

**Christian Schloz / Rainer Gross**

Entwicklung eines konfigurierbaren  
Softwaretools für Kreditinstitute auf  
Internetbasis

**Diplomarbeit**

## **Bibliografische Information der Deutschen Nationalbibliothek:**

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Copyright © 1998 Diplomica Verlag GmbH  
ISBN: 9783832422677

**Christian Schloz, Rainer Gross**

**Entwicklung eines konfigurierbaren Softwaretools für  
Kreditinstitute auf Internetbasis**

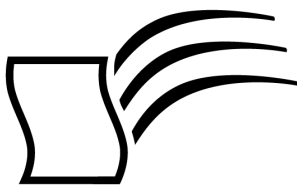


---

Christian Schloz / Rainer Gross

# Entwicklung eines konfigurierbaren Softwaretools für Kreditinstitute auf Internetbasis

Diplomarbeit  
an der Fachhochschule Konstanz  
Fachbereich Informatik  
Prüfer Prof. Dr. O. Bittel  
Juni 1998 Abgabe



**Diplomarbeiten Agentur**  
Dipl. Kfm. Dipl. Hdl. Björn Bedey  
Dipl. Wi.-Ing. Martin Haschke  
und Guido Meyer GbR

Hermannstal 119 k  
22119 Hamburg

agentur@diplom.de  
www.diplom.de

ID 2267

Schloz, Christian / Gross, Rainer: Entwicklung eines konfigurierbaren Softwaretools für Kreditinstitute auf Internetbasis / Christian Schloz / Rainer Gross – Hamburg: Diplomarbeiten Agentur, 2000  
Zugl.: Konstanz, Fachhochschule, Diplom, 1998

---

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Die Informationen in diesem Werk wurden mit Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden, und die Diplomarbeiten Agentur, die Autoren oder Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für evtl. verbliebene fehlerhafte Angaben und deren Folgen.

Dipl. Kfm. Dipl. Hdl. Björn Bedey, Dipl. Wi.-Ing. Martin Haschke & Guido Meyer GbR  
Diplomarbeiten Agentur, <http://www.diplom.de>, Hamburg 2000  
Printed in Germany



**Diplomarbeiten Agentur**

## **Wissensquellen gewinnbringend nutzen**

**Qualität, Praxisrelevanz und Aktualität** zeichnen unsere Studien aus. Wir bieten Ihnen im Auftrag unserer Autorinnen und Autoren Wirtschaftsstudien und wissenschaftliche Abschlussarbeiten – Dissertationen, Diplomarbeiten, Masterarbeiten, Staatsexamensarbeiten und Studienarbeiten zum Kauf. Sie wurden an deutschen Universitäten, Fachhochschulen, Akademien oder vergleichbaren Institutionen der Europäischen Union geschrieben. Der Notendurchschnitt liegt bei 1,5.

**Wettbewerbsvorteile verschaffen** – Vergleichen Sie den Preis unserer Studien mit den Honoraren externer Berater. Um dieses Wissen selbst zusammenzutragen, müssten Sie viel Zeit und Geld aufbringen.

**<http://www.diplom.de>** bietet Ihnen unser vollständiges Lieferprogramm mit mehreren tausend Studien im Internet. Neben dem Online-Katalog und der Online-Suchmaschine für Ihre Recherche steht Ihnen auch eine Online-Bestellfunktion zur Verfügung. Inhaltliche Zusammenfassungen und Inhaltsverzeichnisse zu jeder Studie sind im Internet einsehbar.

**Individueller Service** – Gerne senden wir Ihnen auch unseren Papierkatalog zu. Bitte fordern Sie Ihr individuelles Exemplar bei uns an. Für Fragen, Anregungen und individuelle Anfragen stehen wir Ihnen gerne zur Verfügung. Wir freuen uns auf eine gute Zusammenarbeit

### **Ihr Team der *Diplomarbeiten Agentur***

Dipl. Kfm. Dipl. Hdl. Björn Bedey —  
Dipl. Wi.-Ing. Martin Haschke —  
und Guido Meyer GbR —

Hermannstal 119 k —  
22119 Hamburg —

Fon: 040 / 655 99 20 —  
Fax: 040 / 655 99 222 —

[agentur@diplom.de](mailto:agentur@diplom.de) —  
[www.diplom.de](http://www.diplom.de) —

# Abstract

Die Diplomarbeit *Entwicklung eines konfigurierbaren Softwaretools für Kreditinstitute auf Internetbasis* diente der Neuentwicklung eines Systems, welches bestehende Serverapplikationen eines Kreditinstituts unter einheitlicher, durch eine Skriptsprache konfigurierbare Oberfläche zusammenfaßt. Die Neuerungen des in der vorliegenden Diplomarbeit beschriebenen Systems beziehen sich auf den zugrunde liegenden objektorientierten Systementwurf, die Client-Server-Architektur, die eingesetzte Skriptsprache sowie die Verwendung zeitgemäßer Internettechnologien.

Um die für das entwickelte System geeigneten Internettechnologien auswählen zu können, wurden unter anderem verschiedene WWW-Programmieretechniken, die Programmiersprache Java und das Modell JavaBeans sowie mehrere verteilte Objektsysteme und Mechanismen zur Datensicherheit in Netzwerken analysiert.

Das unter Berücksichtigung der Untersuchungsergebnisse entworfene System verwendet verteilte Objekte zur Realisierung der Client-Server-Kommunikation. Eine Clientapplikation wird anhand einer eigens dafür entworfenen, der des ursprünglichen Systems ähnlichen Skriptsprache entwickelt, wobei der Anwendungsprogrammierer sich nicht um das Objektmanagement zu kümmern braucht, gesteuert durch Skripte, wird es vom System selbst übernommen. Dabei geschieht die Zuweisung von Objektattributen mit Hilfe eines verteilten Parsingverfahrens, jedes durch ein Skript verursachtes Objekt besitzt seinen eigenen Parser und parametrisiert sich anhand des zugehörigen Skriptausschnitts selbst.

Ein nach diesem Systementwurf in der Sprache Java entwickelter Prototyp verwendet zur Kommunikation mit Serverapplikationen alternativ die Protokolle Java-RMI oder DCOM. Damit die Skriptsprache und somit das gesamte System um neue Funktionalität in Form von Variablen, Grafikkomponenten und Instruktionen erweitert werden kann, wurde ein spezieller Mechanismus implementiert, der es ermöglicht, ohne das System erneut zu übersetzen, zusätzliche Klassen mit samt der entsprechenden Skriptsyntax zu ergänzen.

Schließlich galt das Projekt der Erprobung eines disziplinierten Softwareentwicklungsverfahrens, weshalb besonderes Augenmerk auf das Projektmanagement sowie die Dokumentation des Projektverlaufs gerichtet wurde. In diesem Zusammenhang werden sowohl das *Capability Maturity Model* als auch die *Digital Program Methodology* vorgestellt.



# Inhaltsverzeichnis

Vorwort.....	VIII
Abkürzungsverzeichnis .....	X
Darstellungsverzeichnis.....	XII
Abbildungen.....	XII
Tabellen.....	XIII
1 Einleitung.....	1
1.1 Die Diplomarbeitsfirma.....	1
1.2 Das Projekt.....	1
1.3 Vorgehensweise zur Durchführung des Projekts .....	3
1.4 Projektziele.....	4
2 Das Entwicklungstool Vórtex.....	5
2.1 Die Entstehungsgeschichte von Vórtex.....	5
2.2 Der Aufbau von Vórtex.....	6
2.3 Der Transaktionsserver.....	7
2.4 Der Navigator.....	7
2.4.1 Die Navigatoroberfläche und -funktionalität .....	7
2.4.2 Die Navigatorskriptsprache.....	8
2.5 Der Dialogserver .....	10
2.6 Die Modulkommunikation .....	11
3 Untersuchung internetgeeigneter Technologien .....	13
3.1 Netzwerkcomputer .....	13
3.2 WWW-Programmiertechniken.....	15
3.2.1 Hypertext Markup Language (HTML).....	15
3.2.2 Hypertext Transfer Protocol (HTTP).....	17
3.2.3 Dynamische HTML-Seiten .....	18
3.2.3.1 CGI-Server.....	18
3.2.3.2 Active Server Pages.....	19
3.2.4 Java Script.....	20
3.3 Java.....	23
3.3.1 Übersicht zu Java .....	23
3.3.2 Unterschiede zwischen Java und C++ .....	24
3.3.3 Java-Applets.....	27
3.3.4 Exceptions in Java .....	28
3.3.5 Threads und Synchronisation in Java.....	31
3.3.6 Klassenpakete und Java-Bibliotheken .....	32
3.3.7 Möglichkeiten zur Optimierung von Java.....	33
3.3.8 Sicherheitsmechanismen von Java.....	34
3.4 JavaBeans .....	35
3.4.1 Ereignisse.....	36
3.4.1.1 Klassen zur Ereignisbehandlung.....	36

3.4.1.2 Ereignisadapter .....	38
3.4.2 Attribute von Beans .....	41
3.4.2.1 Getter- und Setter-Methoden .....	41
3.4.2.2 Bound-Properties .....	42
3.4.2.3 Constrained-Properties .....	43
3.4.3 Analyse von Beans .....	44
3.4.3.1 Analyse nach Namensgebungsregeln .....	45
3.4.3.2 Analyse durch Bean-Beschreiber .....	45
3.4.3.3 Der Bean-Analysator .....	45
3.4.4 Parametrisierung von Beans .....	46
3.4.4.1 Customizer .....	46
3.4.4.2 Attributeditoren .....	46
3.4.5 Speicherung von Beans .....	47
3.5 Objektorientierte verteilte Client-Server-Systeme .....	47
3.5.1 Prinzipien der Objektorientierung .....	47
3.5.2 Das Client-Server-Prinzip .....	49
3.5.3 Verteilte Objekte .....	50
3.5.4 CORBA .....	50
3.5.5 CORBA-Produkte .....	53
3.5.5.1 Funktionalität der untersuchten CORBA-Produkte .....	53
3.5.5.2 Test unterschiedlicher CORBA-Produkte .....	53
3.5.6 Remote Method Invocation (RMI) .....	56
3.5.7 Distributed Component Object Model (DCOM) .....	58
3.5.7.1 Kommunikation zwischen COM-Objekten .....	58
3.5.7.2 Marshaling .....	60
3.5.7.3 Definition eines COM-Objektes .....	60
3.5.7.4 Instanziierung von COM- und DCOM-Objekten .....	61
3.5.7.5 Anwendung von Methoden auf COM-Objekte .....	62
3.5.7.6 Sicherheitsmechanismen von DCOM .....	63
3.6 ActiveX-Controls .....	63
3.7 Datensicherheit in Netzwerken .....	65
3.7.1 Übersicht zur Datensicherheit .....	65
3.7.2 Umfang des Vortex-Inet-Sicherheitskonzepts .....	66
3.7.3 Authentikation .....	66
3.7.3.1 Authentikation durch Login .....	66
3.7.3.2 Authentikation durch Digitalzertifikate .....	67
3.7.4 Autorisierung .....	69
3.7.4.1 Autorisierung durch Dateiattribute .....	69
3.7.4.2 Dynamische Autorisierung .....	69
3.7.5 Verschlüsselung .....	70
3.7.5.1 Symmetrische Verschlüsselung .....	70
3.7.5.2 Asymmetrische Verschlüsselung .....	70
3.7.5.3 Verschlüsselungsstandards und -produkte .....	70
3.7.6 Datensicherheit im System Vortex Inet .....	71

4 Modellierung eines neuen Systems.....	72
4.1 Am Anfang stand eine Idee .....	72
4.2 Anforderungen an das neue System .....	72
4.3 Vorbemerkung zur Funktionalität des Prototypen .....	73
4.4 Die Funktionalität des Vortex-Inet-Prototypen .....	76
4.4.1 Die Funktionalität des Navigators.....	76
4.4.2 Die Funktionalität von Dialogen.....	77
4.4.3 Die Funktionalität von Transaktionen .....	79
4.4.4 Die Vortex-Inet-Skriptsprache .....	79
4.4.5 Ein grafisches Entwicklungstool.....	85
4.5 Auswahl zur Implementierung verwendeter Technologien.....	86
4.5.1 Festlegung der Auswahlkriterien .....	86
4.5.2 Technologien für den Vortex-Inet-Client.....	87
4.5.3 Technologien für den Vortex-Inet-Server .....	90
4.5.4 Zusammenfassung.....	93
4.6 Die logische Struktur von Vortex Inet.....	93
4.7 Der Softwareentwurf .....	96
4.7.1 Die Systemstruktur.....	96
4.7.2 Die Gruppe des Interface VortexParser.....	98
4.7.3 Die Gruppe des Interface VortexVar.....	101
4.7.3.1 Die Klasse VortexNameSpace.....	102
4.7.4 Die Gruppe des Interface VortexInstruction .....	106
4.7.5 Die Gruppe des Interface VortexComponent.....	108
4.7.6 Die Gruppe des Interface VortexContainer.....	111
4.7.6.1 Die Klasse VortexNode .....	113
4.7.6.2 Die Klasse VortexDataSheet .....	114
4.7.6.3 Die Klasse VortexDialog.....	114
4.7.7 Customizer nach dem JavaBeans-Modell.....	115
4.7.8 Die Exceptionklassen.....	116
4.7.9 Die Serverklassen.....	117
5 Implementierung des Prototypen.....	119
5.1 Vorbemerkung zur Implementierung .....	119
5.2 Vorgehensweise der Parser.....	119
5.3 Systemerweiterungen .....	121
5.4 Probleme bei der Implementierung von DCOM .....	123
5.5 Decodierung von Transaktionsergebnissen.....	124
5.6 Die Verwendung von Heuristiken.....	125
5.6.1 Hintergrund zur Verwendung von Heuristiken.....	125
5.6.2 Heuristiken zur Interpretation von Benutzereingaben .....	125
5.6.3 Heuristiken für die Deutung von Datenfragmenten .....	126
5.7 Qualitätssicherung.....	128
5.7.1 Tests .....	128
5.7.2 Konfigurationsmanagement .....	129
5.7.3 Formelle Konventionen .....	129

5.7.4 Überwachung der Implementierungsphase .....	130
5.7.5 Coderevision .....	130
6 Disziplinierte Softwareentwicklung .....	132
6.1 Ziele disziplinierter Softwareentwicklung .....	132
6.2 Das Capability Maturity Model (CMM) .....	133
6.2.1 Übersicht zum Capability Maturity Model .....	133
6.2.2 Die fünf Reifenniveaus des CMM .....	134
6.2.2.1 Der Initial Level .....	134
6.2.2.2 Der Repeatable Level .....	134
6.2.2.3 Der Defined Level .....	134
6.2.2.4 Der Managed Level .....	135
6.2.2.5 Der Optimizing Level .....	135
6.2.3 Verbesserung des Softwareprozesses mit CMM .....	135
6.3 Die Digital Program Methodology (DPM) .....	136
6.3.1 Der Lebenszyklus nach DPM .....	136
6.3.2 Das Projektmanagement nach DPM .....	137
6.3.3 Anwendung der DPM .....	138
6.4 Die Dokumentation des Projekts Vortex Inet .....	139
6.4.1 Auf DPM basierende Vortex-Inet-Dokumente .....	139
6.4.2 Zusätzliche Vortex-Inet-Dokumente .....	140
6.5 Erfahrungen mit disziplinierter Softwareentwicklung .....	140
7 Schlußbetrachtung und Ausblick .....	142
7.1 Durchführung des Projektes .....	142
7.1.1 Definition des Projektes .....	142
7.1.2 Die Umsetzung der Definition in einen Entwurf .....	143
7.1.3 Die Implementierung des Softwareentwurfes .....	144
7.1.4 Integration und Durchführung der Akzeptanztests .....	144
7.1.5 Abschluß des Projektes .....	146
7.2 Weiterführung des Projektes .....	146
Anhang .....	148
Anhang 1 .....	148
Anhang 2 .....	149
Anhang 3 .....	154
Anhang 4 .....	157
Anhang 5 .....	158
Anhang 6 .....	160
Anhang 7 .....	166
Anhang 8 .....	168
Quellenverzeichnis .....	169
Bücher .....	169
Zeitschriften .....	169
Sonstige Quellen .....	170
Ehrenwörtliche Erklärung .....	173

# Vorwort

Die vorliegende Diplomarbeit ist das Ergebnis eines Gemeinschaftsprojekts. Beide Verfasser arbeiteten während ihrer zweiten praktischen Studiensemester in der Firma Sonda S. A., Santiago de Chile, und im Anschluß an die dort verbrachte Tätigkeit wurde ihnen die Bearbeitung dieses Projekts als Diplomarbeitsthema angeboten.

Grundidee des Projekts war die Verbesserung eines sich bereits im Einsatz befindenden Softwarepakets für Kreditinstitute. Es sollte flexibel konstruiert werden, um es somit problemlos an die sich ständig ändernden Anforderungen anpassen zu können. Um den neuesten Tendenzen der Computerwelt gerecht zu werden, bestand außerdem der naheliegende Wunsch, das Softwarepaket für den Einsatz im Internet mit Hilfe der gängigen Zugangsweisen nutzbar zu machen. Ein Generationswechsel vom herkömmlichen Client-Server-Modell hin zum neuen Modell der Browser-technologie sollte entsprechend verwirklicht werden.

Die zur Verwirklichung dieses Projekts notwendige Untersuchung der neuesten Technologien sowie der Einsatz einer objektorientierten Gestaltung der neu zu entwickelnden Software gaben Anlaß dazu, dieses Projekt als Thema einer Diplomarbeit zu vertiefen. Obwohl diese Arbeit als Ganzes zu betrachten ist, so besteht sie dennoch aus aufeinander abgestimmten Teilbeiträgen. Wer der jeweilige Verfasser eines Abschnitts ist, läßt sich dem Anhang 1 entnehmen.

Bedingt durch die lokale Verteilung der vorliegenden Diplomarbeit auf zwei fern voneinander liegende Regionen - Europa und Südamerika - gestalteten sich Planung, Vorbereitung, Koordination und Durchführung des Projekts nicht immer leicht, weswegen wir uns insbesondere bei unseren beiden Betreuern, Prof. Dr. Oliver Bittel in Konstanz und Hermann von Borries in Santiago, bedanken, ohne deren Unterstützung das beschriebene Projekt in dieser Form nicht zustande gekommen wäre.

Einerseits galt es, dem wissenschaftlichen Inhalt der Arbeit sowie den Vorstellungen der Entwicklungsabteilung, in der das Projekt durchgeführt wurde, gerecht zu werden, andererseits mußte den wirtschaftlichen Interessen der auftraggebenden Abteilung sowie den Vorstellungen eines potentiellen Kunden entsprochen werden. So gilt unser Dank auch all denjenigen, die dazu beigetragen haben, eine für alle Beteiligten akzeptable Lösung durchzusetzen, und natürlich auch unseren Arbeitskollegen und Freunden, die uns bei technischen Schwierigkeiten und Sprachproblemen Unterstützung boten.

Besonderen Dank möchte ich meiner lieben Frau Ximena zukommen lassen, die während der arbeitsreichen Monate stets an meiner Seite gestanden hat und mich moralisch und liebevoll unterstützt hat. Auch meinen Eltern möchte ich bei dieser Gelegenheit herzlichst für ihre Unterstützung, während des gesamten Studiums danken. (Rainer Gross)

Besonderer Dank für die gewährte Hilfe gilt meinem Vater sowie meiner Freundin Silvia für die entgegengebrachte Geduld und Unterstützung während der anstrengenden, schwierigen und arbeitsintensiven Monate der Projektarbeit. (Christian Schloz)

Santiago de Chile, im Juni 1998

Christian Schloz  
Rainer Gross

# Abkürzungsverzeichnis

API	=	Application Programming Interface
ASCII	=	American Standard Code for Information Interchange
ASP	=	Active Server Pages
AWT	=	Abstract Window Toolkit
CGI	=	Common Gateway Interface
CLSID	=	Class Identifier
CMM	=	Capability Maturity Model for Software
COM	=	Component Object Model
CORBA	=	Common Object Request Broker Architecture
DCE	=	Distributed Computing Environment
DCOM	=	Distributed Component Object Model
DEC	=	Digital Equipment Corporation
DES	=	Data Encryption Standard
DII	=	Dynamic Invocation Interface
DLL	=	Dynamic-Link Library
DPM	=	Digital Program Methodology
DSI	=	Dynamic Skeleton Interface
engl.	=	englisch
FTP	=	File Transfer Protocol
GDT	=	Gerencia de Desarrollo Tecnológico, [span.] Forschungs- und Entwicklungsabteilung
GUID	=	Globally Unique Identifier
HTML	=	Hypertext Markup Language
HTTP	=	Hypertext Transfer Protocol
HTTPS	=	secure HTTP
IDEA	=	International Data Encryption Algorithm
IDL	=	Interface Definition Language
IETF	=	Internet Engineering Task Force
IID	=	Interface Identifier
IIOB	=	Internet Interoperable ORB Protocol
IP	=	Internet Protocol
IPC	=	Inter-Process-Communication
JDK	=	Java Development Kit
JIT	=	Just-in-Time-Compiler
LRPC	=	Lightweight RPC
MIME	=	Multipurpose Internet Mail Extensions
MS	=	Microsoft
NC	=	Netzwerkcomputer
NDR	=	Network Data Representation
OBUS	=	Object Bus

ODBC	=	Open Data Base Connectivity
OLE	=	Object Linking and Embedding
OMG	=	Object Management Group
ORB	=	Object Request Broker
OSI	=	Open Systems Interconnect
PGP	=	Pretty Good Privacy
RAM	=	Responsibility Assignment Matrix
RFC	=	Request for Comment
RMI	=	Remote Method Invocation
RPC	=	Remote Procedure Call
RSA	=	River, Shamir, Adleman
SMTP	=	Simple Mail Transfer Protocol
span.	=	spanisch
SSL	=	Secure Sockets Layer
SSLeay	=	SSL von Eric A. Young
STF	=	Sistema de Transacciones Financieras, [span.] System für Finanztransaktionen
TCP	=	Transmission Control Protocol
UDP	=	User Datagram Protocol
URL	=	Uniform Resource Locator
VMS	=	Virtual Memory System
VSL	=	VorteX Inet Scripting Language
VTBL	=	Virtual Function Pointer Table
WBS	=	Work Breakdown Structure
WWW	=	World Wide Web



# Darstellungsverzeichnis

## Abbildungen

Abbildung 1: Modulschema von Vórtex .....	6
Abbildung 2: Ereignisquelle und Ereignislauscher, verbunden durch ein Hauptprogramm .....	38
Abbildung 3: Objektadapter als spezialisierte Klassen .....	40
Abbildung 4: Attributänderung eines Bound Property .....	43
Abbildung 5: Attributänderung mit Einwand .....	44
Abbildung 6: Wichtige ORB-Komponenten und ihre Wechselwirkung, Quelle: [Keahey] .....	51
Abbildung 7: Übertragungszeiten verschiedener CORBA-Systeme .....	55
Abbildung 8: Preise verschiedener CORBA-Systeme .....	55
Abbildung 9: Aufruf von COM-Objekten, Quelle: [Schwarz] .....	59
Abbildung 10: Möglichkeiten zur Wiederverwendung von COM-Objekten, Quelle: [Schwarz] ..	61
Abbildung 11: Ein Digitalzertifikat in der Browseransicht .....	68
Abbildung 12: Vortex-Inet-Navigator .....	77
Abbildung 13: Vortex-Inet-Dialog .....	79
Abbildung 14: Vortex-Inet-Knotencustomizer .....	85
Abbildung 15: Komponenten des Systems Vortex Inet .....	94
Abbildung 16: Das Vortex-Inet-Softwareentwurfsskelett .....	97
Abbildung 17: Vortex-Inet-Systemstruktur .....	98
Abbildung 18: Konfiguration eines VortexParser-Objektes .....	99
Abbildung 19: Auslesen der Konfiguration eines VortexParser-Objektes .....	100
Abbildung 20: Struktur eines Namespace .....	105
Abbildung 21: Aufbau einer Komponente .....	110
Abbildung 22: Allgemeiner Aufbau eines VortexContainer .....	113
Abbildung 23: Arbeitsweise eines Customizer .....	115
Abbildung 24: Klassensystem der Serverklassen .....	117
Abbildung 25: Klassensystem der Serverklassen mit Visual-Basic-Server .....	123
Abbildung A-1: Klassenhierarchie .....	167
Abbildung A-2: Momentaufnahme vom Akzeptanztest .....	168

# Tabellen

Tabelle 1: Kostenvergleich zwischen PC, NC, NetPC. Quelle: Gartner Group, [Doyle] .....	14
Tabelle 2: Übersicht Clienttechnologien .....	90
Tabelle 3: Übersicht Servertechnologien.....	93
Tabelle 4: Übersicht der Variablenklassen .....	105
Tabelle 5: Übersicht der Instruktionsklassen.....	108
Tabelle 6: Übersicht der Komponentenklassen .....	111
Tabelle 7: Die von Serverapplikationen gelieferten unterschiedlichen Datumsformate .....	126
Tabelle A-1: NC vs. PC, Originaltabelle .....	157

# 1 Einleitung

## 1.1 Die Diplomarbeitsfirma

Die vorliegende Diplomarbeit basiert auf einem Projekt, welches in der Firma Sonda S. A. in Santiago de Chile durchgeführt wurde. Sonda ist eine chilenische Aktiengesellschaft, deren Anteile zu 60% von der Familie des Firmengründers Andrés Navarro und zu 40% von der US-amerikanischen Digital Equipment Corporation gehalten werden, welche während der Entstehungszeit dieser Diplomarbeit von der Firma Compaq übernommen wurde. Im Jahr 1974 als Kleinbetrieb gegründet, hat Sonda ein enormes Wachstum erlebt und ist heute in ganz Lateinamerika aktiv. Hauptsitz der Firma ist die chilenische Hauptstadt Santiago. Von den insgesamt ungefähr 1800 Mitarbeitern sind etwa 1300 in Chile beschäftigt.

Sonda bietet Hard- und Software für Großkunden an. Zu den Kundenkreisen gehören unter anderem Banken und Produktionsbetriebe. Das Sondasche Serviceangebot erstreckt sich vom Rund-um-die-Uhr-Bereitschaftsdienst für technische Probleme, beispielsweise in Krankenhäusern, bis hin zu einer breiten Palette von Kursen und Seminaren über die verschiedensten Arten von Software, darunter sowohl Unix, Novell und Microsoft, als auch Sonda-eigene Entwicklungen. In verschiedenen Ländern Lateinamerikas ist Sonda offizieller Vertreiber von Produkten der Digital Equipment Corporation, dieses bezieht sich sowohl auf Hardware, beispielsweise Digital PCs, als auch auf Software, vor allem Betriebssysteme wie VMS oder DEC Unix.

Bezüglich der von Sonda entwickelten Software seien an dieser Stelle Betriebswirtschaftssoftware, Software für Banken, Rentenversicherungen und zur Prozeßsteuerung genannt. Ein großes Softwareprojekt Sondas ist beispielsweise die Ampelsteuerung in der brasilianischen 15-Millionen-Metropole São Paulo. Das dieser Arbeit zugrunde liegende Projekt wurde von der *División Sistemas Bancarios* (Sonda Bancos), zuständig für Entwicklung und Vermarktung von Bankensoftware, beauftragt und innerhalb der *Gerencia de Desarrollo Tecnológico* (GDT), einer Forschungs- und Entwicklungsabteilung für Software, durchgeführt.

## 1.2 Das Projekt

Zusammen mit *Banco Santiago*, der heute größten chilenischen Bank, hatte Sonda Bancos vor einiger Zeit ein Softwaresystem namens Vórtex entwickelt, das dazu dient, die verschiedensten in einem Kreditinstitut eingesetzten Anwendungsprogramme unter einer einheitlichen Benutzeroberfläche zusammenzufassen. Gesteuert von einer eigens zu diesem Zweck entwickelten einfachen Skriptsprache, ermöglicht Vórtex die Zusammenstellung von hierarchisch aufgebauten Menüoberflächen, mit deren Hilfe ein Bankmitarbeiter die jeweils gewünschte Anwendung bedienen kann. Die Dateneingabe geschieht mittels Dialogfenstern, die Kommunikation mit zen-

tralen Datenbanken wird von einem Transaktionsserver übernommen, und die Darstellung der Ausgabedaten erfolgt anhand von Ausgabefenstern, Tabellen oder weiteren Menüs.

Das System Vórtex wird bereits von Banco Santiago erfolgreich eingesetzt. Es unterstützt die 16-Bit-Architektur der Windows-Version 3.1 und wird in verschiedenen Etappen dem aktuellen Windows-Standard angepaßt werden. Davon unabhängig, sollte in einem parallelen Projekt das Gebiet der verschiedenartigen Internettechnologien erforscht und deren Einsatzmöglichkeiten bezüglich des Systems Vórtex untersucht werden. So entstand die Idee von Vortex Inet, einer Vórtex-Version, die mit Hilfe der gängigen Internet-Browser zu bedienen sei. Vortex Inet sollte den Einsatz von Netzwerkcomputern - auf die Fähigkeit, Anwendungen aus dem Internet auszuführen, reduzierte Workstations - ermöglichen, um somit auch diesen Zweig der Entwicklung zu verfolgen.

Das Projekt Vortex Inet bildet die Grundlage der vorliegenden Diplomarbeit. Es war als firmeninterne Auftragsarbeit geplant und wurde, beauftragt von Sonda Bancos, innerhalb der Entwicklungsabteilung GDT realisiert. Obwohl für die Vollendung des Projekts kein unmittelbarer Termindruck herrschte, wie es bei externen Aufträgen häufig der Fall ist, so war der dafür vorhandene Etat dennoch beschränkt, weshalb die Entwicklung innerhalb eines festgelegten Zeitraums vollendet werden mußte. Davon abgesehen, darf ein großer Softwareproduzent natürlich nicht den Anschluß an neue Tendenzen und Entwicklungen verlieren, weshalb der Wunsch bestand, das Projekt unverzüglich in Angriff zu nehmen. Trotzdem ermöglichte der Rahmen dieses Projekts ausreichenden Spielraum, um einen weiteren Trend in der Softwareentwicklung zu verfolgen: die objektorientierte Programmierung. Um aus dem bestehenden Vórtex-System ein objektorientiertes Softwaretool Vortex Inet hervorbringen zu können, mußte das System von Grund auf neu gestaltet werden. Dieser Aufwand erscheint allerdings weniger gravierend, wenn man bedenkt, daß die Ausrichtung auf den Einsatz von Netzwerkcomputern ohnehin schon grundlegende Änderungen in der Systemarchitektur mit sich bringen würde, und daher eine schlichte Portierung nicht möglich gewesen wäre.

Zusätzlich zu den beschriebenen Motivationen, die die Abteilung Sonda Bancos zur Beauftragung des Projekts Vortex Inet veranlaßt hatten, wurde von der Entwicklungsabteilung GDT, in der das Projekt verwirklicht wurde, noch eine weitere interessante Komponente beige-steuert. Als Auftraggeber des Projekts forderte Sonda Bancos fortlaufende Informationen in Form von schriftlichen Dokumenten über den jeweils aktuellen Entwicklungsstand. Gleichzeitig bestand bei GDT das Interesse, Softwareentwicklungen gezielter planen und exakter dokumentieren zu können. So wurde vereinbart, das Projekt Vortex Inet basierend auf der *Digital Program Methodology* (DPM) zu entwickeln und dokumentieren. Vortex Inet sollte gleichzeitig als Pilotprojekt für den Einsatz einer disziplinierten Softwareentwicklungsstrategie dienen.

Für den Auftraggeber Sonda Bancos stand von Anfang an nicht in erster Linie die Erforschung neuer Techniken oder Entwicklungsstrategien im Vordergrund, sondern vielmehr die Möglichkeit, ein für die derzeitigen Gegebenheiten und Anforderungen geschaffenes Produkt verkaufen zu können. Unter dieser Voraussetzung und dem Wissen, mit Banco Santiago bereits über einen potentiellen Kunden für das zu schaffende System Vortex Inet zu verfügen, ist es verständlich, daß die Kompatibilität mit dem vorhanden System beziehungsweise mit den vom Kunden bereits programmierten und eingesetzten Skriptprogrammen gewünscht wurde. Diese Kompatibilität