersprache	159
8.3.1	Sprache für den Webserver	159
8.3.2	Java	160
8.3.3	J#	160
8.3.4	Visual Basic (VB) und Visual Basic for Applications (VBA)	160
8.3.5	Visual Basic Script (VBS) und ASP	160
8.3.6	C#	161
8.3.7	X#	161
8.3.8	Delphi	162
8.3.9	ADA	162
8.3.10	SmallTalk	163
8.3.11	LISP	163
8.4	Objektprogrammierung mit Java, VB und ABAP	164
8.4.1	Grundbegriffe	164
8.4.2	Constructor	171
8.4.3	Destructor	172
8.4.4	Property-Deklarationen	173
8.4.5	Methoden	174
8.4.6	String-Deklarationen	175
8.4.7	Array-Deklarationen	176

9 Webservice als Standard-Nachrichtenformat 179

9.1	Einfache Beispiele für Webservices	180
9.2	Ein Beispiel mit einem winzigkleinen Fehler	181
9.3	Die WSDL zum Amazon.com-Service	186
9.4	Webservice als HTTP GET-Anfrage	189
9.5	Verwendung eines XSLT-Stylesheets	189
9.6	Type- und Namespace-Referenz	189

10 HTTP- und TCP/IP-Protokolle 191

10.1	TCP/IP und UDP/IP Network Protocol	191
10.2	Client-Server-Protokolle	193
10.2.1	Clientseite	193
10.2.2	Serverseite	194
10.2.3	URN, URI und URL – Resource Locator	194
10.2.4	CGI – Common Gateway Interface	196
10.3	HTTP-Kommunikation mit einem Webserver	196
10.3.1	Beispiel einer HTTP-Session	199
10.3.2	Beispiel für einen HTTP POST-Request	201
10.3.3	Anfragen direkt ins Internet	203

11 XML – Extensible Markup Language 205

11.1	XML als Lingua franca des Internet	205
11.2	XSD-Schema und DTD	209
11.3	DOM – Document Object Model	213
11.4	XML-Parser	215
11.4.1	DOM oder SAX	216
11.4.2	Der Microsoft XML-Parser	216
11.4.3	XML-Parser-Libraries für Java	217
11.4.4	XML-Browser und Editoren	219
11.4.5	Namespaces – Namensräume	219
11.5	XSLT – Extensible Stylesheet Language Transformations	220
11.6	HTML-Seiten mit XSL-Stylesheets und XML	220
11.6.1	Inventurtabelle der Animal-Farm	221
11.6.2	XSL – Step by step	222
11.6.3	XSL-Templates	222
11.6.4	XSL auf die Animal-Farm angewandt	223
11.6.5	Anatomie des XSL-Stylesheets	224
11.6.6	Parsing des XML und Anwendung des XSL	227
11.7	SOAP	228
11.8	WSDL-Dokumente	231
11.8.1	WSDL-DOM	232
11.8.2	Beispiel eines WSDL-Dokuments für eine BAPI-Funktion	235

12 Entwicklung von dynamischen Webseiten 239

12.1	Server Pages und Scripting	239
12.1.1	Common Gateway Interface – CGI	240
12.1.2	Active Server Pages – ASP und ASP.NET	240
12.1.3	Java Server Pages – JSP	241
12.1.4	Java Servlets	243
12.1.5	Business Server Pages – BSP	244
12.2	Web Pages mit Active Server Pages	245
12.2.1	ASP »Hello World«	245
12.2.2	Senden von HTML	246
12.2.3	Aufrufe beliebiger Windows-Objekte aus VBS	247
12.2.4	BAPI-Aufruf von ASP	247
12.2.5	Microsoft.NET Connector	250
12.2.6	Parameter aus der URI abfragen	250
12.2.7	ASP Application Variables	253
12.2.8	Global.asa – Die ASP-Autoexec-Datei	254
12.2.9	Extra-Beispiel: File-Liste mit ASP	255
12.3	Entwicklung von Business Server Pages für den SAP Web AS	256
12.3.1	»Hello World« für BSP	256
12.3.2	BAPI-Aufruf aus einer BSP	257
12.3.3	Online Text Repository	259

13 Webseiten von anderen Programmen abrufen 261

13.1	Abfragen einer URI mit Visual Basic	261
13.1.1	Das Windows-Webbrowser-Control	261
13.1.2	Anwendungsbeispiel für das Browser-Control	262
13.1.3	Das WINHTTP-Control	266
13.1.4	Anwendungsbeispiel für das WINHTTP-Control	267
13.2	Webentwicklung mit SAP R/3	271
13.2.1	SAP Function HTTP_GET	272
13.2.2	Proxy-Settings in ABAP	275
13.2.3	SAP und SOAP: Webservice Choreography Interface (WSCI) ...	276
13.3	SAP Internet Transaction Server	276

Anhang

A Übersicht zu wichtigen Protokollen und Utilities 281

A.1	MIME-Typen	281
A.2	Beispiel für HTTP GET und HTTP POST	284
A.3	HTTP-Statuscodes	285
A.4	HTTP- und FTP-Kommandos	288
A.5	Java-Utilities	289
A.6	SAPLPD-Befehlszeilenparameter	292
A.7	Setzen des Suchpfads für SAPLOGON.INI	293
A.8	RFC-Connectoren	294
A.9	Beispiele aus dem SAP Interface Repository	298

B Synopsis der Programmiersprachen ABAP, VB und Java 301

C Der Autor 308

Index 309