

Profil

Dipl.Ing. Roger René Kommer

Anschrift:

Ingenieurbüro Kommer
Dörnbergstrasse 11
34119 Kassel
Tel.: +49 561-780-150
Fax.: +49 561-780-135
<http://www.artefaktur.com>
<mailto:kommer@artefaktur.com>

Unterkapitel:

Lebenslauf

Lebenslauf von Roger Rene Kommer

Skills

Meine beruflichen Schwerpunkte liegen in Design und Implementierung von generellen und branchenspezifischen Frameworks sowie in der Projekt- und Entwicklungsleitung von Softwareprojekten. Weitere fachliche Schwerpunkte liegen in plattformunabhängigen C++ Bibliotheksdesign, Enterprise Applications Integration und Middlewareintegration. Tiefere Branchenkenntnisse bestehen in den Bereich der Logistik, Investmentbanking und Architektur.

Projektphilosophie

Auf Grundlage meiner Erfahrungen mit - auch kritischen - Projekten bevorzuge ich Projekte nach den Grundsätzen der agilen Softwareentwicklung und eXtrem Programming, gekoppelt an klaren und offenen formalen Richtlinien.


Projekte














Projekte der Vergangenheit.

1 Lebenslauf

Lebenslauf von Roger Rene Kommer

Name: Roger René Kommer
 Geburtsdatum: 20.9.1966
 Geburtsort: Zürich (Schweiz)
 Staatsangehörigkeit: Schweiz
 Familienstand: ledig
 Wohnort:
 Dörnbergstraße 11
 34119 Kassel
 Deutschland

Tel.: 0561 / 780 150
 Fax.: 0561 / 780 135
 Email:  <mailto:kommer@artefaktur.com>

Beruflicher Werdegang:	2003/2004	Berater für Softwareentwicklung, Logistik und Simulation für ein mittelständisches Lagerlogistikunternehmen.  Lagerlogistik . Berater für Softwareentwicklung für ein grosses Unternehmen aus dem Bereich der Transportlogistik.  Stabilisierung Logistik .
	2003 - 2004	Softwareentwickler und Projektleiter des OpenSource Projektes ACDK.  acdk.sourceforge.net
	2001 - 2002	Berater für Softwareentwicklung für die Commerzbank AG Frankfurt. Entwicklung eines Marktdatensystems.  Projekt MDDS .
	2000	Geschäftsführer der Micromata_objects GmbH. Entwicklung eines C++-Frameworks und Applikationsservers:  ACDK Entwicklungsleitung einer verteilten B2B-Lösung:  Atrax Projektleitung Konsolidierung E-Commerceanwendung:  FriendScout24 Projektleitung und Entwicklung eines generischen SAP-RFC Connectors:  AILA .
	1999	Projekt- und Entwicklungsleiter bei der Entwicklung einer generischen Simulationsumgebung / Prozessleitsystems.  LMF-Framework
	1997-1998	Projekt- und Entwicklungsleiter bei der Entwicklung eines Materialflusssystemes für die Fresenius AG.  Fresenius
	1995-1997	Softwareentwickler, Berater, und Lizenzgeber für den US-Konzern Computer Associates International.  PowerUser
	1993-1995	Unabhängiger Softwareentwickler und international tätiger Unternehmer mit Vertrieb von eigener Software.  PowerUser
	1991-1999	Softwareentwickler einer Datenbankanwendung und Administrator einer NT-Domäne und Internetdiensten.  ISP

		1993-1995: Dozent in CAD (AutoCAD) für Architekturstudenten.
Hochschule:	1987-93	<p>Studium Universität Kassel Architektur</p> <p>9 Monate Praktika: Bauarchäologisches Institut in Marburg. Wissenschaftliche Arbeit über Schablonenmalereien von 1500-1990</p> <p>9 Monate Praktika: Architekturbüro, AVA, Genehmigungs- und Detailplanung für die Dokumentahalle in Kassel.</p> <p>Abschluss: Diplom-Ingenieur Architektur Note: Sehr gut</p>
Schulbildung:	1987	<p>Technisches Gymnasium Schwenningen - Baden Württemberg</p> <p>Abschluss: Allgemeine Hochschulreife NC: 1,8</p>

2 Skills

Meine beruflichen Schwerpunkte liegen in Design und Implementierung von generellen und branchenspezifischen Frameworks sowie in der Projekt- und Entwicklungsleitung von Softwareprojekten.

Weitere fachliche Schwerpunkte liegen in plattformunabhängigen C++ Bibliotheksdesign, Enterprise Applications Integration und Middlewareintegration.

Tiefere Branchenkenntnisse bestehen in den Bereich der Logistik, Investmentbanking und Architektur.

Meine Schwerpunkte finden sich in folgenden Projekten wieder:

- Design und Entwicklung von generellen und branchenspezifischen Frameworks:
 - [▶ AC DK](#)
 - [▶ Lagerlogistik](#)
 - [▶ Projekt MDDS](#)
 - [▶ LMF-Framework](#)
 - [▶ PowerUser](#)
- Projekt- und Entwicklungsleitung:
 - [▶ AC DK](#),
 - [▶ Atrax](#),
 - [▶ AILA](#),
 - [▶ LMF-Framework](#),
 - [▶ Fresenius](#),
 - [▶ PowerUser](#),
- Enterprise Applications Integration im Zusammenhang von SAP R/3 und anderen Fremdsystemen:
 - [▶ Lagerlogistik](#),
 - [▶ ITalkIDOC](#),
 - [▶ AILA](#),
 - [▶ Fresenius](#)
- Softwareengineering mit Bibliotheksdesign, Interpreterbau und Middlewareintegration:
 - [▶ AC DK](#),
 - [▶ Lagerlogistik](#),
 - [▶ Projekt MDDS](#),
 - [▶ FriendScout24](#),
 - [▶ AILA](#),
 - [▶ ITalkIDOC](#),
 - [▶ LMF-Framework](#),
 - [▶ Fresenius](#),
 - [▶ PowerUser](#),
 - [▶ BSH](#),
 - [▶ WIOSERV](#),
 - [▶ SBSI](#),

Im Detail:

Inhalt dieses Kapitels:

- [▶ IT](#)
- [▶ Methoden](#)

- ☒ Betriebssysteme
- ☒ Programmierung
 - ☒ Sprachen
 - ☒ Entwicklung von Softwaresprachen
 - ☒ Middleware
 - ☒ Implementierte Protokolle
 - ☒ Entwicklungsumgebungen
 - ☒ Make / CASE / Versionskontrolle
 - ☒ Softwarebibliotheken
 - ☒ Compiler-Werkzeuge
- ☒ Internet Server
 - ☒ Datenbanken
 - ☒ SAP R/3
- ☒ Desktop
 - ☒ Office
 - ☒ CAD und GIS
- ☒ Management
 - ☒ Leiterische Tätigkeiten
 - ☒ Allgemeine Managementtätigkeiten
 - ☒ Projektleitung
 - ☒ Produkte
 - ☒ Produkt Design
 - ☒ Vertrieb
- ☒ Branchen
 - ☒ Investment Banking
 - ☒ Logistik, Fördertechnik, Materialfluss und Warenwirtschaft
 - ☒ Architektur
- ☒ Verschiedenes
 - ☒ Sprachen
 - ☒ Führerschein
 - ☒ Arbeitserlaubnis
 - ☒ Ausserberufliche Interessen

IT

Methoden

Methoden:

- Kontrolliertes eXtrem Programming
- Agile Softwareentwicklung
- OOA, OOD, OOP, UML
- Entwicklung und Leitung nach ISO 9001
- Datenmodellierung (ERD)
- Pattern
- Aspect Oriented Programming (AOP) / Intentional Programming (IP)

Siehe auch [☒ Projektphilosophie](#).

Betriebssysteme




































Gute Erfahrungen mit

- MS DOS, Windows 3.1, Windows 95, Windows NT / 2000 / XP / 2003,
- Unix (Sun Solaris, Linux, AIX, BSD, Darwin)

Als Softwareentwickler, Administrator und Nutzer.
Erfahrungen sind bei OS/2, MacOS and OpenStep vorhanden.

Programmierung

Sprachen

- C++ & C: Experte / 'Guru', > 10 Jahre
 -  ACDK,
 -  Lagerlogistik,
 -  Projekt MDDS,
 -  LMF-Framework,
 -  Fresenius,
 -  PowerUser,
- Java: Experte, 5 Jahre
 -  ACDK,
 -  LMF-Framework,
 -  Atrax
- Perl: Experte, 6 Jahre
 -  Projekt MDDS,
 -  FriendScout24,
 -  ACDK,
 -  GenWeb
- Basic dialects (VisualBasic 1.0 – 5.0, Wordbasic, AccessBasic, etc):
Projekterfahren, 10 Jahre
 -  Projekt MDDS,
 -  ISP,
 -  SBSI,
 -  PowerUser
- Tcl/Tk: Projekterfahren, 2 Jahre
 -  ACDK,
 -  Fresenius
- PHP (PHP3/4): Projekterfahren, 3 Jahre
 -  Projekt MDDS.
 -  ProjectForge.
- LISP (ACDKLisp, AutoCAD, Common Lisp, elisp):
Projekterfahren, 4 Jahre
 -  ACDK
- SQL (ESQLC, Informix, Oracle, Sybase, ODBC, Access):
Projekterfahren / Experte, 5 Jahre
 -  Lagerlogistik,
 -  Projekt MDDS,
 -  FriendScout24,
 -  Fresenius,
 -  ISP
- Unix-shell Programmierung (sh, ksh, csh, awk, sed, etc.) Projekterfahren, 4 Jahre
 -  Projekt MDDS.
 -  ProjectForge.
- HTML, CGI in Java, Perl, VBScript, JScript:
Projekterfahren, 6 Jahre
 -  Projekt MDDS,
 -  GenWeb,
 -  ProjectForge
- UML Projekterfahren, 5 Jahre
 -  Projekt MDDS,
 -  ACDK,
 -  LMF-Framework,

- [▶ Fresenius,](#)
- [▶ PowerUser,](#)
- [▶ Atrax](#)

Entwicklung von Softwaresprachen

Experte / 'Guru': 8 Jahre

- C++/Java/Middleware: [▶ ACDK](#)
- Java Dialekt: [▶ ACDK](#)
- Lisp Dialekt: [▶ ACDK](#)
- Basic Dialekt: [▶ SBSI](#)
- Install script: [▶ InstScript](#)
- Shell: [▶ BSH](#)

Middleware

- CORBA (IDL):
Projekterfahren / Experte, 5 Jahre
 - [▶ ACDK,](#)
 - [▶ LMF-Framework,](#)
 - [▶ Fresenius.](#)
- [D]COM:
Projekterfahren / Experte, 4 Jahre
 - [▶ ACDK,](#)
 - [▶ Projekt MDDS,](#)
 - [▶ PowerUser,](#)
 - [▶ BSH,](#)
 - [▶ WIOSERV,](#)
 - [▶ SBSI](#)
- XML-RPC/SOAP:
Projekterfahren, 1 Jahr
 - [▶ ACDK,](#)
 - [▶ Projekt MDDS,](#)
 - [▶ Atrax](#)

Implementierte Protokolle

Folgende Protokolle habe ich selbst implementiert:

- CORBA IIOP, GIOP, POA, DSI
- Sockets (TCP/IP / UDP), HTTP (Server und Client), FTP (Client)
- XML SAX und DOM
- RPC-XML (Server und Client)
- SAP RPC, IDOC (Client)
- COM, DCOM (Client und Server)
- Java JNI (Client), Serialisation (Client/Server), .class
- ODBC (Client)
- Siemens S5/S7
- Profibus
- Perl, Python und Tcl C-Interface

Entwicklungsumgebungen

- VC++ 1.0 – Visual Studio .NET 2003,
- gcc/g++
- Sun Workshop C++,
- Borland C++ 3.0 – X,
- Visual J++, Emacs, Watcom C++ 10.0,
- Sniff+

Make / CASE / Versionskontrolle

- [g]n[un]make (Unix/GNU),
- Rational ClearCase,
- Rational Rose,
- Rational ClearQuest,
- RCS/CVS,
- Visual SourceSafe,
- ObjectDomain,
- Together++

Softwarebibliotheken

- ACDK (design & development),
- STL (HP, SG, ObjectSpace),
- RogueWave (Tools, Threads, DBTools),
- Borland Template Library,
- MFC,
- (D)COM/ OLE2/ActiveX,
- ATL,
- CORBA/ORB (HP, OmniOrb, Mico),
- ODBC,
- wxWidgets/wxWindows
- GTK+
- Qt
- X11/Xt,
- Motif,
- OWL,
- ACE,
- YACL
- PORTWCL (GUI-Bibliothek für Windows Programming (Windows 3.1 (16bit), Win32s, Windows 95, NT 3.51, NT 4.0) for Borland C++ 4 - 5, MSC 1 - 5, Watcom 9 - 11) und OS/2.

Compiler-Werkzeuge

- Lex (Flex, Flex++) and
- Yacc (Bison, byacc, Bison++).
- Diverse Eigenentwicklungen

Internet Server

- Apache: Projekterfahren, 1 Jahr
 -  [ProjectForge](#)
 -  [FriendScout24](#)

- MS IIS 1 - 4 (HTTP, FTP, telnetd), inklusive Frontpage und CGI (Java Servlets, C, Perl, ASP): Projekterfahrungen, 2 Jahre [ISP](#)
- Email Server, MS SQL Server 6.5, MS Transactionserver,
- Firewall

Datenbanken

Projekterfahrungen / Experte: 5 Jahre

- Eigenentwicklung einer JDBC-Like Database Bridge: [ACDK](#)
- Oracle unter Linux, Solaris und Windows:
 - [Projekt MDDS](#),
 - [ACDK](#),
 - [FriendScout24](#),
 - [Atrax](#)
- Sybase unter Solaris: [Projekt MDDS](#)
- Informix unter Solaris: [Fresenius](#)
- MS SQL:
 - [Lagerlogistik](#),
 - [AILA](#)
- MySQL unter Linux:
 - [ACDK](#),
 - [ProjectForge](#)
- MS Access unter Windows [ISP](#)
- Ingres SQL unter AIX [ISP](#)

SAP R/3

Integration von SAP R/3 Systemen mit anderen Anwendungen:
Projekterfahrungen, 5 Jahre

- [Lagerlogistik](#)
- [ACDK](#)
- [ITalkIDOC](#)
- [AILA](#)
- [Fresenius](#)

Desktop

Office

Erfahrungen mit Anwendung und Programmierung:

- MS Word 5 - 6,
- Winword 1 - XP,
- MS Access 1 - XP:
 - [Projekt MDDS](#)
 - [ISP](#)
- MS Excel 1 - XP: [Projekt MDDS](#).

CAD und GIS

Projekterfahrungen, 2 Jahre

- AutoCAD (10 – 2002) unter SunOS/Solaris, MS DOS und Windows NT.
- ArcView GIS V3.
 - [ArcView](#).

Management

Leiterische Tätigkeiten

Allgemeine Managementtätigkeiten

- Führung eines Ingenieurbüros: 12 Jahre
- Geschäftsführung mit 4 Angestellten: 1 Jahr
- Projektentwicklung mit Kunden: 9 Jahre
- Leitung eines Softwareentwicklerteams: 7 Jahre
- Softwareentwicklung von A bis Z: 10 Jahre
- Projektplanung und Kontrolle: 7 Jahre
- Ausschreibung, Angebotsstellung: 6 Jahre
- Pflichtenheft, Spezifikation, Abnahme: 5 Jahre
- Etablierung von QS-Massnahmen: 4 Jahre
- Aufsetzen von Entwicklungsumgebungen: 5 Jahre
- Redesign, Refactoring: 5 Jahre
- Coaching von Entwicklungsteams: 4 Jahre
- Planen und Durchführen von Notfallmassnahmen: 4 Jahre
- Dokumentation (Anwender, System): 7 Jahre
- Inbetriebnahme, Support: 6 Jahre
- Vermarktung von Softwareprodukten: 4 Jahre
- Erstellen eines Businessplans: 1 Jahr
- Verhandlungen mit Auftraggebern, Lizenznehmern und Kapitalgeber: 3 Jahre

Projektleitung







- Projektleitung bei Projekten zwischen einem Mann-Monat und 16 Mann-Jahren.
- Leitung von Projektgruppen bis zu 12 Mitarbeitern.
- Projektleitung und Entwicklungsleitung nach ISO 9000/9001.

Projekte:






- [ACDK](#)
- [Lagerlogistik](#)
- [FriendScout24](#)
- [ProjectForge](#)
- [Atrax](#)
- [AILA](#)
- [LMF-Framework](#)
- [Fresenius](#)
- [PowerUser](#)
- [ISP](#)

Produkte

Produkt Design

-  [ACDK](#),
-  [ProjectForge](#),
-  [ITalkIDOC](#),
-  [LMF-Framework](#),
-  [PowerUser](#),
-  [Win4Help](#)


Vertrieb

-  [ACDK](#),
-  [ITalkIDOC](#),
-  [ProjectForge](#),
-  [PowerUser](#),
-  [Win4Help](#)

Branchen

Investment Banking

2 Jahre Projekterfahrung.

-  [Projekt MDDS](#)

Logistik, Fördertechnik, Materialfluss und Warenwirtschaft

5 Jahre Projekterfahrung, Experte.

-  [Lagerlogistik](#)
-  [Stabilisierung Logistik](#)
-  [AILA](#)
-  [LMF-Framework](#)
-  [Fresenius](#)

Architektur

1 Jahr Projekterfahrung.

- AVA (Ausschreibung, Vergabe, Abrechnung)
- Bauleitung
- Werk- und Detailplanung
- Entwurf

Verschiedenes

Sprachen

- Deutsch: Muttersprache.
- English: Fließend in Wort und Sprache.
Englisch war in mehreren Projekten Kommunikations- und Dokumentationsprache.
- Französisch: Grundkenntnisse.


Führerschein

Führerschein der Klassen 1 und 3.

Arbeitserlaubnis

- Deutschland (Arbeitnehmer und Selbständig)
- Schweiz, da Schweizer

Ausserberufliche Interessen

- Opus Magnum:  acdk.sourceforge.net mit ca 1,200,000 Zeilen Code und 3.000 Klassen.
- Klassische Musik
- Architektur/Kunst
- Klassische europäische Philosophie
- Fahrradfahren

3 Projektphilosophie

Auf Grundlage meiner Erfahrungen mit - auch kritischen - Projekten bevorzuge ich Projekte nach den Grundsätzen der agilen Softwareentwicklung und eXtrem Programming, gekoppelt an klaren und offenen formalen Richtlinien.

Einer der wichtigsten Begriffe bei der Softwareentwicklung und Projekt- und Entwicklungsleitung ist der der Angemessenheit.

Es gibt kein 'wahr' und 'falsch' beim Design von Software und auch nicht bei der Auswahl von Methoden.

Einige Entscheidungen kann man nach dem Muster, soviel A wie möglich und soviel B wie nötig bewerten. Zum Beispiel: So viel Freiheit und Vertrauen in alle Beteiligten wie möglich und soviel Planung und Kontrolle wie nötig.

Die folgenden Richtlinien sind somit nicht absolut zu sehen, sondern resultieren aus meinen Erfahrungen aus früheren Projekten.

Nach Innen wirkende Richtlinien.

- Die Qualität des Softwareproduktes soll sich schon zu einem frühen Zeitpunkt in Tests, Konfigurationsmanagement, Logging, System- und Nutzerdokumentation niederschlagen.
- Als Entwicklungsleiter muss man auch inhaltlich im Zentrum der Softwareentwicklung stehen. Der Entwicklungsleiter muss sämtliche Aspekte des Softwareprozesses verstehen.
- Design und wichtige technische Details werden innerhalb der Entwicklergruppe diskutiert. Im Zweifelsfalle entscheidet der Entwicklungsleiter.
- Der Einsatz von Unittests, die in einem automatischen Regressionstest eingebunden werden können, ist unabdingbar.
- Der Einsatz einer Versionsverwaltung ist unabdingbar und sollte alle projektrelevanten Dokumente umfassen.
- Designwerkzeugen und CASE-Tools (wie etwa Rational Rose) sollen nicht die Softwareentwicklung determinieren, sondern umgekehrt. Reine UML Diagramme überdecken oft die in dem Projekt innewohnende zu lösende Probleme. Klare deutsche (oder englische) Sätze, illustriert mit UML Diagrammen, Pseudocode oder Prototypen müssen ein klares Verständnis innerhalb des Projektteams und bei dem Kunden vermitteln.
Softwareentwicklung soll in media res geschehen.
- Entwicklung möglichst aus 'produktiven' Prototypen.
- Mit den Mitarbeitern (ggf. mit den Kunden) werden Wochen- und Monatsziele vereinbart und jeweils nach Ablauf kontrolliert und protokolliert. Probleme können frühzeitig erkannt werden, unrealistische Aufwandseinschätzungen können korrigiert werden.
- Externe Schulungen können die Aus- und Weiterbildung innerhalb eines Teams nicht ersetzen, höchstens ergänzen. Von der Projekt- und Entwicklungsleitung müssen die Rahmenbedingungen geschaffen werden, damit die Weiterbildung von Anfängern und Fortgeschrittenen den Erfahrenen nicht als Last und Strafe vorkommt.
In der Vergangenheit habe ich zwischen 5 - 50% meiner Zeit in Projekten mit Weiterbildung und Coaching verbracht.
- In komplexen Projekten, in denen keine adäquaten Aufgaben gefunden werden können, haben Anfänger (Umschüler) nichts zu suchen.
- Vom Hinfallen lernt man das Laufen nicht. Einer der wichtigsten Motivation eines Entwicklers ist der Erfolg des Projektes.

Nach Außen wirkende Richtlinien.

- Bei größeren Projekte, gestufte Pflichtenhefte mit jeweiliger Beschreibung des Abnahmeprozesses.
- Dokumentation des Projektverlaufes mittels eines Projektstagebuches. Das für Entwickler und Kunden möglichst offene Projektstagebuch enthält folgende Elemente:
 - Pflichtenhefte, Spezifikationen.
 - Projektplanung und Personaleinsatzplanung.
 - Kommunikationsprotokolle mit Kunden (Jour Fixe, Telefonate).
 - Teambesprechungen mit Wochenzielen (siehe unten).
 - Testprotokolle, Offene Punkte, Fehlerliste.
 - Bei Produktionsbetreuung, wichtige Logfiles Fehlerberichte.

Email und Word-Dateien haben oft den Charakter von Write-only Dokumente. Im optimalen Fall ist das Projektstagebuch ein Intranet, an dem sich alle Projektbeteiligten einbringen können.

- Enger Kontakt zu Kunden, um Fehlentwicklungen zu vermeiden.
- Hierbei möglichst ehrlichen und offener Kommunikation mit Kunden.
- Eigentlich banal aber dennoch in der Praxis problematisch: Besprechungen innerhalb des Projektes oder mit dem Kunden in möglichst kleinem Kreis. Ergebnisse werden im Projektstagebuch festgehalten, die für alle Beteiligten Pflichtlektüre sein sollte.
- Ein großes Problem bei Client-Server Projekten, bei denen der Kunde mittels des Frontends mit dem Produkt in Kontakt kommt, ist die relative lange Zeit in der das Projektteam mit Backend und technischer Infrastruktur zu kämpfen hat und aus Sicht des Kunden das Projekt scheinbar nicht vorankommt. Mit einem Prototyp, dem Projektstagebuch mit einem transparenten Projektplan kann vor Frustration vorgebeugt werden.
- Bei mittleren (> 3 Entwickler) und großen Projekten soll die Rolle des Projektleiters (globale Steuerung des Projektes, Kommunikation mit Geschäftsführung) und Entwicklungsleitung (Leitung/Steuerung des Entwicklerteams, auch inhaltlicher Kopf der Entwicklung) nicht durch die gleiche Person abgedeckt werden. Projektleiter und Entwicklungsleiter haben in vielen Aspekten gegensätzliche Interessen zu vertreten. Wenn beide Rollen von einer Person abgedeckt werden, passiert es oft, dass eine der Rollen zum Schaden des Projektes vernachlässigt wird. Projekt- und Entwicklungsleiter müssen eine gemeinsame Strategie entwickeln und in ihrem Wirkungskreis umsetzen.

Siehe auch [ProjectForge](#).

4 Projekte

Projekte der Vergangenheit.

Unterkapitel:

- Projekt Automatisierte Lagerlogistik
 Unterstützung beim Softwareentwicklungsprozess mit Einführung von ACDK als Entwicklungsplattform. Eigenständige Realisation von Kundenprojekten im Bereich von automatisierte Lagerlogistik.
- Stabilisierung webbasierte Logistik Lösung
 Stabilisierungsmassnahmen, Entwicklung von Test- und Releasemanagement und Performancetuning für eine PHP Oracle SQL Lösung für einen Konzern im Bereich Transportlogistik.
- Projekt Market Data Distribution System (MDDS)
 Das Market Data Distribution System (MDDS) ist eine Datenbanklösung zum Sammeln, Verwalten und Verteilung von Börsendaten.
- Projekt ACDK (Maja Objects)
 ACDK (früher Maja Objects) ist ein plattformunabhängiges Framework für C++ für die Realisierung von Serveranwendungen mit einer Orientierung an Java.
- Projekt FriendScout24
 Stabilisierung und Reengineering einer E-Commerce-Plattform unter Sun, Oracle, Apache, Perl.
- Projekt ProjectForge
 Webbasiertes Projektmanagement-Werkzeug.
- Projekt GenWeb
 Autorenwerkzeug für die Erstellung von Webseiten, Handbüchern, Hilfedateien und Quelltextdokumentationen.
- Projekt Atrax
 B2B Plattform auf der Basis von ACDK, Oracle, Java, XML.
- Projekt ITalkIDOC
 SAP R/3-Koppelung für ACDK
- Projekt AILA - Constructor
 Realisierung einer SAP R/3-Koppelung Aircabin/DASA für ein Lager der Aircabin GmbH (Tochter im DASA/DC Konzern) mittels ITalkIDOC.
- Projekt LMF-Framework
 Im Auftrag der MAN Logistics GmbH entwickelte ich als Projekt- und Entwicklungsleiter ein generisches Materialflusssystem und Prozessleitsystem mit einer Simulationsumgebung.
- Projekt ArcView - Team Heese
 Für die Team Heese AG entwickelte ich eine Schnittstelle für ArcView GIS zu MS Access.
- Projekt LynxTerm
 Erweiterung eines textbasierten Webbrowsers für den Einsatz als GUI für Funkterminals im Bereich Lagerlogistik
- Projekt Fresenius
 Als Entwicklungs- und Projektleiter führte ich 1997 ein Softwareentwicklungsprojekt im Bereich

Lagerlogistik, Lagerverwaltung und SAP R/3-Anbindung durch.

Projekt PowerUser, Computer Associates

Für eine Produktpalette, die für den internationalen Massenmarkt bestimmt war, lizenzierte Computer Associates International das von mir erstellte Programm 'New Menus'. Von 1993 bis 1997 war ich für diese Firma auch als Berater tätig.

Projekt Basic Shell (BSH)

Eine einfache Scripting Shell.

Projekt Windows IO-Server

WioServ (Windows IO Server) stellt durch den OLE2 Mechanismus unter Windows 3.1 und Windows95/NT (16/32) ein Fenster zur Verfügung, in dem man textbasierte Ausgaben machen kann.

Projekt Simple BasicScript Interpreter (SBSI)

Ein Basic Interpreter mit Wordbasic kompatibler GUI-Funktionalität.

Projekt Win4Help

Integriert die erweiterte Funktionalität der Windows 95/NT4.0-Hilfe in Ihre Windows 3.1-Hilfe.

Projekt Universität Kassel, ISP

Seit 1993 unterstütze ich die Bibliotheken der Fachbereiche Architektur und Stadt- und Landschaftsplanung in EDV-Belangen von Aufbau eines Servers bis hin zu Erstellung eines Datenbanksystems für die Medien-Bibliothek

4.1 Projekt Automatisierte Lagerlogistik

Unterstützung beim Softwareentwicklungsprozess mit Einführung von ACDK als Entwicklungsplattform. Eigenständige Realisation von Kundenprojekten im Bereich von automatisierte Lagerlogistik.

Zeit:	2003/2004
Kunde:	Mittelständisches Unternehmen für Lagerlogistik
Position:	Troubleshooting, Entwicklung, Beratung, Projektrealisation
Technologien:	C++, ACDK, SQL 2000, MFC, Fördertechnik, Materialfluss, Warenwirtschaft

Inhalt dieses Kapitels:

- Entwicklerunterstützung/Entwicklung von Basiskomponenten
- Entwicklung eines automatisierten Test- und Simulationsumgebung
- Realisation eines Kundenprojektes Lagerverwaltung

Entwicklerunterstützung/Entwicklung von Basiskomponenten

- Entwicklung von C++-Basiskomponenten für den Einsatz im Bereich Steuerung von automatischer Fördertechnik, Materialfluss, Lagerverwaltung auf Grundlage von ACDK.
Darunter:
 - Basiskontainer-Klassen.

- Exceptionhandling.
- Logging
- IO
- SQL, selects, updates, stored procedures/functions
- Entwickeln von verbesserten Debugging (pdbutils).
- Troubleshooting bei diversen Kundenprojekten.

Entwicklung eines automatisierten Test- und Simulationsumgebung

Auf Grundlage einer Anlagenvisualisierungssoftware und rudimentären Simulationssoftware wurde eine automatisierte Simulations- und Testumgebung entwickelt. Diese Testumgebung hatte folgende Ziele und Eigenschaften:

- Funktionelle Abbildung aller Fördertechnikelemente, wie Förderbahnen, Regalbediengeräte, Umsetzer, Scanner, Waagen, etc. auf Protokollebene zur unterlagerten Steuerung (SPS 5, SPS 7).
- Anpassbarkeit von projekt/kundenspezifischen Fördertechnik und Schnittstellen innerhalb der Simulation. Die Testumgebung muss eine Funktionsabdeckung der kundenspezifischen Materialfluss/Lagerverwaltungssoftware von 100% erreichen.
- Projekt-/kundenspezifischen Anpassungen müssen ausserhalb der eigentlichen Simulationssoftware vorgenommen werden, ansonsten wird die Stabilität und Wartbarkeit der eigentlichen Simulations- und Testsoftware gefährdet.
- Für jeden Funktionspunkt der Software wird ein entsprechender Unittest erstellt. Diese Unittests müssen wiederholbar sein. Die Simulationstests müssen demnach automatisierbar sein.
- Die Erstellung der einzelnen Unittests muss begrenzten Aufwand und technischem Wissen erstellt werden.
- Einbindung von interaktiven Elementen, wie etwa GUI Bedienersoftware.
- Projekt-/kundenspezifischen Pflichtenheft, Testspezifikation und Testsuite müssen eine kontinuierlicher ableitbarer Prozess von Pflichtenhefterstellung mit den Kunden bis zur Inbetriebnahme der Software vor Ort ergeben.

Hieraus ergaben sich folgende Lösungen:

- Entwicklung eines allgemeinen - fachunabhängigen - ereignisgesteuerten Simulationskernes.
- Einbindung einer SPS-5/7-Simulation in den Simulationskern.
- Pluginschnittstelle für die Simulation kundenspezifischer Fördertechnik.
- Pluginschnittstelle für die Simulation der Schnittstellen zu Fremdsystemen, wie SAP oder andere Warenwirtschaftssystemen.
- Pluginschnittstelle für erweiterte, ggf. projektspezifische Simulationskommandos
- Kommandoschnittstelle für alle Simulationsbefehle. Die eigentlichen Skripte für die einzelnen Unittests werden in Batches (.cmd) definiert.

Die automatisierte Test- und Simulationsumgebung wurde in mehreren Projekten erfolgreich für die Entwicklung und Qualitätssicherung eingesetzt und wurde als Standardwerkzeug innerhalb des Unternehmens eingeführt.

Realisation eines Kundenprojektes Lagerverwaltung

Ein niederländischer Lebensmittelkonzern setzte seit ca. 15 Jahren die Lagerlösung (Palettenfördertechnik, Regalbediengeräte für mehrfache Lagerung nebst SPS-Steuerung

und Lagerverwaltung- und Materialflussoftware) für ein vollautomatisches Tiefkühlager (-28 Grad Celsius) ein. Da die Anbindung an die Produktionlinien hardware- und softwaretechnisch überholt, in der Fördertechnik Erweiterungen notwendig wurden, sowie in dem Unternehmen SAP eingeführt wurde, wurde eine neue Software für die Steuerung und Verwaltung des Tiefkühlagers notwendig.

- Erstellung eines Feinpflichtenheftes.
- Anpassen des Materialflusses.
- Erstellung der Produktionsanbindung.
- Einbindung von Einlager- und Kontrollscannern.
- Anpassungen der Bediener-GUI.
- SAP-Anbindung.
- Erstellung der Simulationstests.
- Inbetriebnahme der Software.

Der Kunde zeigte sich sehr positiv überrascht, wie schnell und reibungslos die Inbetriebnahme der neuen Software durchgeführt werden konnte. So musste der Betreiber seine Produktionsanlagen nicht herunterfahren und die neue Software konnte nach ca. 2 Stunden Installation und Datenübernahme von der alten Software produktiv bleiben.

4.2 Stabilisierung webbasierte Logistik Lösung

Stabilisierungsmassnahmen, Entwicklung von Test- und Releasemanagement und Performancetuning für eine PHP Oracle SQL Lösung für einen Konzern im Bereich Transportlogistik.

Zeit:	2003-09
Kunde:	Norddeutscher Konzern für Transportlogistik
Position:	Troubleshooting, Entwicklung, Beratung
Technologien:	Oracle SQL, PHP, Distributionsplanung, Logistics

4.3 Projekt Market Data Distribution System (MDDS)

Das Market Data Distribution System (MDDS) ist eine Datenbanklösung zum Sammeln, Verwalten und Verteilung von Börsendaten.

Zeit:	2002-03 - 2003-12
Kunde:	Commerzbank AG Frankfurt
Position:	Softwareentwicklung
Technologien:	C++, Solaris E10000, NT 4, SQL, Sybase, Asset Control, Perl, PHP, TCP/IP, XML, Apache, COM, VisualBasic, Access, Excel, Rational Rose, ClearCase, Purify, GenWeb, Client-Server Architektur

Inhalt dieses Kapitels:

- Projektbeschreibung
- Allgemeine Tätigkeiten
- Tätigkeitsschwerpunkte
- Werkzeuge

Projektbeschreibung

Innerhalb einer deutschen Grossbank werden die Börsendaten mehrerer Lieferanten (z.B. Reuters, Bloomberg) in einem zentralen System gesammelt und aufgearbeitet (normalisiert, priorisiert, mit Snapshot-Views versehen, etc.) und den verschiedenen Abteilungen der Bank wieder zur Verfügung gestellt.

Da das bestehende Altsystem nicht mehr den formalen und inhaltlichen Erfordernissen genügte, wurde eine Neuentwicklung des Market Data Distribution System (MDDS) vorgenommen.

Die Marktdaten bestehen aus den statischen Eigenschaften der Finanzinstrumente (Handelsplatz, Währung, Laufzeit, Unterliegende Instrumentdefinitionen, etc.) und den zeitabhängigen Daten, wie etwa Tick-by-Tick Preise, Tageshöchstpreise etc.

Die statischen Daten werden in einer Sybase Datenbank vorgehalten.


Die dynamischen Daten werden in einem System namens "Asset Control" einer Niederländischen Firma vorgehalten.

Allgemeine Tätigkeiten

- Softwareentwicklung C++ unter Solaris (Sun Workshop, Emacs, Sniff+) und NT (Visual Studio 6) in gesamter Bandbreite (Design bis Distribution) unter Verwendung der STL mit Erweiterungen von Boost.
- Coaching von Softwareentwicklern.
- Leitung eines Entwicklerteams (2-5 Softwareentwickler).
- Beratung und Unterstützung externer Nutzer und Entwickler.

- Softwareentwicklung Perl.
- Softwareentwicklung COM.
- Erstellung der System- und Anwendungsdokumentation in Englisch.
- Softwareentwicklung Basic/Excel.
- Beratung der Projektleitung.

Tätigkeitsschwerpunkte

- Entwicklung einer C++-Schnittstelle für eine Datenbank zur Verwaltung von Zeitreihen von Börsendaten (Produkt: "Asset Control"). Implementierung diverser Filtermöglichkeiten innerhalb der Zeitreihen.
- Entwicklung von Frameworkklassen: Threads, Logging, Job-Scheduling, Generische Typen, AnyVal-Typen, Propertysets, Exceptions, Sockets, etc. für Solaris und Windows NT.
- Design und Entwicklung eines Kommunikationsframeworks für Sockets, Filetransfer, XML und Binärprotokollen: Market Data Customer Interface (MDIF). Spezifikation des Protokolls für die Abfrage von Marktdaten. Entwicklung der MDIF Serveranwendung. Entwicklung der C++-Client Bibliothek. Entwicklung einer alternativen C++-Schnittstelle für die Koppelung mit Java-Clients. Entwicklung einer COM-Clientschnittstelle. Entwicklung eines Excel-Sheet auf Grundlage der COM-Clientschnittstelle für das Abfragen von Börsendaten.
- Entwicklung eines verteilten Kommunikationsservers.
- Entwicklung eines Dokumentationssystems auf Grundlage von HTML, PDF und HTMLHelp auf Grundlage von  GenWeb.
- Entwicklung einer Webanwendung zur Überwachung und Steuerung des MDDS systems unter Apache 2, Perl, C++, PHP, GenWeb, CSV
 - Log-File Analyse
 - Abfrage von Instrumenten.
 - Statistiken zur Auslastung des Systems
 - Statistiken zur Abrechnung mit den Kunden des MDDS
 - Fehlerterminal
 - Scheduling-Kontrolle der Batch-Jobs und Serverprozesse.
 - Plugin-Schnittstelle für weitere Kundespezifische Reports
 - Graphische Darstellung von Statistiken.
 - Integriertes Hilfesystem.

Werkzeuge

- Solaris (E4500 & E10000)
- NT 4.0
- Sun Workshop 4.x - 5.2
- Sniff+
- Visual Studio 6
- Sybase
- Asset Control
- Rational Rose, ClearCase, Purify, ClearQuest
- GenWeb

4.4 Projekt ACDK (Maja Objects)

ACDK (früher Maja Objects) ist ein plattformunabhängiges Framework für C++ für die Realisierung von Serveranwendungen mit einer Orientierung an Java.

Zeit:	1999 - 2003
Kunde:	Diverse Endkunden, Inhouse
Position:	Geschäftsführung, Entwickler, Entwicklungsleitung
Technologien:	C++, Java, Compilerbau, COM, CORBA, SQL, XML, Perl, Python, PHP, Tcl, Lisp, TCP/IP

Inhalt dieses Kapitels:

- ☒ Motivation der Entwicklung eines C++-Frameworks
- ☒ Eigenschaftskatalog des Frameworks
- ☒ Geschäftsführung Micromata_objects
 - ☒ Entwickeln der Kernbestandteile und SAP R/3-Schnittstelle
 - ☒ Erstellen eines Businessplans
 - ☒ Projektrealisierung Atrax
- ☒ Nachfolge des Frameworks 2001 bis heute

☒ **Motivation der Entwicklung eines C++-Frameworks**

Als Grundstein für die Abwicklung von Projekten im Bereich Logistik brauchte die Firma Micromata ein Software Framework. Fachliche Grundlage für solch ein Framework erarbeitete ich für die MAN Logistics (siehe ☒ [LMF-Framework](#)).

☒ **Eigenschaftskatalog des Frameworks**

Besondere Anforderungen an dieses Framework waren:

- Für die Ansteuerung von Fördertechnik, Regalbediengeräten und Roboter zeitnahe Reaktionen der Software-Systeme.
- Ansteuerung von hunderten Hardwaresystemen (Scanner, Waagen, Steuerungssysteme, Funkterminals, etc.)
- Koppelung gegen SAP R/3 über die IDOC RFC Schnittstelle.
- Nutzung von SQL-Datenbanken.
- Threading / Verteilte Anwendungen.
- Unterstützung von diversen Protokollen, wie TCP/IP, OSI, Profibus, Seriell, Funkterminals, CORBA.
- Einbettung von Scriptsprachen für Customizing des Systems (z.B. Perl, Tcl).
- Muss auf Unix und NT Systemen laufen.
- 365 Tage / 24 Stunden Verfügbarkeit.
- Unterbrechbarkeit, Wiederanlauf, Persistenz des Gesamtsystems

Weitere Anforderungen ergeben sich aus der Projektabwicklung:

- Framework muss für einen durchschnittlichen Programmierer beherrschbar sein.
- Das Softwaresystem muss in kurzer Zeit erstellt werden, da ansonsten zwischen Aufbau der Fördertechnik und Inbetriebnahme der Logistikgesamtlösung sehr viel Kapital gebunden ist.
Dies hat zur Konsequenz, dass für alle technischen Standardanforderungen (wie Netzprotokolle, Datenbankzugriff, SAP-Anbindung, Persistenz, Threading, Logging, etc.) schon vorgefertigte Lösungen bereitstehen müssen.
- Fehler müssen schon durch das Bibliothekdesign vermeidbar sein (z.B. Speichermanagement, das oft noch Entwickler Monate nach offizieller Abnahme des Projektes bindet).
- Gute Logging, Tracing, Debugging Funktionalitäten.

☒ **Geschäftsführung Micromata_objects**

Mit der Micromata_application GmbH und Micromata_logistics GmbH gründete ich die Micromata_objects GmbH - der ich als alleiniger Geschäftsführer vorstand - mit dem Ziel, ein C++-Framework zu entwickeln um sie einerseits im Projektgeschäft im Bereich Logistics und E-Commerce einzusetzen und andererseits mit Hilfe von Venture Capital zu vermarkten.

☒ **Entwickeln der Kernbestandteile und SAP R/3-Schnittstelle**

Meine Aufgabe bestand in den ersten 3 Monaten darin, die Kernbestandteile des Frameworks Maja (heute ACDK) zu entwickeln.
In diesem Rahmen wurde auch die SAP R/3 Schnittstelle entwickelt (siehe ☒ [AILA](#) und ☒ [ITalkIDOC](#)) und eingesetzt.

☒ **Erstellen eines Businessplans**

Für die Gewinnung eines Venture Kapitalgebers entwickelte ich einen Businessplan für Micromata_objects GmbH. Parallel entwickelte ich mit 3 Angestellten weitere Komponenten des Frameworks.

Nachdem wir eine Reihe von Gesprächen und Vorverhandlungen mit möglichen Kapitalgebern geführt hatten, kamen wir aber zu dem Schluss, dass einerseits die vorgestellte Geschäftsidee für die typischen Venture-Kapitalgesellschaften nicht attraktiv genug war (technisches Produkt, Industrieprojekte, kein Webportal) und andererseits die von den möglichen Kapitalgebern gestellten Bedingungen für die Gesellschafter von Micromata_objects GmbH nicht akzeptabel waren.

☒ **Projektrealisierung Atrax**

Für ein B2B Projekt unter der Federführung der Micromata_applications GmbH entwickelten wir eine generische SQL-Schnittstelle (ODBC, Oracle).
Siehe auch ☒ [Atrax](#).

☒ **Nachfolge des Frameworks 2001 bis heute**

Die Weiterentwicklung des Frameworks ACDK wird weiterhin durch mich als OpenSource Projekt vorangetrieben.

Für weitere Informationen siehe bitte:

☒ <http://acdk.sourceforge.net>.

4.5 Projekt FriendScout24

Stabilisierung und Reengineering einer E-Commerce-Plattform unter Sun, Oracle, Apache, Perl.

Zeit:	2000-06 2000-8
Kunde:	FriendScout24
Position:	Geschäftsführung, Projektleitung, Entwicklung
Technologien:	Sun Solaris, Oracle, Fast-CGI, Perl

Für die Firma FriendScout24, die unter www.friendscout24.de einen populären Dating-Service betreibt, hat Micromata_objects eine Stabilisierungsmaßnahme und Steigerung der Performance der Serversoftware vornehmen können.

4.6 Projekt ProjectForge

Webbasiertes Projektmanagement-Werkzeug.

Zeit:	2000
Kunde:	Inhouse, Endkunden
Position:	Geschäftsführung, Projektleitung, Vertrieb, Entwicklung
Technologien:	PHP, CMS, MySQL, Perl, Shell, Linux

Bei meinen Tätigkeiten als Entwickler, Projekt- und Entwicklungsleiter konnte ich schon früh erfahren, wie wichtig eine gute Kommunikation und Steuerung innerhalb des Projektteams und mit den Kunden ist.

Einer der wichtigsten Gründe für das mögliche Scheitern eines Projektes liegt nicht in technologischen oder inhaltlichen Problemlagen, sondern im Sand des Kommunikationsgetriebes zwischen Projektteams und Kunden.

Während ein Kunde aus dem Felde des E-Commerce gleich von Beginn ein 'Du' anbietet, erwartet ein Kunde aus der Industrie die Einhaltung eines verbindlichen Tones. Auch bei den Standards des Projektmanagements findet man sehr verschiedenartige Vorstellungen. Unabhängig von diesen unterschiedlichen Erwartungen ist eine offene und dokumentierte Kommunikation des Projektes wesentlich. Dabei ist es unerheblich, ob Protokolle von mündlichen Vereinbarungen fixiert werden, oder eine Projektabwicklung nach ISO 9000/9001 dokumentiert wird.

Eine Dokumentation schafft Vertrauen und Sicherheit bei allen Beteiligten.

Bei verschiedenen Projekte musste ich beim Eintritt zunächst grundlegende Formen der Dokumentation einführen - angefangen vom bislang fehlenden Firmenbriefkopf, bis hin zu zum Erstellen eines Qualitätsmanagement Handbuchs.

Eine nicht unwesentliche Aufgabe ist die Durchsetzung der QS-Maßnahmen innerhalb eines Teams. Papier ist geduldig und Grundsätze und Richtlinien haben oft eine Halbwertszeit von wenigen Wochen.

Eine Alternative besteht in einer Plattform für Zusammenarbeit, bei der die Zugangsschwelle für alle möglichst niedrig gehalten wird und mit der auch alle gerne arbeiten.

- Werkzeug für alle Teammitglieder und keine Zugangsgrenzen für reine Managerwerkzeuge (z.B. MS Project) und Entwicklerwerkzeuge (z.B. Versionsverwaltung).
- Direkte Dokumentation innerhalb des Systems.
- Dokumentation der Kundenkommunikation (Telefonate, Protokolle).
- Einbindung externer Dokumentation (Word, PDF)
- Integriertes Task- und Bugtracking.
- Projektsteuerung (Zeitpläne, Aufwandaufschätzungen, Personalplanung).
- Verschiedene Zugriffslevel für Kunden, Entwickler und Manager.
- Eingebautes Versions-Management.
- Abrechnungen für den Kunden können aus Tagesberichten der Entwickler erstellt werden.
- Verwaltung von Regressionstests und Releases.

Als Vorlage für solch eine Plattform fand ich in SourceForge (<http://sourceforge.net>, mit denen OpenSource Projekte verwaltet werden. SourceForge ist als LAMP-System (Linux, Apache, MySQL, Perl/PHP) aufgebaut und selbst als OpenSource verfügbar.

Durch Anpassungen und Erweiterungen, die für den Einsatz innerhalb kommerziellen Projekten notwendig waren, entwickelten wird ProjectForge.

Die Anpassungen mussten im Bereich Rechteverwaltung, Struktur und Benennung der Webseiten und Integration einiger Bauteile der Plattform vorgenommen werden.

Die Erweiterung besteht hauptsächlich aus einem Modul für die Erstellungen von Dokumenten ([GenWeb](#)), mit dem man Standarddokumente (Lastenheft, Spezifikation, Protokolle, Telefonnotizen, etc.) in HTML erstellen und ggf. als PDF exportiert kann. Besonders wichtig hierbei ist, dass die erstellten Dokumente in dem Projektweb strukturiert eingefügt werden und so für alle einfach wieder zu finden sind.

Im Gegensatz zu einem externen Qualitätsmanagement Handbuches fällt es bei ProjectForge auch allen Mitarbeitern leicht, zu einer guten Projektabwicklung beizutragen.

Nach einer Evaluation innerhalb der Firmengruppe Micromata_group (weitere Produkte aus dem LotusNotes und Exchange Umfeld wurden getestet) wurde ProjectForge zum Standard für die Abwicklung von Projekten gemacht.

Das Feedback von den Projektmitarbeitern und vor allem auch der Kunden aus dem Umfeld Industrie und Dienstleistung war sehr positiv.

Ich stellte ProjectForge auch auf der Veranstaltung "Open Exchange Deutsche Bank & Guests on 2000-05-23" unter weiteren Beiträgen von IBM, Oracle, SAP und der Deutsche Bank zu dem Thema "Managing Software Development through Key Performance Indicators" vor.

4.7 Projekt GenWeb

Autorenwerkzeug für die Erstellung von Webseiten, Handbüchern, Hilfedateien und Quelltextdokumentationen.

Zeit:	2000
Kunde:	Inhouse, Commerzbank AG
Position:	Entwicklung, Geschäftsführung, Projektleitung, Beratung
Technologien:	Perl, PHP, CMS

Für die einfache und effiziente Erstellung von Dokumentationen und Webseiten erstellte ich ein Autorensystem, das den Alltagsanforderungen eines Unternehmens im Softwarebereich entspricht, aber auch von ungeschulten Mitarbeitern bedient werden kann. Innerhalb der Micromata_group etablierte sich GenWeb als Standard für die Erstellung von Webseiten, Projektbüchern, Projektdokumentationen, Softwaredokumentationen für Endbenutzer und interne Softwaredokumentation.

4.8 Projekt Atrax

B2B Plattform auf der Basis von ACDK, Oracle, Java, XML.

Zeit:	2000
Kunde:	Dialog AG
Position:	Geschäftsführung, Projektleitung, Entwicklung, Beratung
Technologien:	ACDK, C++, Java, XML, Oracle, RPC

Beim Projekt Atrax handelt es sich um die Entwicklung eines Systems, mit dem die Firma Dialog AG den Vertrieb von Telekommunikationsverträgen (Mobil, Festnetz und Internet) mittels Vertriebspartner organisiert.

Dabei werden Produkte verschiedener Firmen vertrieben, wobei dem Kunden das günstigste Produkt für seine Zwecke angeboten wird. Das von der Micromata_applications entwickelte System ermöglicht eine optimale Abbildung der Geschäftsprozesse der Dialog AG. So gestaltete sich beispielsweise die Zuordnung von Provisionen zu Vertriebsmitarbeitern und Verträgen in der Vergangenheit als äußerst kompliziert. Der Grund dafür lag nicht nur darin, dass die Anbieter verschiedene Modelle vertrieben, sondern auch, weil das Finden und Vergleichen aller Verträge per Hand schlicht nicht möglich war.

Serverseitig wurde auf der Basis des ACDK ein Applicationserver erstellt. Dieser Applicationserver bietet ein gesichertes Umfeld, um unter dem Betriebssystem Solaris von Sun auf eine Datenbank von Oracle über die MDBC-Schnittstelle zugreifen zu können. Über das offene Protokoll SOAP können Client-Anwendungen Funktionen des Applicationserver aufrufen.

Vorteilhaft bei diesem Vorgehen war die Laufzeitstabilität und Entwicklungskomfort bei der Implementierung der serverseitigen Businesslogik. Auf der Klientenseite wurde für die Kunden einerseits ein Zugang über einen Web-Server ermöglicht und andererseits für Vertriebsmitarbeiter ein erweiterter Zugang über eine mittels einer in Java erstellten Clientanwendung. Besonders vorteilhaft wurde der Einsatz von ACDK als Implementierungsgrundlage für die serverseitigen Prozesse, da Algorithmen und Prozeduren

zwischen dem C++-Server und Java-Clients 1:1 übertragen werden konnten. Auch sonst konnte im Projekt auf einen durchgängigen Standard vertraut werden.

4.9 Projekt ITalkIDOC

SAP R/3-Koppelung für AC DK

Zeit:	2000
Kunde:	Endkunden, Inhouse
Position:	Geschäftsführung, Projektleitung, Entwicklung
Technologien:	ACDK, C++, SAP R/3, SAP RPC, SAP IDOC, Oracle

Für die Anbindung von SAP R/3 an C++ Projekten entwickelte ich mit meinen Mitarbeitern eine generische Koppelung auf der Basis von SAP RPC, die die IDOC Datentabellen in Collections verfügbar macht.

Durch den Einsatz von ACDK als Basistechnologie ist es möglich diese Bridge über folgende Technologien anzubinden:

- C++-Bibliothek
- SQL Datenbank (Oracle, ODBC)
- XML oder Textformaten über Dateiaustausch oder TCP/IP.
- COM oder CORBA
- Java
- Scriptsprachen, wie Perl, Python, Tcl, Lisp

Siehe auch:

- [▶ AILA](#)
- [▶ LMF-Framework](#)
- [▶ Fresenius](#)

4.10 Projekt AILA - Constructor

Realisierung einer SAP R/3-Koppelung Aircabin/DASA für ein Lager der Aircabin GmbH (Tochter im DASA/DC Konzern) mittels ITalkIDOC.

Zeit:	2000
Kunde:	Constructor GmbH, Aircabin GmbH, Micromata_logistics GmbH
Position:	Geschäftsführung, Projektleitung, Beratung
Technologien:	ACDK, C++, SAP R/3

Für ein Lager der Aircabin GmbH (Tochter im DASA/DC Konzern) in Laupheim stellte die Constructor GmbH einen Lagerverwaltungsrechner. Für eine Koppelung dieses Lagerverwaltungsrechners und dem SAP R/3 System von Aircabin musste eine Middleware erstellt werden.

In Zusammenarbeit mit Micromata_logistics wurde hierfür die mit und für ACDK entwickelte SAP-Schnittstelle eingesetzt. ACDK-Klassen ermöglichten einen komfortablen und einfachen Zugriff auf die IDOC-Struktur des SAP-Systems. In kürzester Zeit konnte zur vollsten Zufriedenheit der Constructor und Aircabin die Schnittstelle zwischen den Systemen erstellt werden, so dass die Inbetriebnahmezeit vor Ort auf einen Tag und die Nacharbeiten (kleine

Änderungen in den Telegramminhalten) innerhalb weniger Wochen abgeschlossen werden konnte.

4.11 Projekt LMF-Framework

Im Auftrag der MAN Logistics GmbH entwickelte ich als Projekt- und Entwicklungsleiter ein generisches Materialflusssystem und Prozessleitsystem mit einer Simulationsumgebung.

Zeit:	1999
Kunde:	MAN Logistics GmbH
Position:	Projektleitung, Entwicklungsleitung
Technologien:	C++, Java, CORBA, SQL, Fördertechnik, Materialfluss, Warenwirtschaft

Das LMF-Framework behandelte folgende grundlegenden Aspekte:

- Konzeption für alle Projektphasen, von der Akquise bis zur Optimierung bei der Realisation von Lagerlogistiklösungen.
- Softwarekomponenten für diese Projektphasen (Akquise, Planung, Entwicklung, Integration/Inbetriebnahme, Betrieb, Optimierung, Erweiterung).
- Methoden des Softwareentwurfs für die flexible und generische Realisierung der Softwarekomponenten für die verschiedenen Projektphasen.
- Technologische Konzepte für die Umsetzung auf der Grundlage von C++ (ACDK), Java und CORBA.
- Technologische Konzepte für die generische Realisierung aus den Bereichen von Lagerverwaltung, Materialfluss und Fördertechnik.
- Realisierungsfahrplan unter Einbeziehung von Projektrealisierungen.



*Komponentenbasiertes Business-Framework für
Lagerverwaltung, Materialfluss und Fördertechnik*

Folgende Komponenten sollen durch das LMF-Framework angesteuert werden:

- Hochregalbediengeräte
- Rollenförderer

- Kettenförderer
- Förderbänder
- Tablarlager
- Roboter und Maschinen wie Wickler
- Funkterminals, Scanner, Drucker

4.12 Projekt ArcView - Team Heese

Für die Team Heese AG entwickelte ich eine Schnittstelle für ArcView GIS zu MS Access.

Zeit:	1998
Kunde:	Team Heese AG
Position:	Entwicklung, Beratung
Technologien:	ArcView GIS, MS Access, Visual Basic, DDE

4.13 Projekt LynxTerm

Erweiterung eines textbasierten Webbrowsers für den Einsatz als GUI für Funkterminals im Bereich Lagerlogistik

Zeit:	1998
Kunde:	INIS GmbH
Position:	Entwicklung
Technologien:	C, HTML, HTTP, VT100

In einem Lagerbetrieb werden Transport- oder Kommissionieraufträge über Funkterminals an die Lagerarbeiter übermittelt. Bei diesen Funkterminals handelt es sich um einfache zeichenbasierte Displays. Die Terminalanwendungen selbst laufen auf einem zentralen Terminalserver unter Unix oder NT.

Um Pfleg-, Wart- und Erweiterbarkeit sowie die Möglichkeit von Wiederverwendung von Klient- und Serveranwendungen zu erhöhen, ist es sinnvoll, für die Terminalmasken und für die Client-Server-Kommunikation etablierte Standards zu verwenden.

Die Idee, Terminalanwendungen als http-Sessions zu handhaben, liegt nahe. Mit Apache als Server und lynx als zeichenorientierten Browser stehen etablierte, auf nahezu allen Plattformen frei verfügbare Softwarekomponenten zur Verfügung. Durch die Verwendung von HTML als Maskenbeschreibungssprache kann man auf eine Vielzahl von Werkzeugen zurückgreifen.

Für die Verwendung als Terminalanwendung brauchte lynx aber einige Erweiterungen:

- Standardfunktionen wie etwa History, Favoriten, etc. müssen deaktiviert sein
- Standardtastaturbelegungen wie etwa Q für Quit müssen deaktiviert sein
- Statuszeile muss abschaltbar oder modifizierbar sein.
- Links müssen mit (mehreren) Hotkeys belegbar sein.
- Interne Browserfunktionen (Quit, Back to last page, etc.) müssen frei mit Links und Tastatur assoziierbar sein.

- Da bei klassischen Terminalanwendungen für jedes Terminal ein eigener Prozess auf dem Terminalserver läuft, sollte der Speicherverbrauch durch shared libraries (.so, .dll) bei mehreren Instanzen minimiert werden.

4.14 Projekt Fresenius

Als Entwicklungs- und Projektleiter führte ich 1997 ein Softwareentwicklungsprojekt im Bereich Lagerlogistik, Lagerverwaltung und SAP R/3-Anbindung durch.

Zeit:	1997-03 - 1997-12, ca 200 Manntage
Kunde:	INIS GmbH, IRS, MAN Logistik, Fresenius AG
Position:	Entwicklungs- und Projektleitung, Entwicklung
Technologien:	C++, Sun Solaris, Informix, ESQL, SAP R/3, Perl, Tcl, Motif, Logistik, Fördertechnik, Produktionsautomatisierung, Lagersteuerung, UST, OSI-Bus, Funkterminal, Robotersteuerung, Scanner

Die Fresenius AG hat in Friedberg bei Frankfurt für die Kundenauslieferung und Umschlagstelle für Produktion und Nebenlager eine Logistikzentrale für Deutschland errichtet. Diese Logistikzentrale besteht aus einem Hochregallager, Förderstrecken von einem Rampenbereich für die An- und Auslieferung der Waren via LKW, mehreren Kommissionierbereichen sowie einer Anbindung des Lagerbereiches an einen Produktionsbereich.

Die Fördertechnikelemente wie das Hochregallager und die Förderstrecken für den Transport von Euro-A Paletten wurden von dem Generalunternehmen MAN Logistik erstellt. Die Firma Ro-ber installierte Umpack- und Kommissionierroboter. Die buchhalterische Steuerung des Lagers wird durch ein SAP-System mit firmeneigenen Erweiterungen der Fresenius AG vorgenommen. Die Firma IRS GmbH Berlin hatte die Aufgabe, durch Erstellung von geeigneten Hardware- und Softwarelösungen eine Verbindung der verschiedenen oben genannten Teilsysteme für ein funktionierendes Logistiksystem zu erstellen.

An Hardwarelösungen fallen darunter mobile Funkterminals, Scannersysteme, verschiedene Drucker und deren Anbindung an die unterlagerten Systeme. Softwareseitig mussten die verschiedenen über- und unterlagerten Systeme gekoppelt werden (SAP, Fördertechnik, Funkterminals, Drucker, Scanner, Roboter), Transportstrategien für ein vollautomatisches Lager entworfen werden, eine Benutzerführung für die Lagerarbeiter (vt100 Terminals, X-Terminals, PCs unter Sun Solaris) und eine Leitstandsoftware inklusive einer Visualisierung der Anlage für die Steuerung der Anlage erstellt werden.

Inhalt dieses Kapitels:

- Beschreibung
- Tätigkeiten als Projektleiter
- Tätigkeiten als Entwicklungsleiter
- Tätigkeiten als Entwickler

Beschreibung

Durch die Vermittlung der Firma INIS GmbH Kassel wurde ich von der Firma IRS beauftragt, das Projekt über den Zeitraum von einer Woche zu beobachten. Nach dieser Zeit wurde ich beauftragt, das Projekt als Softwareentwickler zu unterstützen. Nach 4 Wochen wurde ich als Projekt- und Entwicklungsleiter benannt.

Tätigkeiten als Projektleiter

- Beratung der Firmenleitung zu Geschäftsentscheidungen.
- Besprechungen mit den beteiligten Firmen.
- Erstellung von Pflichtenheften.
- Aufsicht des Projekts gemäß ISO 9001.

Tätigkeiten als Entwicklungsleiter

- Zusammenstellung eines geeigneten Programmiererteams.
- Leitung und Koordination von 9 Softwareentwicklern.
- Auswahl der Entwicklungswerkzeuge.
- Erstellung von Funktions- und Terminplänen.
- Erstellung und Durchführung von Testplänen.
- Koordination mit den beteiligten Firmen.

Tätigkeiten als Entwickler

- Überarbeitung der Schnittstelle zu dem übergelagerten SAP-System
- Kontrolle der Anbindung zu der unterlagernden Steuerung.
- Design und Entwicklung eines mehrschichtigen Modells für das Materialflusssystem und ihre internen Protokolle.
- Ausarbeitung eines Objektmodells für das Materialflusssystem und Beteiligung an der Entwicklung.

Erstellung einer Simulationsumgebung mit Visualisierung der Steuervorgänge:

- zur Überprüfung im Vorfeld der Funktions- und Durchsatzvorgaben auf Machbarkeit
- für die Durchführung von Tests der einzelnen Module des Materialflusssystems auf Programmierfehler.
- Für die Durchführung von Tests der Module im Zusammenspiel untereinander und den beteiligten über- und unterlagerten Systemen auf korrekte Ablaufsteuerung.

4.15 Projekt PowerUser, Computer Associates

Für eine Produktpalette, die für den internationalen Massenmarkt bestimmt war, lizenzierte Computer Associates International das von mir erstellte Programm 'New Menus'. Von 1993 bis 1997 war ich für diese Firma auch als Berater tätig.

Zeit:	1993-1997, davon 1995 - 1997 im Auftrag von CA (ca. 500 Manntage)
Kunde:	Computer Associates International (CA), Endkunden
Position:	Softwareentwicklung, Vertrieb
Technologien:	C++, Windows 3.1/95/NT, OS/2, COM, Window-Manager



Der Eingangsbildschirm des Produktes HomePlace

Aus Interesse und um die Programmiersprache C und C++ zu erlernen, schrieb ich während meines Studiums ein Desktop-Tool 'New Menus for Windows' für Windows 3.1, das einige Aspekte (Look and Feel) der Window-Manager des X-Window-Systems unter Unix (Motif, OpenLook, NextStep) für den Windows 3.1 PC zur Verfügung stellt.

Nachdem ich erste Ergebnisse vorzuweisen hatte, veröffentlichte ich das Programm kostenlos im Internet. Durch ein entsprechendes weltweites Feedback wurde ich dazu motiviert, das Programm in einer deutschen und einer englischen Fassung weiterzuentwickeln. Ich vermarktete das Programm als Shareware, wobei ich Distributoren in Deutschland, Frankreich, USA und Japan für den Vertrieb gewinnen konnte.

Die Firma Computer Associates International wurde auf dieses Produkt aufmerksam und bot mir Anfang 1995 für die Integration dieses Produktes in ein für den Consumer-Markt abgestimmtes Paket eine Lizenznahme an. Mit der Lizenzierung war ein Beratervertrag über 2 Jahre für die Portierung dieses Programms nach Windows 95, Windows NT und OS/2 verbunden.

Das Gesamtprojekt 'HomePlace' wurde verteilt abgewickelt, wobei Teams in Florida, Californien und NY beteiligt waren. Ich steuerte meine Komponente 'PowerUser' aus Kassel bei.

Die Hauptkomponente des Produktes "HomePlace", durch 4Home Productions veröffentlicht

(eine Division von CA), bestand in einer graphischen Shell für Windows, die Elemente eines 3-D Spieles hat und für verschiedene Altersstufen und Fähigkeiten eine Metapher für die Bedienung von Windows zur Verfügung stellt.

Weitere Dienste, wie Spracherkennung und -steuerung, Anbindung an Internet und Onlinedienste sowie eCommerce waren integrale Komponenten. Speziell für den Aufbau von virtuellen Verkaufs- und Serviceräumen wurden Drittkunden ein SDK angeboten. Das Produkt wurde mit einer Auflage von 5 bis 7 Millionen Kopien in Japan und USA für das erste Jahr projiziert.

Für die erfahrenen Nutzer, denen die graphischen Metaphern zu spielerisch sind, gibt es das Modul 'PowerUser', das im Wesentlichen eine angepasste Version von 'New Menu Window Manager' darstellt.

Um eine Portabilität zwischen Windows 3.1, Windows 95, Windows NT 3.51, NT 4.0 und OS/2 (letzteres wurde nach kurzer Zeit aus der Anforderungsliste gestrichen) zu gewährleisten, musste das Programm von Grund auf neu geschrieben werden. Hierbei sollte das Programm nicht nur auf verschiedenen Plattformen laufen, sondern auch weitgehend kompilierunabhängig geschrieben werden.

Ich entwickelte eine Klassenbibliothek, die die Windows & OS/2 API in C++-Klassen kapselt. Vorhandene Bibliotheken, wie etwa OWL oder MFC waren nicht in einer plattform- und kompilierübergreifenden Version verfügbar und entsprachen auch sonst nicht den spezifischen Bedürfnissen des Projekts. Aus der Entwicklung der Vorgängerversion gewann ich die Erkenntnis über die Wichtigkeit eines möglichst klaren, flexiblen und leistungsstarken objektorientierten Designs, das nicht nur die bestehenden Spezifikationen erfüllt, sondern auch zukünftige Entwicklungen ermöglicht.

Der Entwurf gliedert sich in folgende Objektgruppen:

- Items beinhalten die wesentlichen Funktionselemente des Programms: Verzeichnisse, Dateien, Programme, Prozesse, Virtuelle Desktops, etc. Sie haben die Fähigkeiten sich selbst darzustellen, auf Ereignisse (wie etwa Mausklick) zu reagieren, ggf. selbst wieder eine Liste von Items zu erzeugen.
- Container beinhalten Listen, Hierarchien oder Gruppen von Items. Sie legen fest, in welcher Weise Items dargestellt werden (als Popupmenu, Liste, Baumstruktur, Browser, Titlebar, Taskbar oder als Deskbar etc.) und welche Nachrichten an die Items weitergeleitet werden.
- Windows bilden nicht nur die Schnittstelle zu den dem Betriebssystem-GUI-API sondern funktionieren als Nachrichtendispatcher.
- Frames entsprechen den sichtbaren GUI-Fenstern und erweitern diese auch, so dass verschiedene 'Look and Feel's implementiert werden können.
- Widgets sind erweiterte Controls in spezifischen Kontexten (wie etwa Frames, Windows und Dialoge).
- Styles sind umschaltbaren 'Look and Feel's für die oben genannten Klassen.

Einige Window Controls, die erst unter Windows 95 eingeführt wurden wie etwa die Karteikartenreiter für Dialoge, mussten für Windows 3.1 nachgebildet werden.

Soweit mir von Entwicklern mitgeteilt, bildete NMWM auch eine Referenz für das Design von Gnome und der Swing-Klassenbibliothek für das Java AWT.

NMfW ist im Wesentlichen ein Programm- und Windowmanager. NMWM wurde hier durch ein Konzept des Dateimanagements erweitert.

Wesentliche Features:

Verfügbarkeit von kompatiblen Versionen für folgende Plattformen:

- Windows 3.1, WfW 3.11 (16 Bit)
- Windows 95 (32 Bit)
- Windows NT 3.51 (32 Bit)
- Windows NT 4.0 (32 Bit)

Die Nutzerprofile können in einer gemischten Netzumgebung gemeinsam genutzt werden.

Um unter Windows 3.1 auch eine erweiterte Hilfe anbieten zu können, entwickelte ich ein erweitertes Hilfesystem für Windows 3.1.

Siehe auch [Win4Help](#).

Programm-Management

- Möglichkeit, Programme und Dokumente mit sehr flexiblen Einstellungen zu öffnen.
- Integration der Windows 95 Shell
- Integration der Programm-Manager (Windows 3.1, NT 3.51)

Window-Management

- Verschiedene Window-Stile 'look and feel' (im Sinne von X-Window Window Manager) verfügbar. Diese Wandlungsfähigkeit der Oberfläche war damals recht neu und wurde später in Java Swing, Windows XP, oder auch KDE übernommen.
- Virtueller Desktop



Virtueller Desktop

Datei-Management

- Unterstützung der Standard-Dateioperationen via Drag'n Drop, Shortcuts (Ctrl+C, Ctrl+V, etc.), Kontextmenüs (COM/OLE2).
- Transparente Unterstützung von komprimierten Archiven als Verzeichnisse: ZIP, LZH, ZOO, GZ, ARJ, RAR.
- Unterstützung aller Dateioperationen über eine integrierte Kommandozeilenshell.
- Unterstützung der Shell-Verzeichnisse von Windows 95/NT 4.0 und weiteren Funktionen der Windows Shell.

Konfigurationsmanagement

- Unterstützung mehrerer Nutzer, Unterstützung der Netzwerknutzer unter Windows 95 / NT
- Restriktive Einstellungen für bestimmte Nutzerkreise (daraus entwickelte sich auch ein Miniprojekt 'DeskLock' für HP).
- Transparente Einbindung der Systemsteuerung und anderer externer Konfigurationen.

Scriptfähigkeit

- Integrierte Kommandozeile. Siehe auch [WIOSERV](#), [BSH](#).
- Entwicklung eines erweiterten Scriptsprache in einer .bat und WordBasic kompatiblen Variante. Siehe auch [BSH](#), [SBSI](#).
- Integrierter Tastaturrecorder (nicht mehr realisiert).

4.16 Projekt Basic Shell (BSH)

Eine einfache Scripting Shell.

Zeit:	1997
Position:	Entwickler, Vertrieb
Technologien:	C++, Windows 95/NT, COM

In New Menus Window Manager gibt es vielfältige Möglichkeiten, Dateioperationen durchzuführen. NMWM unterstützt dabei sowohl die virtuellen Verzeichnisse von Windows 95/NT 4.0 als auch Archive als Verzeichnisse. Um den Nutzern diese Möglichkeiten auch in einem Kommandozeileninterface zur Verfügung stellen zu können, entwickelte ich eine Basic Shell, die im wesentlichen die Konstrukte der cmd unter NT und ein paar weitere Merkmale unterstützt. Als Interface wurde WIOSRV verwendet, so daß eine Kommandozeilenshell sowohl allein stehend in einem eigenen Fenster, sowie auch als Control in anderen Fenstern integriert verwendet werden kann.

4.17 Projekt Windows IO-Server

WioServ (Windows IO Server) stellt durch den OLE2 Mechanismus unter Windows 3.1 und Windows95/NT (16/32) ein Fenster zur Verfügung, in dem man textbasierte Ausgaben machen kann.

Zeit:	1997
Kunde:	Endkunden
Position:	Softwareentwicklung, Vertrieb
Technologien:	C++, Windows 3.1/95/NT, COM

WioServ (Windows IO Server) stellt durch den OLE2 Mechanismus unter Windows 3.1 und Windows95/NT (16/32) ein Fenster zur Verfügung, in dem man textbasierte Ausgaben machen kann. Es werden hierbei Escapecodes nach vt100 und Ansi (mit Farben) unterstützt. Im Zusammenhang mit Callbackfunktionen oder OLE2-Interfaces kann der Client Informationen abfragen oder Einstellungen vornehmen (Kodierung, raw/cooked, prompt, Cursor, Zeilenumbruch, Scrollbars, Tab-Größe, etc.).

4.18 Projekt Simple BasicScript Interpreter (SBSI)

Ein Basic Interpreter mit Wordbasic kompatibler GUI-Funktionalität.

Zeit:	1997
Kunde:	Endkunden
Position:	Softwareentwicklung, Vertrieb
Technologien:	C++, Windows 3.1/95/NT, COM

Die in InstScript verwendete Sprachmittel und technische Konzepte für den Interpreterbau sind in vielen Aspekten begrenzt. Um für New Menus Window Manager eine entsprechende Installation und Deinstallation zur Verfügung stellen zu können, brauchte ich ein Werkzeug, daß flexibler in der Handhabung ist. Auch aus Interesse an der Technologie des Interpreter- und Compilerbaus entwickelte ich einen Interpreter für Basic. Die Sprache Basic habe ich nicht wegen ihrer Sprachmerkmale (da wäre wohl fast jede andere Sprache besser geeignet) gewählt, sondern da sie unter den Windowsplattformen durch VBA und VB recht verbreitet ist.

Der unterstützte Sprachstandard entspricht in etwa dem von Wordbasic Version 6.

- Alle Basic Typen inklusive type
- Dialoge über Wordbasic kompatible Dialogdefinitionen.
- Zugriff auf Windows API und andere DLL-Funktionen über 'declare function'
- Präprocessor-Direktiven, wie #include, #if, #const, etc. .

Der eigentliche Interpreter wurde als OLE2/ActiveX-Control realisiert, der in andere Programme integriert werden kann oder via WIOSRV im Interaktiven Modus verwendet werden kann.

4.19 Projekt Win4Help

Integriert die erweiterte Funktionalität der Windows 95/NT4.0-Hilfe in Ihre Windows 3.1-Hilfe.

Zeit:	1996-1997
Kunde:	Endkunden
Position:	Softwareentwicklung, Vertrieb
Technologien:	C++, Windows 3.1

Win4Help ist eine Erweiterung des Windows 3.1-Hilfesystems um die Funktionalität des Hilfesystems von Windows 95 / Windows NT 4.0:

- Auswertung der .cnt-Dateien und Darstellung eines hierarchischen Inhaltsverzeichnisses.
- Volltextsuche innerhalb einer .hlp-Datei.
- Extrahierung des gesamten Textes einer .hlp-Datei.

4.20 Projekt Universität Kassel, ISP

Seit 1993 unterstütze ich die Bibliotheken der Fachbereiche Architektur und Stadt- und Landschaftsplanung in EDV-Belangen von Aufbau eines Servers bis hin zu Erstellung eines Datenbanksystems für die Medien-Bibliothek



Zeit:	1993-2000
Kunde:	Universität Kassel, Informationssystem Planung (ISP)
Position:	Entwicklung, Administration
Technologien:	Access, AIX, Ingres, C++, Windows 95/NT, NT Administration

Access, SQL, C++, Java, Ingres, IIS, Windows 95/NT 4.0 Administration einer NT-Domäne, FTP, email, WWW-Servers.

Portierung und Entwicklung eines Datenbanksystems für eine Media-Bibliothek von BS 2000 nach Ingres/AIX und später nach Access.
DB-Recherche über WWW.

Inhalt dieses Kapitels:

 Administration

-  GraumDB
-  Weitere Arbeiten für das ISP

Administration

Seit 1993 bis heute pflege ich für das Informationssystem Planung (ISP) der Universität Kassel (bestehend aus einer Bibliothek für Graue Literatur, einem Pressearchiv für Planung und Sozialwissenschaften, einem Videoarchiv, einem Lateinamerikanischen Dokumentationszentrums und einem Hochschulverlag für Schriften für den Fachbereich Architektur, Stadt- und Landschaftsplanung) einen Rechnerpool und entwickelte Datenbankanwendungen sowie andere Anwendungen, die im Betrieb und Administration des ISP gebraucht wurden.

In der ersten Phase wurden für das ISP 3 PCs mit Windows für Workgroups installiert und mit dem Hochschulnetz mit TCP/IP verbunden. Mit einem Anwachsen der Benutzer und Rechner wurde eine Serverlösung unter NT installiert. Hierbei wird der NT Server nicht nur für die Speicherung von Daten, Programmen sowie der Benutzerkonfigurationen genutzt, sondern bildet auch eine Schnittstelle zu Diensten des Internets. Email-, FTP- und WWW-Server geben den Betreibern und Nutzern des ISP die Möglichkeit, mit anderen planungsrelevanten Informationsquellen zu kooperieren.

GraumDB

Position: Planende und ausführende Position, Sponsor.


Konversion mehrerer Datenbankanwendungen für den Biobliotheksbereich, die ursprünglich unter einer BS 2000 Anlage mit hauseigener PASCAL-Entwicklung gepflegt wurde. Auf einem Servercluster des Hochschulrechenzentrums wurde unter AIX mit Ingres aus den bestehenden Daten eine relationale Datenbank aufgebaut.

Aufgrund verwaltungstechnischer Umstellungen wurde eine Portierung dieser DB-Anwendung auf eine PC-basierte Lösung notwendig. Wir entschlossen uns, hierfür MS Access zu verwenden.

Entwicklung einer Abfragemöglichkeit der Datenbank über das WWW. Hierfür entwickelte ich mit Java Servlet eine webbasierte Rechercheseite, die die Eingabewerte einer Form auswertet, über JDBC auf die Access-Datenbank zugreift und die Ergebnisse im Internet-Browser anzeigt. Da diese Lösung leider einige Probleme in bezug auf Speicherlöcher und ungenügende Sicherungen gegenüber Konflikten aus gleichzeitigen Abfragen (Irgendwo in der Schnittstelle Java/JDBC/ODBC) aufwies, wurde die Lösung auf C++ umgesetzt. Da Logik und Struktur des Programms beibehalten werden konnte, konnte die Neuimplementierung innerhalb 2 Tagen vorgenommen werden.

Damit auch Rechner älteren Jahrganges (486er mit 8 MB Ram unter Windows 95) als Publikumsrechercheplätze eingesetzt werden können, wurde ein eigener WWW-Browser unter Zuhilfenahme des MS Internet Explorer 4.0 als ActivX-Control entwickelt, der einerseits schlanker ist als der originale Internet Explorer und auch restriktiver eingesetzt werden kann (Keine andere Auswahl, als die vorgesehenen HTML-Seiten, kein Starten anderer Programme oder gar Shell-Zugriff).

Eine Demonstration kann man unter:

 , <http://www.isp.uni-kassel.de/pri/cgi-bin/pubmaske.asp>.
anschauen.

Weitere Arbeiten für das ISP


Aufbau und Administration eines NT-Servers (3.5 – 4.0) mit Windows 95 als Workstation-Klienten. Hierbei sollten Benutzereinstellung (wie Desktop, etc.) unabhängig von den Rechnern verwaltet werden.

Einrichtung und Administration eines FTP, WWW, und email-Servers.

Erstellung diverser Briefvorlagen für MS Winword 2-7, inklusive Adressenverwaltung

Erstellung diverser Excel-Vorlagen für Inventur, Abrechnung, etc.

Siehe auch:

 <http://www.isp.uni-kassel.de>.