



Thema:

**Entwicklung eines webbasierten Informationssystems
zur Unterstützung der Bemusterung im Bereich Qualitätsmanagement
der Robert Bosch Elektronik GmbH**

Diplomarbeit

Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Prof. Dr. rer. pol. habil. Hans-Knud Arndt

Betreuer: Prof. Dr. rer. pol. habil. Hans-Knud Arndt
Dipl.-Ing. Jürgen Hofmann
Robert Bosch Elektronik GmbH — Salzgitter
SzPIQMM1

Vorgelegt von: Stephan Greif

Abgabetermin: 17.01.07

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abbildungsverzeichnis	IV
Verzeichnis der Abkürzungen und Akronyme	VI
1 Einleitung.....	1
2 Grundlagen.....	3
2.1 Qualitätsmanagement und Qualitätssicherung	3
2.2 Qualitätsmanagementsystem	6
2.2.1 Die Normreihe DIN EN ISO 9000:2000.....	6
2.2.2 QM-Systemforderungen in der Automobilindustrie	7
2.3 Bemusterung von Produkten	9
2.3.1 Zweck und Anwendungsbereich.....	9
2.3.2 Grundlegende Forderungen.....	10
2.3.3 Dokumentation und Dokumente	11
3 Ist-Analyse	12
3.1 Allgemeine Vorüberlegungen	12
3.2 Abgrenzung des Untersuchungsgegenstandes.....	14
3.3 Beschreibung und Darstellung des Ist-Zustandes	14
3.3.1 Aufbauorganisation	15
3.3.2 Ablauforganisation.....	16
3.4 Schwachstellenanalyse	24
4 Soll-Konzept	27
4.1 Zielbestimmung.....	27
4.2 Anforderungen.....	27
4.2.1 Funktionale Anforderungen	28
4.2.2 Nicht-funktionale Anforderungen.....	31
4.3 Datenmodellierung	32
4.3.1 Rollenkonzept	33
4.3.2 Mehrsprachigkeit	34
4.3.3 Gesamtkonzept.....	35
4.4 Neustrukturierung der Ablauforganisation.....	37
5 Systementwurf	42
5.1 Gesamtarchitektur	42
5.2 Relationaler Datenbankentwurf.....	44
5.2.1 Grundlegende Vorgehensweise.....	44
5.2.2 Abbildung auf das relationale Modell.....	45
5.3 Entwurf der Applikation.....	49
5.3.1 Dateien	49
5.3.2 Klassen	49

5.4	Entwurf der Benutzeroberfläche	51
5.4.1	Allgemeine Gestaltungsrichtlinien.....	52
5.4.2	Layout und Navigationselemente.....	54
5.5	Konzept zur Internationalisierung und Lokalisierung.....	57
5.5.1	Allgemeine Empfehlungen	58
5.5.2	Grundlegende Konzepte.....	59
5.5.3	Lösungsansatz	60
5.6	Sicherheitskonzept.....	63
5.6.1	Authentifizierung	64
5.6.2	Autorisierung	65
5.6.3	Netzwerk- und Datensicherheit.....	67
6	Realisierung	68
6.1	Datenhaltungsschicht.....	68
6.2	Applikationsschicht	70
6.2.1	Verzeichnisstruktur	71
6.2.2	Konfiguration	72
6.2.3	Datenbankkommunikation	74
6.2.4	Sessionmanagement	77
6.2.5	Lokalisierung	79
6.2.6	Sicherheitsmechanismen.....	83
6.3	Präsentationsschicht	87
6.3.1	Startseite.....	87
6.3.2	Bestellformular.....	89
6.3.3	Suchfunktion	96
6.3.4	Administrationsbereich	98
7	Bewertung der Lösung.....	102
8	Zusammenfassung und Ausblick	106
	Literaturverzeichnis.....	108

Abbildungsverzeichnis

Abb. 2.1: Managementbezogene Begriffe	4
Abb. 2.2: Tätigkeiten und Ergebnisse bei der Dokumentation.....	11
Abb. 3.1: Organigramm der RB Elektronik GmbH (Ist-Zustand).....	16
Abb. 3.2: Beispiel einer EM-Checkliste	18
Abb. 3.3: eEPK zur Abwicklung der EM-Bestellung (Ist-Zustand).....	19
Abb. 3.4: eEPK zur Materialdatenblatterstellung (Ist-Zustand).....	20
Abb. 3.5: eEPK zur Dokumentationsvorbereitung (Ist-Zustand).....	21
Abb. 3.6: eEPK zur EM-Bereitstellung und -Prüfung (Ist-Zustand).....	22
Abb. 3.7: eEPK zur Erstellung der Bemusterungsunterlagen (Ist-Zustand).....	23
Abb. 4.1: Systemübersicht als Anwendungsfalldiagramm.....	30
Abb. 4.2: eERM zum Rollenkonzept (Soll-Zustand).....	34
Abb. 4.3: eERM zur Umsetzung der Mehrsprachigkeit (Soll-Zustand).....	35
Abb. 4.4: Gesamtdatenbankmodell in eER-Notation ohne Attribute.....	36
Abb. 4.5: eEPK zur Abwicklung der EM-Bestellung (Soll-Zustand).....	38
Abb. 4.6: eEPK zur Dokumentationsvorbereitung (Soll-Zustand).....	39
Abb. 4.7: eEPK zur EM-Bereitstellung und -Prüfung (Soll-Zustand).....	40
Abb. 4.8: eEPK zur Erstellung der Bemusterungsunterlagen (Soll-Zustand).....	41
Abb. 5.1: Systemarchitektur als Verteilungsdiagramm	43
Abb. 5.2: Datenbankmodellldiagramm – Teil 1	47
Abb. 5.3: Datenbankmodellldiagramm – Teil 2	48
Abb. 5.4: Ausschnitt aus dem Klassendiagramm der Webanwendung	51
Abb. 5.5: Erste prototypische Realisierung der Benutzeroberfläche.....	55
Abb. 5.6: Navigationsleiste des Informationssystems	56
Abb. 5.7: Mögliche XML-Datenstruktur für die Lokalisierung.....	60
Abb. 5.8: Ausschnitt aus einer XML-basierten Ressourcendatei	62
Abb. 5.9: Trichterprinzip beim Trusted-Subsystem-Modell	67
Abb. 6.1: Benutzeroberfläche des Oracle SQL Developer.....	69
Abb. 6.2: SQL-Abfrage zur Ermittlung der formatierten RB-Sachnummern	70
Abb. 6.3: Verzeichnisstruktur von ISDOS	71
Abb. 6.4: Aufbau der XML-Konfigurationsdatei von ISDOS.....	73
Abb. 6.5: Konstruktor der Klasse clsDBManager	74
Abb. 6.6: Auslesen der Datenbankverbindungszeichenfolge	74
Abb. 6.7: GetOracleCommand()-Methode der Klasse clsDBManager	75

Abb. 6.8: Ausführung eines SQL-Befehls mit anschließender Datenbindung	76
Abb. 6.9: Speichern und Auslesen des Namens einer Benutzerrolle.....	79
Abb. 6.10: Speichern der Spracheinstellung in einem Cookie	80
Abb. 6.11: InitializeCulture()-Methode der Klasse clsI18N.....	81
Abb. 6.12: Auslesen eines Spracheintrags aus einer Ressourcendatei	82
Abb. 6.13: Erzeugung eines Authentifizierungstickets.....	84
Abb. 6.14: Überprüfung der Authentifizierung und Autorisierung in der Klasse clsAuthFilter	85
Abb. 6.15: Startseite mit Anmeldedialog.....	88
Abb. 6.16: Startseite nach erfolgreicher Anmeldung.....	89
Abb. 6.17: Formular zur Erfassung einer Bestellung – Teil 1	90
Abb. 6.18: Formular zur Erfassung einer Bestellung – Teil 2.....	91
Abb. 6.19: Eingabeblock für das PPAP-Verfahren	93
Abb. 6.20: Eingabeblock für ein kundenspezifisches Bemusterungsverfahren.....	94
Abb. 6.21: Information über die erfolgreiche Speicherung einer Bestellung	95
Abb. 6.22: Suchformular für die erweiterten Suche	96
Abb. 6.23: Ergebnisanzeige nach erfolgreicher Suche	97
Abb. 6.24: Formular zum Anlegen von Kundendaten.....	99
Abb. 6.25: Formular zum Ändern von Benutzerrollen.....	100
Abb. 6.26: Formular zum Löschen von Verteilerlisten	101

Verzeichnis der Abkürzungen und Akronyme

ADO.NET	ActiveX Data Objects .NET
AIAG	Automotive Industry Action Group
ANSI	American National Standards Institute
APQP	Advanced Product Quality Planning
ARIS	Architektur integrierter Informationssysteme
ASP.NET	Active Server Pages .NET
C#	CSharp
CI	Corporate Identity
CSS	Cascading Style Sheet
DB	Datenbank
DBMS	Datenbankmanagementsystem
DBS	Datenbanksystem
DDL	Data Definition Language
DGQ	Deutsche Gesellschaft für Qualität e. V.
DHTML	Dynamic Hypertext Markup Language
DIN	Deutsches Institut für Normierung e. V.
DLL	Dynamic Link Library
DmbA	Dokumente mit besonderer Archivierung
DML	Data Manipulation Language
EA	Erstausrüster
eEPK	erweiterte ereignisgesteuerte Prozesskette
eERM	erweitertes Entity-Relationship-Modell
EM	Erstmuster
EN	Europäische Norm
EPK	ereignisgesteuerte Prozesskette
ERM	Entity-Relationship-Modell
EVA	Erstausrüster-Vertriebs-Abwicklung
FCL	.NET Framework Class Library
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IATF	International Automotive Task Force
ID	Identifikator
IIS	Internet Information Server
IMDS	Internationales Materialdatensystem
ISDOS	Initial Sample Documentation Order System
ISO	International Organization for Standardization
J#	JSharp
MAT	Abteilung Materialwirtschaft
MS	Microsoft
MSB/H	Abteilung Fertigung Steuergeräte
OEM	Original Equipment Manufacturer
PPAP	Production Part Approval Process
PPF	Produktionsprozess- und Produktfreigabe
QM	Qualitätsmanagement
QMM1	Abteilung Produktqualität
RB	Robert Bosch
RFC	Request for Comments

SHA	Secure Hash Algorithm
SQL	Structured Query Language
TEF	Abteilung Technische Funktionen
TS	Technische Spezifikation
UML	Unified Modeling Language
URL	Uniform Resource Locator
VB	Visual Basic
VBA	Visual Basic for Applications
VDA	Verband der Automobilindustrie e. V.
VS	Visual Studio
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

1 Einleitung

Die Robert Bosch (RB) GmbH wurde 1886 als „Werkstätte für Feinmechanik und Elektrotechnik“ von Robert Bosch gegründet und ist heute ein weltweit agierender Hersteller von Kraftfahrzeugtechnik, Industrietechnik sowie Gebrauchsgütern und Gebäudetechnik. Im Geschäftsjahr 2005 beschäftigte die RB GmbH rund 251.000 Mitarbeiter in über 140 Ländern. Die Kraftfahrzeugtechnik bildet dabei den größten Unternehmensbereich der RB GmbH. Weltweit waren in diesem Bereich im Jahr 2005 rund 158.000 Mitarbeiter an 120 Fertigungsstandorten beschäftigt (vgl. Robert Bosch GmbH 2006, S. 4 ff.).

Wie in jedem anderen Unternehmen spielt die Qualität der Produkte zur Sicherung der Wettbewerbsfähigkeit eine sehr wichtige Rolle. Für Zulieferer der Automobilindustrie, wie es die RB GmbH ist, gelten branchenspezifische Normen für das Qualitätsmanagement. In diesen werden übergreifende Abläufe und Verfahren dargestellt, mit dem Ziel ein einheitliches Qualitätsniveau in der Branche zu schaffen. Im Rahmen dieser Diplomarbeit wird die Bemusterung von Produkten als ein Verfahren zur Sicherung der Qualität von Lieferungen am Beispiel der RB Elektronik GmbH genauer betrachtet.

Mit der Bemusterung soll ein Lieferant vor Serienbeginn den Nachweis erbringen, dass seine Produkte die mit dem Kunden vereinbarten Qualitätsanforderungen erfüllen (vgl. VDA – Verband der Automobilindustrie e. V. 2004, S. 25). In der RB Elektronik GmbH, einem Fertigungsstandort des Unternehmensbereichs Kraftfahrzeugtechnik, wird die Bemusterung über einen Erstausrüster-Vertrieb (EA-Vertrieb) der RB GmbH von einem Kunden angestoßen. Hierbei bekommen alle an der Bemusterung beteiligten Abteilungen eine so genannte Erstmuster-Checkliste (EM-Checkliste) per E-Mail zugesendet, welche u. a. die Festlegungen über Art und Umfang des vereinbarten Bemusterungsverfahrens enthält.

Gegenwärtig existieren diverse Varianten dieser EM-Checkliste, die sich sowohl inhaltlich als auch vom Layout her unterscheiden. Zudem sind in vielen EM-Checklisten Informationen enthalten, die bei der Bemusterung nicht benötigt werden. Andere Listen bieten hingegen nicht die Möglichkeit, alle notwendigen Informationen in übersichtlicher Form zu erfassen. Ein weiteres Manko besteht darin, dass die EM-Checklisten im EA-Vertrieb und den Abteilungen des Werks redundant in elektronischer und teilweise zusätzlich in ausgedruckter Form vorgehalten werden. Ein schneller Zugriff auf eine benötigte EM-Checkliste ist derzeit nicht gewährleistet, da keine komfortable Suchfunktion zur Verfügung steht.

Aufgrund der zuvor angesprochenen Probleme ist die Aufgabenstellung für diese Diplomarbeit entstanden, den aktuellen Ablauf der Bemusterung in der RB Elektronik GmbH zu analysieren und ein webbasiertes Informationssystem zu entwickeln. Mit diesem sollen alle für die Bemusterung benötigten Informationen in einheitlicher Form über das Intranet erfasst und zentral in einer Datenbank verwaltet werden können. Ziel ist es die Qualität des Bemusterungsablaufs zu verbessern, was nicht nur der RB Elektronik GmbH und dem EA-Vertrieb, sondern letztlich auch den Kunden zugute kommt.

Für die Implementierung der Webanwendung soll die serverseitige Technologie Active Server Pages .NET (ASP.NET) verwendet werden. Das Websystem ist dabei so zu konzipieren und umzusetzen, dass dieses auch an anderen Fertigungsstandorten der RB GmbH eingesetzt werden kann.

Die vorliegende Arbeit ist in acht Kapitel gegliedert. Im 2. Kapitel werden zunächst die theoretischen Grundlagen vermittelt, wobei die Begriffe Qualitätsmanagement und Qualitätsmanagementsystem (QM-System) erläutert werden sowie die Bedeutung der Bemusterung innerhalb des Qualitätsmanagements ausführlicher herausgearbeitet wird.

Das anschließende 3. Kapitel enthält die Ist-Analyse, welche neben der detaillierten Beschreibung der gegenwärtigen Strukturen und Abläufe der Bemusterung in der RB Elektronik GmbH als Ergebnis Schwachstellen des bestehenden Systems aufzeigt.

In Kapitel 4 wird das fachliche Soll-Konzept erarbeitet. Hierbei werden in Abstimmung mit dem Auftraggeber und den zukünftigen Nutzern die Anforderungen an das zu entwickelnde webbasierte Informationssystem festgelegt sowie die Datenbasis des Systems spezifiziert. Den Abschluss bildet die Vorstellung eines möglichen Soll-Zustandes für den Bemusterungsablauf.

Anschließend beschäftigt sich das 5. Kapitel mit dem Entwurf des Websystems, wobei u. a. die Systemarchitektur festgelegt wird sowie der Entwurf der relationalen Datenbank, der Applikation und der Benutzeroberfläche beschrieben werden.

Nach der Konzeption und dem Entwurf zeigt das 6. Kapitel die praktische Realisierung des Websystems. Hierbei wird anhand von Programmcodebeispielen und Screenshots der Benutzeroberfläche auf die umgesetzte Funktionalität sowie zentrale Bestandteile des webbasierten Informationssystems eingegangen.

Bevor in Kapitel 8 die Diplomarbeit zusammengefasst und ein Ausblick auf mögliche zukünftige Weiterentwicklungen gegeben wird, erfolgt im 7. Kapitel eine Bewertung der entwickelten Lösung anhand der im Rahmen der Soll-Konzeption formulierten Anforderungen.

2 Grundlagen

Organisationen müssen heute eine Vielzahl von Anforderungen erfüllen, um ihre Wettbewerbsfähigkeit zu sichern. Die Wettbewerbskomponente Qualität ist dabei neben den traditionellen Wettbewerbsfaktoren Kosten und Zeit von entscheidender Bedeutung. So werden langfristig nur die Organisationen Erfolg haben, denen es gelingt, die Bedürfnisse und Ansprüche ihrer Kunden schnell und kostengünstig in Produkte umzusetzen. Zur Bewältigung dieser Herausforderung ist die Einführung und Aufrechterhaltung eines effizienten QM-Systems in einer Organisation notwendig (vgl. Pfeifer 2001, S. XXV). Darüber hinaus ist, aufgrund immer kürzerer Entwicklungszeiten und der Zunahme von Outsourcing, die intensive Zusammenarbeit zwischen Organisationen und ihren Lieferanten von entscheidender Bedeutung (vgl. VDA – Verband der Automobilindustrie e. V. 2004, S. 13).

In den folgenden Abschnitten sollen die Begriffe Qualitätsmanagement und QM-System erläutert werden. Dabei wird insbesondere auf die Normreihe DIN EN ISO 9000 sowie die branchenspezifischen Normen der Automobilindustrie eingegangen. Im letzten Abschnitt wird die Bemusterung von Produkten zur Sicherung der Qualität von Lieferungen ausführlicher betrachtet.

2.1 Qualitätsmanagement und Qualitätssicherung

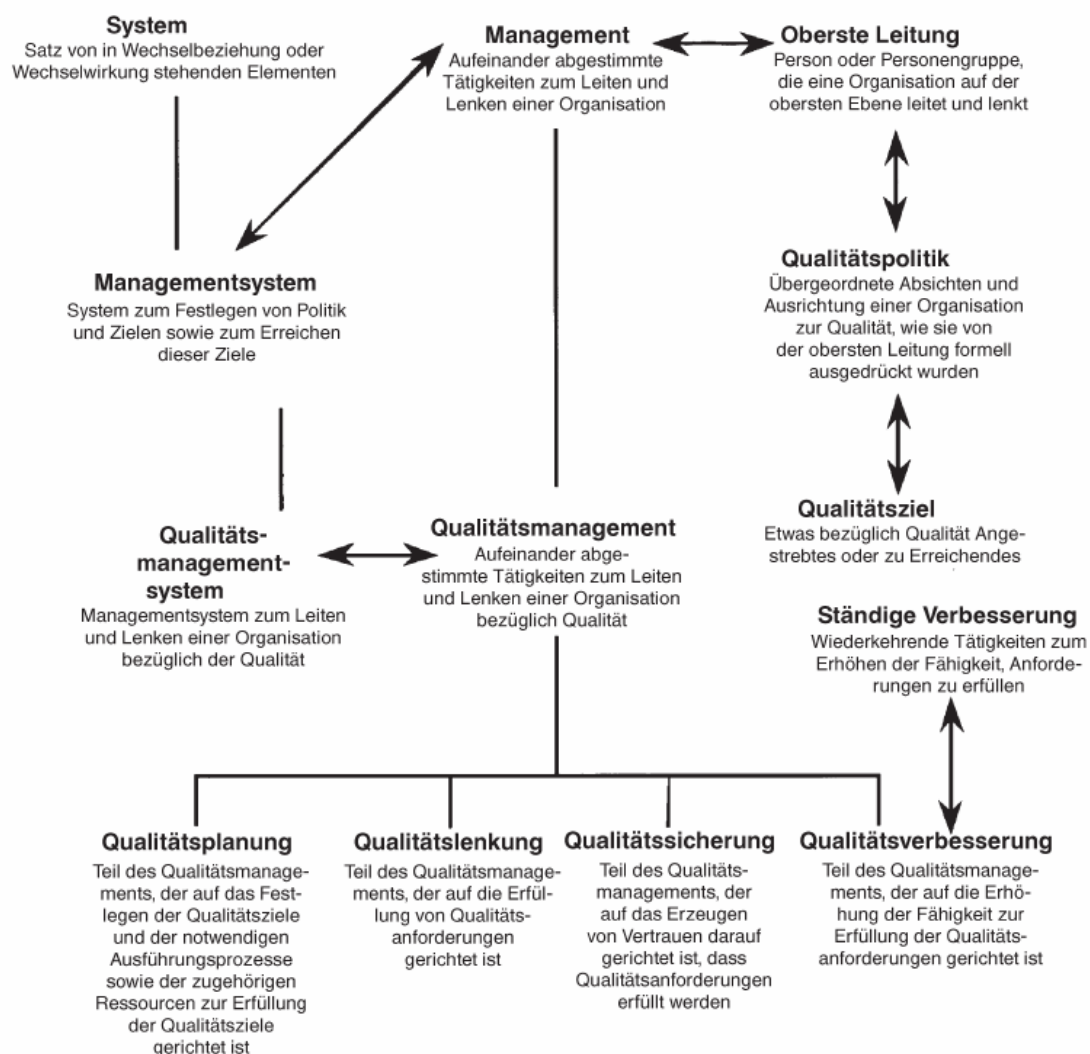
Die Begriffe Qualitätsmanagement und Qualitätssicherung sind schon vor längerer Zeit entstanden. So wurde die Benennung Qualitätssicherung 1974 durch die Deutsche Gesellschaft für Qualität e. V. (DGQ) festgelegt und galt seitdem im gesamten deutschen Sprachraum. International wurden wahlweise die Benennungen quality assurance, quality control und quality management verwendet. Durch einen Beschluss der International Organization for Standardization (ISO) wurde 1990 quality management als Oberbegriff definiert. Die Einführung dieser Änderung erfolgte international durch geltende Normen, für den deutschen Sprachraum jedoch erst 1994 (vgl. Geiger/Kotte 2005, S. 94 f.).

In der internationalen Norm DIN EN ISO 9000:2000¹ wird Qualitätsmanagement als die Gesamtheit aller qualitätsbezogenen und aufeinander abgestimmten Tätigkeiten zur Leitung und Lenkung einer Organisation bezüglich Qualität² definiert (s. Abb. 2.1). Dabei

¹ Die Abkürzung DIN steht für Deutsches Institut für Normierung e. V., das EN für Europäische Norm.

² Der Qualitätsbegriff lässt sich auf das lateinische Wort qualitas zurückführen, was mit Beschaffenheit eines Gegenstandes übersetzt werden kann (vgl. Kamiske/Brauer 2005, S. 174). Die Norm DIN EN ISO 9000:2000 definiert Qualität als „Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt“ (DIN – Deutsches Institut für Normung e. V. 2000, S. 18). Qualität wird demnach als messbare

beinhaltet die Leitung und Lenkung üblicherweise die Festlegung der Qualitätspolitik und der Qualitätsziele (vgl. DIN – Deutsches Institut für Normung e. V. 2000, S. 21), wobei die Qualitätspolitik die Ziele und Absichten der obersten Organisationsleitung ausdrückt. Die Festlegungen der Qualitätspolitik werden durch die Qualitätsplanung, Qualitätslenkung, Qualitätssicherung und Qualitätsverbesserung zur Ausführung gebracht (vgl. Kamiske/Brauer 2005, S. 207). Ziel des Qualitätsmanagements ist die Sicherstellung einer kontinuierlichen Produkt- und Leistungsverbesserung sowie einer optimalen Kundenzufriedenheit (vgl. DGQ – Deutsche Gesellschaft für Qualität e. V. 2000, S. 11).



Quelle: In Anlehnung an DIN – Deutsches Institut für Normung e. V. 2000, S. 37.

Abb. 2.1: Managementbezogene Begriffe

Differenz zwischen realisierter und geforderter Beschaffenheit eines Produkts oder einer Leistung verstanden und bedeutet aus Kundensicht die Erfüllung der kundenspezifischen Anforderungen (vgl. Kamiske/Brauer 2005, S. 170).

Die Norm DIN EN ISO 9000:2000 stellt acht Grundsätze des Qualitätsmanagements auf, welche von der obersten Leitung benutzt werden können, um die Leistungsfähigkeit der Organisation zu verbessern. Im Folgenden werden diese Grundsätze kurz aufgeführt (vgl. DIN – Deutsches Institut für Normung e. V. 2000, S. 6 f.):

a) Kundenorientierung:

Organisationen hängen von ihren Kunden ab und sollten daher aktuelle und zukünftige Erfordernisse der Kunden verstehen, ihre Anforderungen erfüllen und danach streben, ihre Erwartungen zu übertreffen.

b) Führung:

Das Management sollte ein internes Umfeld schaffen und erhalten, in dem sich Personen voll und ganz für die Erreichung der Organisationsziele einsetzen können.

c) Einbeziehung der Personen:

Die Fähigkeiten der einzelnen Mitarbeiter sollten durch ihre vollständige Einbeziehung zum Vorteil der Organisation genutzt werden.

d) Prozessorientierter Ansatz:

Ein erwünschtes Ergebnis wird effizienter erreicht, wenn die Tätigkeiten und dazugehörigen Ressourcen als Prozess³ geleitet und gelenkt werden.

e) Systemorientierter Managementansatz:

Das Identifizieren, Verstehen, Leiten und Lenken von miteinander in Wechselbeziehung stehenden Prozessen als System tragen zur Wirksamkeit und Effizienz der Organisation beim Erreichen ihrer Ziele bei.

f) Ständige Verbesserung:

Die ständige Verbesserung der Gesamtleistung der Organisation sollte ein permanentes Ziel der Organisation sein.

g) Sachbezogener Ansatz zur Entscheidungsfindung:

Wirksame Entscheidungen basieren auf der Analyse von Daten und Informationen.

³ Ein Prozess stellt eine zielgerichtete, inhaltlich abgeschlossene, zeitliche und sachlogische Abfolge von Funktionen dar, die zur Bearbeitung eines betriebswirtschaftlichen Objekts ausgeführt werden (vgl. Kugeler 2000, zit. nach Mertens 2001, S. 386 f.).

h) Lieferantenbeziehungen zum gegenseitigen Nutzen:

Eine Organisation und deren Lieferanten sind voneinander abhängig. Beziehungen zum gegenseitigen Nutzen steigern daher die Wertschöpfungsfähigkeit beider Seiten.

2.2 Qualitätsmanagementsystem

Unter einem QM-System wird ein System zum Festlegen von Qualitätspolitik und Qualitätszielen sowie zum Erreichen dieser Ziele verstanden (vgl. DIN – Deutsches Institut für Normung e. V. 2000, S. 20). Ein QM-System beschreibt die Gesamtheit der erforderlichen Verfahren, Prozesse, Zuständigkeiten, Mittel bzw. Ressourcen, die zur Sicherstellung der Qualität benötigt werden (vgl. Pfeifer 2001, S. 50). Dabei bezieht es die gesamte Organisation, einschließlich der Beziehungen zu ihrem Umfeld, ein und bildet den strukturellen Rahmen zur systematischen Umsetzung des Qualitätsmanagements. Da Aufbau und Umfang eines QM-Systems u. a. von den individuellen Zielsetzungen, Festlegungen und der Organisationsgröße abhängen, weisen QM-Systeme organisationsspezifische Unterschiede auf (vgl. Kamiske/Brauer 2005, S. 210).

2.2.1 Die Normreihe DIN EN ISO 9000:2000

Die Normreihe DIN EN ISO 9000 ist eine branchenunabhängige, weltweit anerkannte Rahmenempfehlung, die entwickelt wurde, um Organisationen jeder Art und Größe beim Verwirklichen von QM-Systemen und beim Arbeiten mit diesen zu helfen (vgl. DIN – Deutsches Institut für Normung e. V. 2000, S. 6). Sie ist 1987 entstanden, wurde 1994 und 2000 überarbeitet und enthält in der aktuellen Version die folgenden Kernnormen (vgl. Kamiske/Brauer 2005, S. 67 f. und Pfeifer 2001, S. 69):

- DIN EN ISO 9000:2000 – QM-Systeme / Grundlagen und Begriffe

Diese Norm dient der Unterstützung von Organisationen bei der Einführung und dem Umgang mit dem QM-System und ist als Einführung in das Thema QM-Systeme zu verstehen. Dazu beschreibt sie die Grundlagen von QM-Systemen und erklärt Begriffe aus dem Bereich Qualität und Qualitätsmanagement.

- DIN EN ISO 9001:2000 – QM-Systeme / Anforderungen

Diese Norm legt die Forderungen an ein QM-System fest und beinhaltet konkrete Hinweise, wie ein solches System normkonform aufzubauen und weiterzu-

entwickeln ist. Sie bildet die Grundlage für die Zertifizierung⁴ eines QM-Systems.

- DIN EN ISO 9004:2000 – QM-Systeme / Leitfaden zur Leistungsverbesserung

Diese Norm baut auf den Grundsätzen der DIN EN ISO 9001:2000 auf und stellt Empfehlungen bzw. Anregungen zur Einführung und Verbesserung von QM-Systemen bereit. Sie dient daher als Ergänzung bzw. Weiterführung und richtet sich an Organisationen, die über die Forderungen der DIN EN ISO 9001:2000 hinausgehen wollen.

2.2.2 QM-Systemforderungen in der Automobilindustrie

Die Unternehmen der Automobilindustrie haben maßgeblich an der Schaffung und Weiterentwicklung der branchenunabhängigen Normreihe DIN EN ISO 9000 mitgewirkt, jedoch schnell erkannt, dass die Forderungen dieser Normen für ein einheitliches Qualitätsniveau innerhalb der Branche – mit Blick auf die Zulieferer – noch nicht spezifisch genug waren. Zahlreiche Automobilhersteller formulierten daher ihre Forderungen an die QM-Systeme ihrer Lieferanten in eigenen Forderungskatalogen. Für deutsche Zulieferunternehmen sind hierbei die spezifischen Forderungen der US-amerikanischen Automobilproduzenten (QS-9000) und die des deutschen Verbandes der Automobilindustrie e. V. (VDA) von besonderer Bedeutung (vgl. Pfeifer 2001, S. 72).

VDA

Der VDA beschreibt in insgesamt 13 Bänden alle wesentlichen Definitionen, Regelungen und Forderungen zum Qualitätsmanagement für die Lieferanten der deutschen Automobilindustrie (vgl. Pfeifer 2001, S. 73). Der VDA Bd. 6, Teil 1 (kurz: VDA Bd. 6.1) gilt hierbei als das zentrale Regelwerk. Er baut auf der DIN EN ISO 9001 in der Fassung von 1994 auf und stellt eine Empfehlung zur Einführung, Aufrechterhaltung und Bewertung von QM-Systemen dar. Inhaltlich verlangt der VDA Bd. 6.1, dass eine Beurteilung des QM-Systems der Produzenten erfolgen muss, bevor diese als Lieferanten festgelegt werden können (vgl. Kamiske/Brauer 2005, S. 351 ff.).⁵ Darüber hinaus enthält die VDA-Schriftreihe Leitfäden für eine einheitliche

⁴ Die Zertifizierung ist der Vorgang des Nachweisens der Wirksamkeit und Funktionsfähigkeit eines Managementsystems im Unternehmen und wird durch unabhängige Auditoren einer akkreditierten Zertifizierungsgesellschaft durchgeführt (vgl. Kamiske/Brauer 2005, S. 377).

⁵ Die Beurteilung des QM-Systems kann bspw. durch ein Systemaudit seitens des Lieferanten oder durch eine Zertifizierung erfolgen (vgl. Kamiske/Brauer 2005, S. 353).

Abwicklung automobilspezifischer Prozesse (z. B. Lieferantenauswahl oder Bemusterung⁶) (vgl. Pfeifer 2001, S. 73).

QS-9000

Die Richtlinie QS-9000 wurde von den US-amerikanischen Automobilherstellern Chrysler, Ford und General Motors („Big Three“) entwickelt. Sie beinhaltet alle Forderungen der DIN EN ISO 9001:1994 und ergänzt diese um spezifische Forderungen und Empfehlungen für die Entwicklung eines QM-Systems nach den Vorstellungen der drei Automobilproduzenten sowie einiger US-amerikanischer Nutzfahrzeughersteller (vgl. Kamiske/Brauer 2005, S. 166 f.).⁷ Ähnlich der Vielzahl an VDA-Bänden setzt sich die Richtlinie QS-9000 aus insgesamt sieben Dokumenten zusammen, welche analog zur Definition möglichst einheitlicher Verfahren dienen. Hervorzuheben sind hierbei neben dem Handbuch „QS-9000 – Quality System Requirements“ (der QS-9000 im engeren Sinne) die Referenzhandbücher „Advanced Product Quality Planning“ (APQP)⁸ und „Production Part Approval Process“ (PPAP)⁹ (vgl. Pfeifer 2001, S. 73 f.).

Als weltweite Harmonisierung der branchenbezogenen Richtlinien gilt die Technische Spezifikation (TS) 16949 (vgl. Pfeifer 2001, S. 73). Sie ist ein offizielles ISO-Dokument und wurde in Zusammenarbeit mit der International Automotive Task Force (IATF), einer international besetzten Arbeitsgruppe mit Vertretern von Herstellern und Verbänden der Automobilindustrie erarbeitet und im März 2002 veröffentlicht. Die ISO/TS 16949:2002 basiert auf der DIN EN ISO 9001:2000 und vereinigt alle weltweit existierenden und veröffentlichten Qualitätsnormen bzw. Forderungen der internationalen Automobilindustrie an ein QM-System. Sie gilt gleichermaßen für die Automobilproduzenten (Original Equipment Manufacturer (OEM), Erstausrüster) sowie deren Lieferanten und wird von allen Herstellern weltweit anerkannt. Die ISO/TS 16949:2002 bildet so eine einheitliche Grundlage für die Zertifizierung von QM-Systemen und soll damit die häufig vorgekommenen Mehrfachzertifizierungen in europäischen Ländern (z. B. nach VDA Bd. 6.1) und den USA (z. B. nach QS-9000) verhindern (vgl. Kamiske/Brauer 2005, S. 90 f.).¹⁰

⁶ Auf die Bemusterung nach VDA Bd. 2 wird im nächsten Abschnitt ausführlicher eingegangen.

⁷ Seit Inkrafttreten der QS-9000 müssen alle Lieferanten der „Big Three“ die QS-9000-Forderungen in ihr QM-System implementieren (vgl. Kamiske/Brauer 2005, S. 166).

⁸ Das Handbuch APQP beschreibt den Ablauf eines Produkt- und Prozessentwicklungsprozesses in der Automobilindustrie (vgl. Pfeifer 2001, S. 74).

⁹ Auf das Produktionsteil-Freigabeverfahren wird im nächsten Abschnitt ausführlicher eingegangen.

¹⁰ QM-Systeme und ihre Zertifizierungen nach DIN EN ISO 9001:2000, VDA Bd. 6.1 oder QS-9000 behalten dabei jeweils ihre Gültigkeit, da sie mit der ISO/TS 16 949 harmonisieren. Durch die Zertifizierung nach ISO/TS 16 949:2002 erfüllt eine Organisation die Forderungen der DIN EN ISO 9001:2000 und gleichzeitig die branchenspezifischen Forderungen der Automobilindustrie (vgl. Kamiske/Brauer 2005, S. 91).

Darüber hinaus wird die ISO/TS 16949:2002 den Qualitätsstandard QS-9000 vollständig ersetzen, welcher zum 14. Dezember 2006 seine Gültigkeit verliert. Viele Organisationen haben daher bereits ihr Qualitätsmanagement auf dieses neue Regelwerk umgestellt (vgl. DQS GmbH 2006). Die Referenzhandbücher zur QS-9000, also APQP, PPAP etc., haben jedoch weiter ihre Gültigkeit und sind zur ISO/TS 16949 mitgeltend (vgl. AIAG – Automotive Industry Action Group 2006, S. i).

2.3 Bemusterung von Produkten

Zur Sicherung der Qualität von Lieferungen fordern die Automobilhersteller die Einhaltung ihrer Normen. Dabei werden insbesondere im VDA Bd. 2 und im Referenzhandbuch PPAP übergreifende Abläufe und Verfahren dargestellt, welche die Reibungsverluste an den Nahtstellen zwischen Lieferanten und Kunden durch eine enge partnerschaftliche Zusammenarbeit minimieren sollen.¹¹ Ein Beispiel für ein solches Verfahren ist die Bemusterung.

2.3.1 Zweck und Anwendungsbereich

Die Bemusterung bzw. das Produktionsteil-Freigabeverfahren soll vor Serienbeginn und bei Änderungen den Nachweis erbringen, dass die in Konstruktionszeichnungen und Spezifikationen¹² vereinbarten Qualitätsanforderungen erfüllt werden (vgl. VDA – Verband der Automobilindustrie e. V. 2004, S. 25) und die Fertigung des Lieferanten in der Lage ist, Produkte herzustellen, die diese Forderungen während eines tatsächlichen Produktionslaufs (Serienlauf) mit den vorgegebenen Produktionszahlen erfüllen (vgl. AIAG – Automotive Industry Action Group 2006, S. 1).¹³ Die Bemusterung umfasst dabei den gesamten Prozess von der Festlegung technischer und kaufmännischer Rahmenbedingungen zwischen den Vertragspartnern bis zur Freigabe der Serienfertigung/-lieferung durch den Kunden.

In der ISO/TS 16949:2002 ist festgelegt, dass ein Lieferant ein vom Kunden anerkanntes Bemusterungsverfahren einhalten muss (vgl. DIN – Deutsches Institut für Normung e. V. 2002, S. 37). Der Lieferant hat somit entweder das so genannte PPAP-Verfahren

¹¹ Im Folgenden werden zur Vereinfachung für die Lieferbeziehungen zwischen nur zwei Partnern die Begriffe Kunde und Lieferant verwendet.

¹² Als Spezifikationen werden Dokumente bezeichnet, die Anforderungen enthalten (vgl. DIN – Deutsches Institut für Normung e. V. 2000, S. 14).

¹³ Im VDA Bd. 2 wird der Begriff Bemusterung verwendet, wohingegen im Referenzhandbuch PPAP vom Produktionsteil-Freigabeverfahren die Rede ist. Im Rahmen der Diplomarbeit wird nachfolgend der Begriff Bemusterung verwendet.

für Kunden mit Forderungen nach QS-9000 bzw. ISO/TS 16949:2002 oder das Produktionsprozess- und Produktfreigabe (PPF)-Verfahren für Kunden mit VDA-Forderungen durchzuführen.

2.3.2 Grundlegende Forderungen

Wie zuvor erwähnt, erfordern neue oder geänderte Produkte und Verfahren immer eine Freigabe gemäß VDA Bd. 2 oder PPAP. Die Notwendigkeit und der Umfang des Bemusterungsverfahrens sind von den Vertragspartnern im Rahmen der Vertragsprüfung im Voraus schriftlich festzulegen. Dies betrifft z. B. die Anzahl der erforderlichen Erstmuster sowie den verbindlichen Termin zur Vorstellung der Erstmuster. Als Erstmuster werden hierbei Produkte und Materialien bezeichnet, die vollständig mit serienmäßigen Betriebsmitteln (Betriebseinrichtungen und Werkzeugen) unter serienmäßigen Bedingungen hergestellt wurden. Im Folgenden werden einige grundlegende Forderungen, die sowohl für PPAP- als auch PPF-Verfahren gelten, aufgeführt (vgl. VDA – Verband der Automobilindustrie e. V. 2004, S. 27 ff. und AIAG – Automotive Industry Action Group 2006, S. 2 ff.).¹⁴

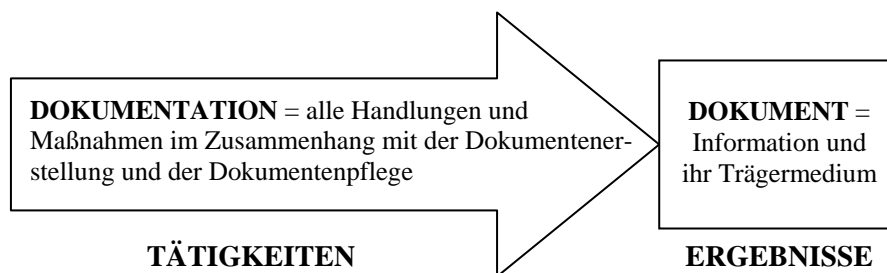
- Die Erstmuster müssen einem repräsentativen Produktionslauf entstammen und als solche besonders gekennzeichnet sein.
- Die Erstmuster sind einer Erstmusterprüfung (EM-Prüfung) zu unterziehen, wobei die Musterteile gegen die im Voraus vereinbarten Forderungen geprüft, bewertet und die Ergebnisse dokumentiert werden.
- Der Lieferant hat im Rahmen der Bemusterung eine vollständige Dokumentation zu erstellen, damit der Kunde alle Untersuchungen, Messungen und Prüfungen am Produkt nachvollziehen kann.
- Die Erstmuster sowie die kompletten vereinbarten Bemusterungsunterlagen sind vom Lieferanten in einer eigenen, besonders gekennzeichneten Verpackung an den Kunden zu übermitteln.
- Der Kunde hat schließlich die Erstmuster sowie die dazugehörenden Bemusterungsunterlagen zu bewerten.¹⁵

¹⁴ Im Rahmen dieser Diplomarbeit kann nur ein kurzer Überblick gegeben werden. Für weiterführende Informationen sei an dieser Stelle auf den VDA Bd. 2 und das Referenzhandbuch PPAP verwiesen.

¹⁵ Die kundenseitige positive Bewertung hat die Freigabe der Serienfertigung zur Folge.

2.3.3 Dokumentation und Dokumente

Bei der Bemusterung sind die Dokumentation und Dokumente von großer Bedeutung. Der Begriff Dokumentation kann dabei in unterschiedlichster Weise benutzt werden. Im deutschen Dokumentationswesen werden unter Dokumentation die für die Fachinformation¹⁶ wesentlichen Tätigkeiten verstanden, die das systematische Sammeln und Auswählen, formale Erfassen, inhaltliche Auswerten und Speichern von Dokumenten umfassen, um sie zum Zweck der gezielten Information rasch und treffsicher auffinden zu können. Hierbei ist der wesentliche Begriffsinhalt zu beachten, nämlich dass es sich bei der Dokumentation um Tätigkeiten handelt. Der essentielle Unterschied zwischen Tätigkeiten und ihrem Ergebnis ist in Abb. 2.2 dargestellt. In der Umgangssprache und auch in Fachsprachen werden spezielle Ergebnisse einer Dokumentation ebenfalls mit diesem Namen bezeichnet. So wird unter Dokumentation (z. B. in DIN EN ISO 9001:2000 und 9004:2000) auch die Gesamtheit der Dokumente zu einer betrachteten Einheit verstanden (vgl. Geiger/Kotte 2005, S. 473 f.).



Quelle: Geiger/Kotte 2005, S. 473.

Abb. 2.2: Tätigkeiten und Ergebnisse bei der Dokumentation

Für die Dokumentation der Bemusterung sind generell die im VDA Bd. 2 bzw. im Referenzhandbuch PPAP geforderten Formulare zu verwenden. Es können jedoch auch an diese Schriften angepasste unternehmensspezifische Formulare zum Einsatz kommen. Grundsätzlich wird somit die Art der Dokumentation durch das vereinbarte Bemusterungsverfahren bestimmt. Der Umfang der Dokumente, welche dem Kunden zur Freigabe zu übermitteln sind, wird dagegen durch die im Rahmen der Vertragsprüfung vereinbarte Vorlagestufe festgelegt.¹⁷ Auf eine Aufzählung der durch die jeweilige Vorlagestufe geforderten Dokumente wird an dieser Stelle verzichtet. Im späteren Verlauf der Arbeit sollen jedoch einige dieser Dokumente aufgeführt werden.

¹⁶ Als Fachinformation wird die Gesamtheit des auf einem Datenträger gespeicherten und bereitgestellten Wissens, das zum Zweck der Erfüllung fachlicher Aufgaben erschlossen, aufbereitet und bereitgestellt wird, bezeichnet (vgl. Geiger/Kotte 2005, S. 474).

¹⁷ Während es beim PPAP-Verfahren fünf Vorlagestufen gibt, sind beim PPF-Verfahren nur drei Vorlagestufen möglich.

3 Ist-Analyse

Am Anfang eines Entwicklungsprozesses steht in der Regel die Ist-Analyse, welche die fachliche Basis für das Entwicklungsvorhaben definiert (vgl. Burghardt 2002, S. 302).

Die Zielsetzung der folgenden Ist-Analyse besteht darin, eine detaillierte Beschreibung der gegenwärtigen Strukturen und Abläufe der Bemusterung in der RB Elektronik GmbH zu erhalten. Bevor jedoch mit der eigentlichen Erhebung der Ist-Situation begonnen wird, werden kurz einige Methoden der Informationsgewinnung vorgestellt und auf ihre Eignung für die geplante Ist-Analyse hin untersucht. Anschließend gilt es, das betrachtete System zu identifizieren und in angemessener Weise von seiner Umgebung abzugrenzen. Neben der ausführlichen Dokumentation des Ist-Zustandes soll die Ist-Analyse als Ergebnis Schwachstellen des bestehenden Systems aufzeigen.

3.1 Allgemeine Vorüberlegungen

Am Anfang eines Entwicklungsvorhabens liegen meist keine bzw. nur geringe Informationen über das zu analysierende System vor. Damit eine umfassende Darstellung des Ist-Zustandes gewährleistet werden kann, ist es von entscheidender Bedeutung zuerst eine gute Informationsgrundlage zu schaffen. Eine der wichtigsten Informationsquellen sind hierbei die Mitarbeiter einer Organisation, da diese die Arbeitsabläufe kennen und über Erfahrungen mit Dokumenten, Daten und Arbeitsmitteln verfügen. Des Weiteren können Dokumente unterschiedlichster Art wichtige Informationen über das zu analysierende System enthalten (vgl. Häuslein 2004, S. 47). Für die Erfassung der Informationen hat sich eine Vielzahl von Methoden herausgebildet, die kurz vorgestellt werden sollen.

Interview

Unter einem Interview wird die direkte mündliche Befragung von Personen verstanden. Diese Befragung kann entweder auf einem zuvor festgelegten Fragenkatalog basieren (standardisiertes Interview) oder aber auf Fragen, die im Verlauf des Interviews entwickelt werden (freies Interview). Darüber hinaus können Interviews nicht nur in Form von Einzelgesprächen, sondern auch als Gruppeninterviews durchgeführt werden (vgl. Häuslein 2004, S. 49). Allgemein gilt, dass Interviews immer gut vorbereitet sein sollten, sowohl hinsichtlich ihres Inhaltes als auch des zu befragenden Personenkreises (vgl. Burghardt 2002, S. 302).

Fragebogen

Bei der Fragebogen-Technik werden ausgewählte Fragen schriftlich festgelegt und den Wissensträgern zugänglich gemacht. Fragebögen sind demnach schriftliche Interviews, die verteilt, ohne direkten persönlichen Kontakt zum Ersteller bearbeitet, anschließend wieder eingesammelt und ausgewertet werden. Ein kritischer Aspekt bei dieser Art der Informationsgewinnung besteht darin, dass Missverständnisse und Fehlinterpretation kaum geklärt werden können, weshalb die Fragen mit großer Sorgfalt formuliert werden müssen (vgl. Häuslein 2004, S. 53 f.). Fragebögen werden meist dann eingesetzt, wenn die Anzahl der Interview-Partner zu groß oder eine örtliche Zusammenkunft nicht möglich ist (vgl. Burghardt 2002, S. 303).

Beobachtungen

Bei der Beobachtung wird ein bestimmter Sachverhalt durch sinnliche Wahrnehmung aufgenommen. Diese Methode wird besonders im Produktionsbereich eingesetzt, im administrativen Bereich kommt sie dagegen seltener zur Anwendung. Die Beobachtung kann als Dauerbeobachtung über einen festgelegten Zeitraum oder auch als so genanntes Multimomentverfahren zu einzelnen, zufälligen Zeitpunkten durchgeführt werden. Bei Letzterem werden die Ergebnisse der Einzelbeobachtungen dann statistisch ausgewertet. Eine weitere Form der Beobachtung ist die Selbstaufschreibung, bei welcher die Aufnahme des Sachverhalts durch den Wissensträger selbst erfolgt (vgl. Häuslein 2004, S. 58).

Dokumentenanalyse

Bei dieser Methode werden Informationen aus Dokumenten unterschiedlichster Art extrahiert. Die so genannte Inventur-Technik kommt dann zum Einsatz, wenn die Dokumente einen engen Bezug zum analysierenden System haben. Solche Dokumente sind bspw. Organisations- und Aufgabenpläne, Arbeits- und Verfahrensanweisungen oder Statistiken. Die Auswertung von Dokumenten, die außerhalb der Organisation vorliegen, wird als Quellenauswertung bezeichnet (vgl. Häuslein 2004, S. 59 ff.).

Die Eignung der vorgestellten Erhebungsmethoden ist je nach Art und Umfang des Untersuchungsgegenstandes unterschiedlich zu bewerten. Für dieses Entwicklungsvorhaben erfolgt die Informationserhebung durch eine Kombination verschiedener Methoden. Grundbaustein dieser Ist-Analyse sind Interviews (meist Einzelgespräche), die durch Dokumentenanalysen ergänzt werden. Zu Beginn dienen überwiegend freie Interviews dazu, einen guten Überblick über den zu erfassenden Systembereich zu er-

halten. Im weiteren Verlauf der Informationserhebung werden dann teilweise auch standardisierte Interviews durchgeführt. Mittels Inventur-Technik werden alle für den Untersuchungsgegenstand relevanten Dokumente ausgewertet.

3.2 Abgrenzung des Untersuchungsgegenstandes

Wie bereits erwähnt, soll im Folgenden der aktuelle Ablauf der Bemusterung in der RB Elektronik GmbH beschrieben und analysiert werden. Dieser beginnt mit der Erstmusterbestellung (EM-Bestellung), also der Festlegung der technischen und kaufmännischen Rahmenbedingungen. Er beinhaltet u. a. die Erstmusterbereitstellung (EM-Bereitstellung) und -prüfung sowie die Erstellung der Bemusterungsunterlagen und endet schließlich mit der Auslieferung der Erstmuster an den Kunden. Im Rahmen dieser Ist-Analyse soll speziell darauf eingegangen werden, wie die an der Bemusterung beteiligten Abteilungen über Art und Umfang des vereinbarten Bemusterungsverfahrens informiert werden. Hierbei steht besonders die so genannte EM-Checkliste als Mittel der Informationsversorgung im Vordergrund. Die einzelnen Arbeitsschritte bei der EM-Bereitstellung sowie der anschließenden Dokumentation werden nicht detailliert beschrieben.

3.3 Beschreibung und Darstellung des Ist-Zustandes

In diesem Abschnitt wird, basierend auf den in der Bestandsaufnahme gewonnenen Informationen, die aktuelle Ist-Situation dokumentiert. Zur Beschreibung der Prozesse können dabei unterschiedlichste Methoden zum Einsatz kommen. Im Folgenden werden die Aufbau- und Ablauforganisation zuerst informal in Textform beschrieben und anschließend mit Hilfe des ARIS-Toolsets¹⁸ modelliert. Für die Dokumentation der Prozesse wären z. B. auch Zustands-, Aktivitäts- und Flussdiagramme aus der Unified Modeling Language (UML)¹⁹, Netzpläne oder Petrinetze denkbar. Auf die EM-Checklisten, welche im Rahmen der Bemusterung eine wichtige Rolle spielen, wird während der Beschreibung der Ablauforganisation näher eingegangen.

¹⁸ ARIS steht für „Architektur integrierter Informationssysteme“ und ist ein von August-Wilhelm Scheer theoretisch entwickeltes Konzept zur Beschreibung (Modellierung) von Organisationen und computergestützten Informationssystemen (vom Fachkonzept bis zur Implementierung) (vgl. Scheer 2002, S. 1). Das Konzept wird in Form eines von der IDS Scheer AG entwickelten Softwareprodukts namens ARIS Toolset umgesetzt, welches zum weltweit führenden Prozessmodellierungstool geworden ist (vgl. Seidlmeier 2002, S. 12).

¹⁹ Die UML ist eine Familie grafischer Notationen, die insbesondere bei der Beschreibung und Entwicklung von objektorientierten Softwaresystemen eingesetzt wird (vgl. Fowler 2004, S. 19).

3.3.1 Aufbauorganisation

Die Aufbauorganisation bildet das hierarchische Gerüst einer Organisation und betrachtet die Aufgabenträger (Organisationseinheiten) und deren Beziehungen zueinander (vgl. Scheer 2002, S. 2 f.). Bevor die Ablauforganisation der RB Elektronik GmbH beschrieben und in Form eines Organigramms dargestellt wird, erfolgt eine kurze Einordnung in die RB GmbH.

Die RB Elektronik GmbH gehört wie bereits erwähnt zur RB GmbH, welche aus den drei großen Unternehmensbereichen Kraftfahrzeugtechnik, Industrietechnik sowie Gebrauchsgüter und Gebäudetechnik besteht. Der Unternehmensbereich Kraftfahrzeugtechnik unterteilt sich wiederum in die acht Geschäftsbereiche Benzinsysteme, Dieselsysteme, Chassis Systems Brakes, Chassis Systems Control, Energie- und Karoseriesysteme, Car Multimedia, Automobilelektronik und Automobiltechnik Handel. Jeder dieser Geschäftsbereiche besteht aus verschiedenen Einheiten, die den Bereichen Marketing & Verkauf, Entwicklung, Fertigung & Qualität, Kaufmännische Aufgaben & IT-Koordination zugeordnet sind. Darüber hinaus sind diverse Werke in die einzelnen Geschäftsbereiche eingeordnet.

Die RB Elektronik GmbH ist ein Werk aus dem Geschäftsbereich Automobilelektronik und produziert am Standort Salzgitter hauptsächlich elektronische Motorsteuergeräte für Diesel- und Benzin-Einspritzsysteme. Der Vertrieb der Produkte und Erstmuster erfolgt jedoch meist über die Verkaufshäuser des Geschäftsbereich Benzin- bzw. Dieselsysteme und nicht über die des Geschäftsbereichs Automobilelektronik. Da die RB Elektronik GmbH im Vordergrund dieser Untersuchung steht und die Aufbauorganisation der RB GmbH nicht dargestellt wird, ist im Folgenden allgemein vom EA-Vertrieb die Rede, der vereinfacht in einen technischen und kaufmännischen Verkaufsbereich unterteilt ist.

Für die Abwicklung der Bemusterung und die anschließende, termingerechte Zustellung an den Kunden ist in der RB Elektronik GmbH die Abteilung Produktqualität (QMM1) verantwortlich. Die Abteilung Materialwirtschaft (MAT) ist u. a. zuständig für Fertigungsaufträge und Arbeitspläne²⁰, die Abteilung Technische Funktionen (TEF) für die Erstellung der Stücklisten²¹. Die Bereitstellung und Prüfung der Erstmuster erfolgt durch die Abteilung Fertigung Steuergeräte (MSB/H). Im abgebildeten Organigramm (s. Abb. 3.1) sind auch die Bezeichnungen der anderen Abteilungen der RB Elektronik GmbH zu finden. Unterabteilungen, Stellen und interne Personen wurden nicht dargestellt, da z. B. die Namen der einzelnen Mitarbeiter für die im Folgenden beschriebenen

²⁰ Ein Arbeitsplan fasst eine Folge von auszuführenden Arbeitsvorgängen zusammen, die für die Herstellung eines Teils benötigt werden (vgl. Mertens 2001, S. 219).

²¹ Eine Stückliste ist eine strukturierte Anordnung von Teilen bzw. Baugruppen, die für die Herstellung eines anderen Teiles erforderlich sind.

Abläufe nicht bedeutend sind. Darüber hinaus sind alle Sachbearbeiter der QMM1 für die Bemusterung zuständig, weshalb die Darstellung der Abteilung hier ausreichend ist.

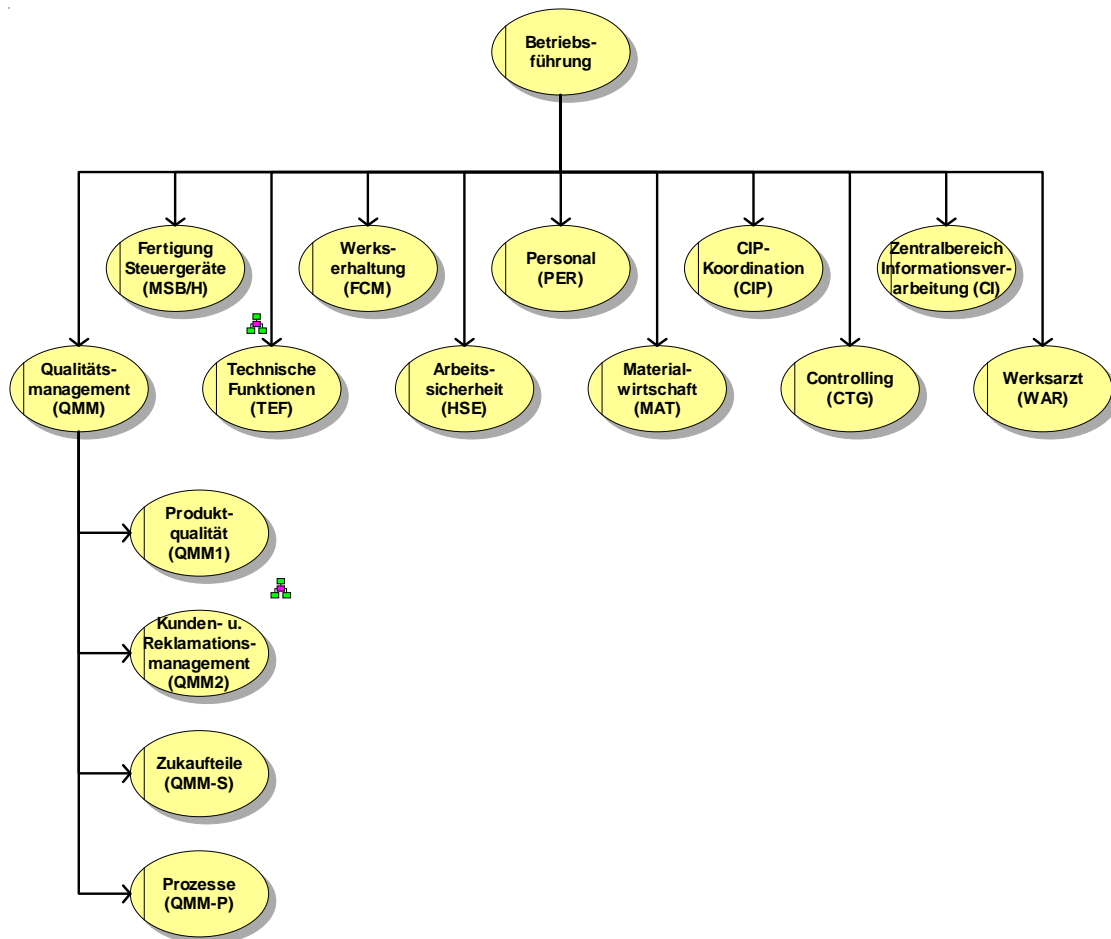


Abb. 3.1: Organigramm der RB Elektronik GmbH (Ist-Zustand)

3.3.2 Ablauforganisation

Die Ablauforganisation beschreibt das dynamische Verhalten von Vorgängen, die der Aufgabenerfüllung einer Organisation dienen (vgl. Scheer 2002, S. 3). Im Folgenden soll der Ablauf der Bemusterung informal beschrieben und mit Hilfe von erweiterten ereignisgesteuerten Prozessketten (eEPK)²² dargestellt werden.

²² Die ereignisgesteuerte Prozesskette (EPK) ist ein wesentlicher Bestandteil des theoretischen ARIS-Konzeptes und ist 1992 am Institut für Wirtschaftsinformatik der Universität des Saarlandes entstanden (vgl. Scheer 2002, S. 20). Es handelt sich um eine semiformale Methode, mit der Arbeitsprozesse grafisch dargestellt werden können. EPKs bestehen aus drei Grundelementen: Ereignisse, Funktionen und Verknüpfungsoperatoren. Die eEPK stellt eine erweiterte Form der Modellierungsmethode EPK dar und enthält zusätzlich Aussagen zu Input-/Output-Daten, Organisationsobjekten wie Organisati-

Die Bemusterung wird ausgelöst, wenn ein Kunde Erstmuster beim EA-Vertrieb der RB GmbH bestellt. Bei der Erfassung der EM-Bestellung ist von den Vertriebsmitarbeitern unter Verwendung von Microsoft (MS) Word bzw. MS Excel ein Dokument anzulegen, welches als EM-Checkliste bezeichnet wird. Gegenwärtig existieren verschiedene Varianten von EM-Checklisten, die je nach Kunden und Fertigungsstandort im EA-Vertrieb ausgewählt werden. Einige Dokumente sind inhaltlich etwas umfangreicher als andere, die grundlegenden Informationen sind jedoch in allen vorhanden. Das Layout der EM-Checklisten variiert hingegen meist stark. Abbildung 3.2 zeigt ein Beispiel für eine EM-Checkliste. Der Arbeitsablauf des Ausfüllens und Verteilens der EM-Checkliste ist nicht vereinheitlicht, sondern gestaltet sich vielmehr je nach Verkaufshaus und Geschäftsbereich unterschiedlich. In der Regel wird die EM-Checkliste jedoch nacheinander von zwei Vertriebsmitarbeitern aufgenommen.

Ein Mitarbeiter des technischen Verkaufsbereichs beginnt mit dem Ausfüllen des Dokuments (s. Abb. 3.3), wobei dieser die benötigten Informationen nach und nach zusammenträgt. Hierbei werden u. a. die technischen Merkmale in Abstimmung mit der Qualitätsmanagementabteilung des Kunden, Informationen zum Kunden (z. B. der Name), zum Erstmuster (z. B. die RB-Sachnummer²³ und die zu liefernde Menge) sowie die Vereinbarungen über Art und Umfang der bevorstehenden Bemusterung aufgenommen (s. Abb. 3.2). Anschließend versendet der Vertriebsmitarbeiter die fast vollständig ausgefüllte EM-Checkliste unter Verwendung von MS Outlook an einen Mitarbeiter des kaufmännischen Verkaufsbereichs, welcher einen Erstmuster-Auftrag (EM-Auftrag) im Anwendungssystem Erstausrüster-Vertriebs-Abwicklung (EVA)²⁴ anlegt. Die daraus resultierende EVA-Auftragsnummer wird mit weiteren ergänzenden Informationen, z. B. zum Kunden, in die EM-Checkliste eingetragen. Letztendlich wird die ausgefüllte EM-Checkliste per E-Mail an die, für die Bemusterung zuständigen Abteilungen der RB Elektronik GmbH sowie an den für das Internationale Materialdatensystem (IMDS)²⁵ zuständigen Beauftragten des Geschäftsbereiches Automobilelektronik der RB GmbH geschickt.

onseinheiten, Stellen oder Personen, Anwendungssystemen und Sachmitteln (vgl. Seidlmeier 2002, S. 71).

²³ Durch die RB-Sachnummer ist ein jedes Teil, in diesem Fall das Erstmuster, eindeutig identifizierbar.

²⁴ Auf das EVA-System kann in der RB Elektronik GmbH von der QMM1 und MSB/H nicht zugegriffen werden. Das System ist ausschließlich mit der MAT und TEF verbunden, wo der EM-Auftrag, sofern alle Mussfelder ausgefüllt sind, automatisch in SAP R/3 übernommen und geplant wird. In diesem Zusammenhang entstehen u. a. die Fertigungsaufträge, Arbeitspläne und Stücklisten. Die Ablaufstrukturen dieser Prozesse sind im Rahmen dieser Arbeit bewusst ausgeblendet worden, da die Informationsversorgung der QMM1 und MSB/H mittels der EM-Checkliste im Vordergrund steht.

²⁵ Das IMDS ist eine Datenbank, in der alle in einem Fahrzeug verwendeten Werkstoffe online archiviert und verwaltet werden (vgl. EDS Deutschland GmbH 2006).

Der IMDS-Beauftragte legt daraufhin für die angeforderten Erstmuster, unter Verwendung der EM-Checkliste, im IMDS ein so genanntes Materialdatenblatt (IMDS-Datenblatt) an (s. Abb. 3.4).

Checkliste - Erstmusterprüfbericht (EMPB)	
RB-Sachnr.:	<input type="text"/>
Version:	<input type="text"/>
vom:	<input type="text"/>
RG-TV-Bereich	
Name:	<input type="text"/>
Telefon:	<input type="text"/>
Abteilung:	
Fax:	<input type="text"/>
E-Mail:	<input type="text"/>
EVA-Auftrags-Nr.:	<input type="text"/> "1MU"-Aufträge
Datum:	<input type="text"/>
Kunde:	
Kundenkennzahl:	<input type="text"/>
Kundenauftragsnr.:	<input type="text"/>
Kunden-Werk:	<input type="text"/>
Kunden-Nr.:	<input type="text"/>
Kundenwunschtermin:	<input type="text"/>
Zugesagter Kundentermin:	<input type="text"/>
Grund für EMPB	
<input type="text"/>	
Neufreigabe <input type="checkbox"/> Fertigungsverlagerung <input type="checkbox"/> Technische Änderung <input type="checkbox"/> Sonstiges: <input type="text"/>	
Erzeugnis-Kurzbezeichnung:	<input type="text"/>
RB-Sachnr.:	<input type="text"/>
Kunden-Sachnummer:	<input type="text"/>
Zeichnung-RB:	<input type="text"/>
Änderungsindex/Gültigkeitsdatum:	<input type="text"/>
Mindest-Losgröße für die Nullserie (Erstmuster):	<input type="text"/>
Anzahl Erstmuster:	<input type="text"/>
Anzahl Referenzmuster (bei PPAP mind. 1 Stück):	<input type="text"/>
Preis:	<input type="text"/>
Bezahlung der Restmenge:	<input type="text"/>
Verpackung (Teile)	
Art: <input type="checkbox"/> Einweg (Styropor) <input type="checkbox"/> Mehrweg (KLT)	kundenspezifischer Behälter: <input type="text"/>
	kundenspezifischer Einsatz: <input type="text"/>
Palette	
Größe: <input type="checkbox"/> Euro-Maße <input type="checkbox"/> Iso-Maße	kundenspezifisch: <input type="text"/>
Art: <input type="checkbox"/> Einweg <input type="checkbox"/> Mehrweg	
Kennzeichnungen/kundenspezifische Vordrucke	
Teile: <input type="text"/>	Kundenspezifische Aufkleber: <input type="checkbox"/> Ja <input type="checkbox"/> Nein wenn "Ja", welche? <input type="text"/> im Werk vorhanden? <input type="checkbox"/> Ja <input type="checkbox"/> Nein wenn "Nein", wann im Werk: <input type="text"/> Sonstige Kundenforderungen für die Vorserie erforderlich? <input type="checkbox"/> Ja <input type="checkbox"/> Nein wenn "Ja", welche? <input type="text"/>
Verpackung: <input type="text"/>	
Palette: <input type="text"/>	
Lieferschein: <input type="text"/>	
Rechnung: <input type="text"/>	
Verkaufsbereich	
Abteilung:	
Name:	<input type="text"/>
Telefon:	<input type="text"/>
Fax:	<input type="text"/>
E-Mail:	<input type="text"/>
Zeichnung RB:	<input type="text"/>
Änderungsindex/Gültigkeitsdatum:	<input type="text"/>
Zeichnung Kunde:	<input type="text"/>
Änderungsindex/Gültigkeitsdatum:	<input type="text"/>
TKU RB:	<input type="text"/>
Änderungsindex/Gültigkeitsdatum:	<input type="text"/>
Kundenspezifikation:	<input type="text"/>
Änderungsindex/Gültigkeitsdatum:	<input type="text"/>
Durchzuführende Prüfungen:	<input type="text"/>
Vorgaben zur MAE,	<input type="text"/>
Änderungs-Schein,	<input type="text"/>
etc.	<input type="text"/>
Verfahren zur Abnahmeprüfung	
PPAP (nach QS9000): <input type="checkbox"/>	Ausgabe: <input type="text"/>
PPAP-Level:	<input type="text"/>
PPF (nach VDA): <input type="checkbox"/>	Ausgabe: <input type="text"/>
Vorlage-Stufe:	<input type="text"/>
Kundenspezifisch: <input type="checkbox"/> zusätzlich PPF erforderlich! Beschreibung <input type="text"/>	
<input type="checkbox"/> Funktionsprüfung <input type="checkbox"/> Maßprüfung <input type="checkbox"/> Material-Daten <input type="checkbox"/> Zuverlässigkeitsprüfung <input type="checkbox"/> Erprobung-Entwicklung	
Dokumentation	
Kundenformular: <input type="checkbox"/> im Werk vorhanden? <input type="checkbox"/> Ja <input type="checkbox"/> Nein wenn "Nein", wann im Werk <input type="text"/>	
VDA-Vordruck (RB): <input type="checkbox"/>	
PPAP-Vordruck: <input type="checkbox"/>	
<input type="checkbox"/> komplette Bemusterung <input type="checkbox"/> nur Änderung bemustern und Verweis auf Vorgänger-Erstmuster	

Abb. 3.2: Beispiel einer EM-Checkliste

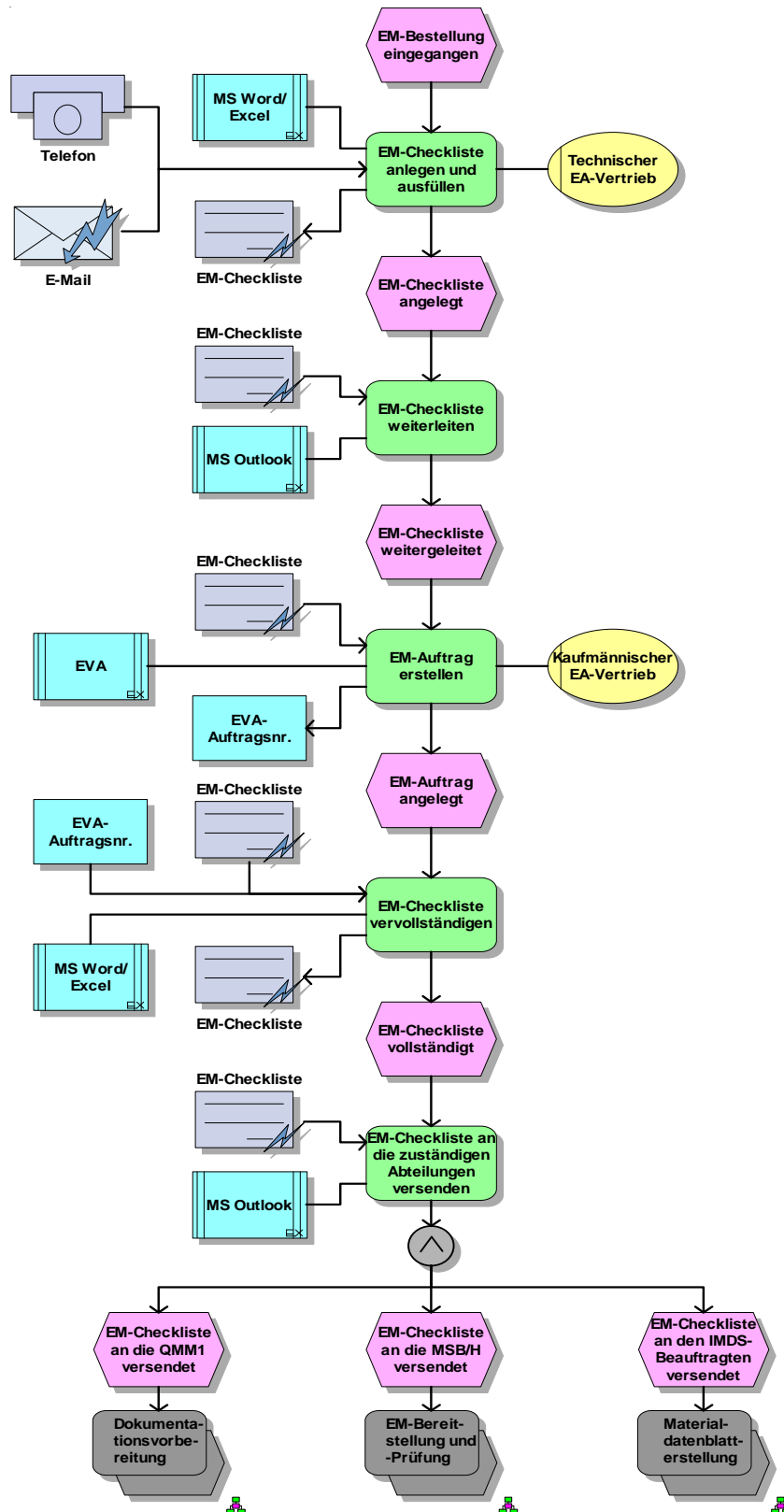


Abb. 3.3: eEPK zur Abwicklung der EM-Bestellung (Ist-Zustand)²⁶

²⁶ Organisationseinheiten, Anwendungssysteme, Dokumente etc. werden in den folgenden eEPKs nur abgebildet, wenn sie in dieser Kombination erstmalig auftreten. Auf die Modellierung wird aus Gründen der Übersichtlichkeit verzichtet, solange sich keine Änderungen im Prozessablauf ergeben.

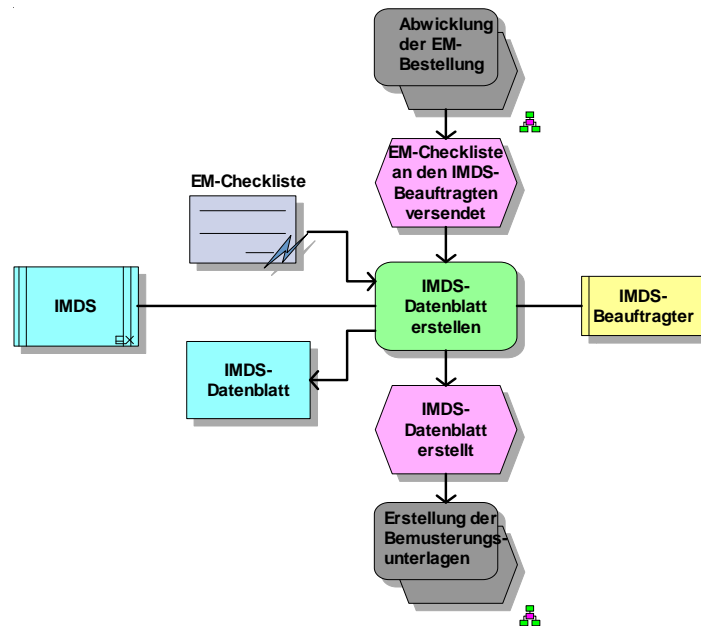


Abb. 3.4: eEPK zur Materialdatenblätterstellung (Ist-Zustand)

Wenn eine EM-Checkliste in der QMM1 eingegangen ist, wird zuerst überprüft, ob auch alle an der Bemusterung beteiligten Abteilungen und Stellen über die EM-Bestellung informiert worden sind (s. Abb. 3.5). Hat bspw. nur die QMM1 die EM-Checkliste bekommen, muss diese an die MSB/H und den IMDS-Beauftragten per E-Mail weitergeleitet werden. Sind alle Beteiligten informiert, wird die in MS Outlook vorliegende EM-Checkliste ausgedruckt. Von dieser werden dann ausgewählte Daten in eine so genannte Erstmusterübersichtsliste (EM-Übersichtsliste), welche wiederum als elektronisches Dokument vorliegt, übernommen. In dieser Liste sind alle Bemusterungen u. a. mit den RB-Sachnummern der Erstmuster, Terminen und den Namen der verantwortlichen Sachbearbeiter gespeichert. Nachdem die benötigten Daten übertragen wurden, wird die ausgedruckte EM-Checkliste anhand des vom Kunden gewünschten Zustellungstermins und des Namens des zuständigen Sachbearbeiters in einem physischen Ordner abgelegt.

Bei der EM-Bereitstellung in der MSB/H wird die EM-Checkliste ebenfalls ausgedruckt. Anschließend werden alle für die Abteilung relevanten Informationen in ein Anwendungssystem namens Erstmuster-Datenbank übernommen (s. Abb. 3.6). Hierzu zählen u. a. der Kundename, die RB-Sachnummer, die Anzahl der bereitzustellenden Erstmuster sowie der Kundenwunschtermin der Lieferung. Darüber hinaus werden auch Informationen aufgenommen, die nicht in der EM-Checkliste enthalten sind und nur von der MSB/H benötigt werden (z. B. die Leiterplattennummer). Über das Anwendungssystem lassen sich dann bspw. alle noch zu bearbeitenden EM-Bestellungen, geordnet nach dem Kundenwunschtermin, anzeigen. Auf diesem Wege wählen die Mitarbeiter der MSB/H die als nächstes bereitzustellenden Erstmuster aus.

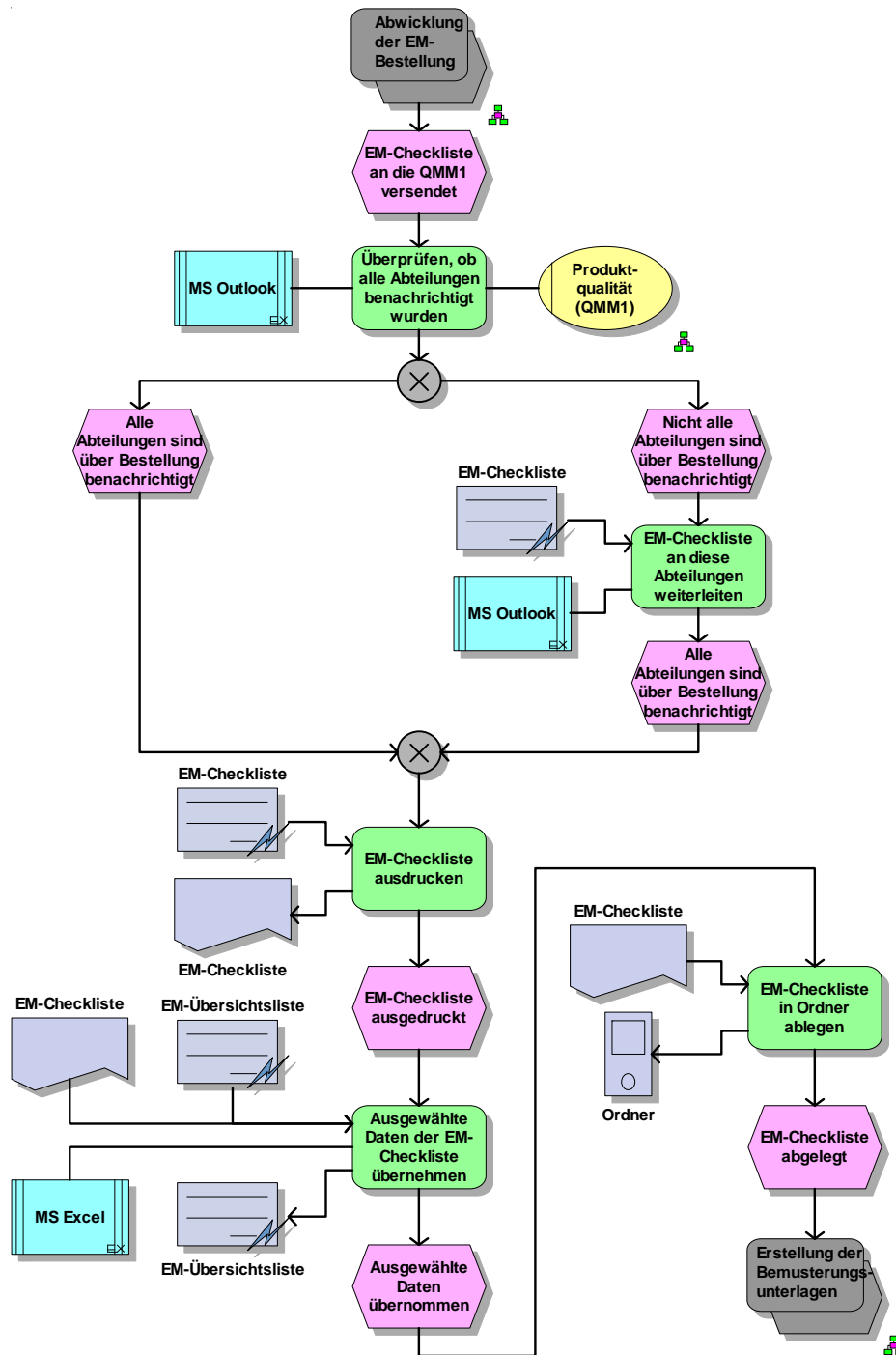


Abb. 3.5: eEPK zur Dokumentationsvorbereitung (Ist-Zustand)

Ist z. B. eine Hardwareänderung Gegenstand der Bemusterung, wird die Produktion neuer Steuergeräte angestoßen. Muss dagegen eine Softwareänderung bemustert werden, sind bereits vorhandene Steuergeräte (also die Hardware) vom selben Typ, den der Kunde wünscht, aus dem Lager zu entnehmen und mit der geänderten Software zu versehen. Die für die Bereitstellung der Erstmuster benötigten Informationen, werden dabei der ausgedruckten EM-Checkliste sowie der Erstmuster-Datenbank entnommen.

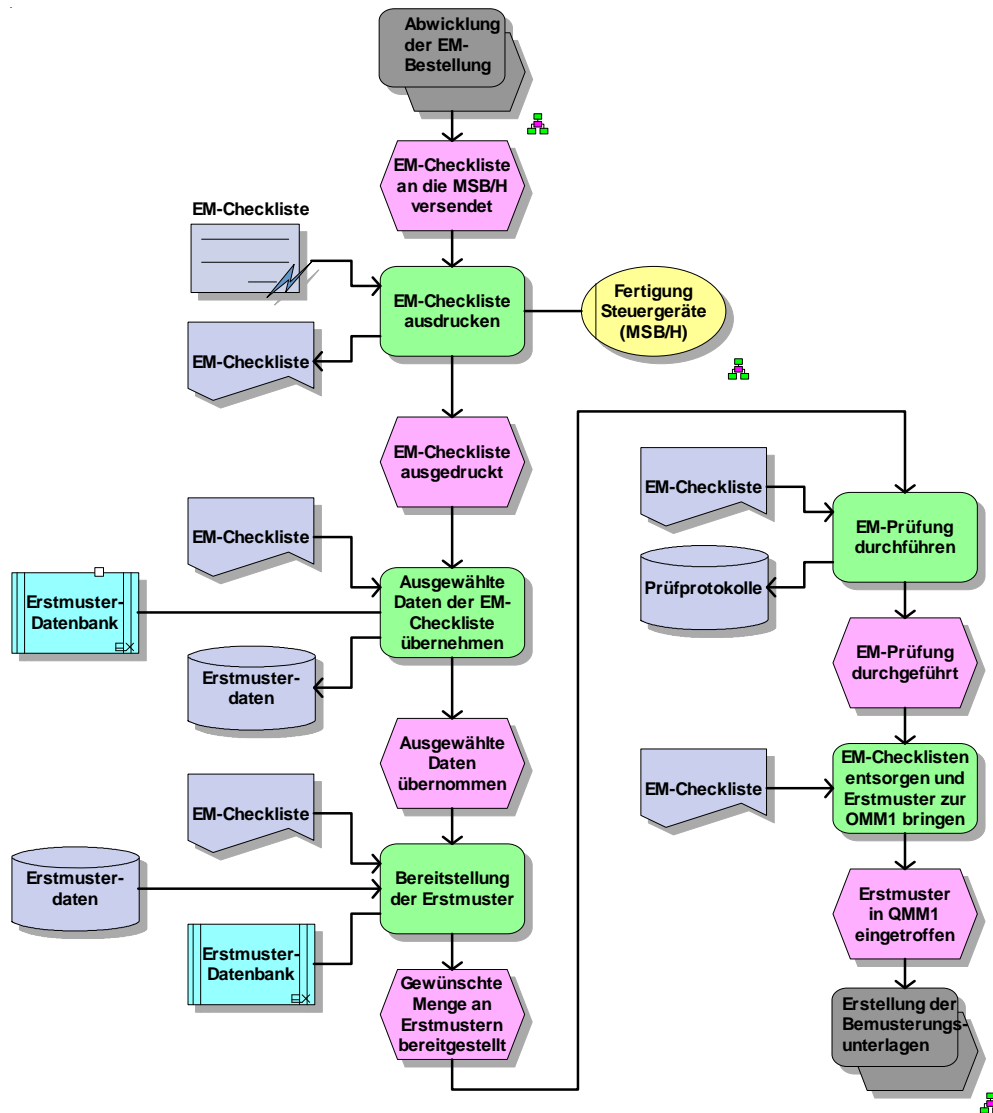


Abb. 3.6: eEPK zur EM-Bereitstellung und -Prüfung (Ist-Zustand)

Darüber hinaus sind die Erstmuster den in der EM-Checkliste vereinbarten Prüfungen zu unterziehen, wobei die entstehenden Prüfprotokolle automatisch in einer Datenbank gespeichert werden. Ist die EM-Prüfung abgeschlossen, werden die ausgedruckten EM-Checklisten entsorgt und die Erstmuster zur QMM1 gebracht.

Sobald Erstmuster in der QMM1 eingetroffen sind, wird in der EM-Übersichtsliste nach dem vom Kunden gewünschten Zustellungstermin gesucht (s. Abb. 3.7). Der Zugriff erfolgt dabei über die RB-Sachnummer, welche auf der Beschriftung der Erstmuster zu finden ist. Im Fall, dass mehrere Bemusterungen abzarbeiten sind, wird zuerst die mit dem am nächsten gelegenen Kundenwunschtermin ausgewählt. Bevor jedoch mit der Erstellung der Bemusterungsunterlagen begonnen werden kann, muss die passende EM-Checkliste gesucht werden. Hierfür greift der zuständige Sachbearbeiter über den in der EM-Übersichtsliste ermittelten Kundenwunschtermin auf einen physischen Ordner zu.

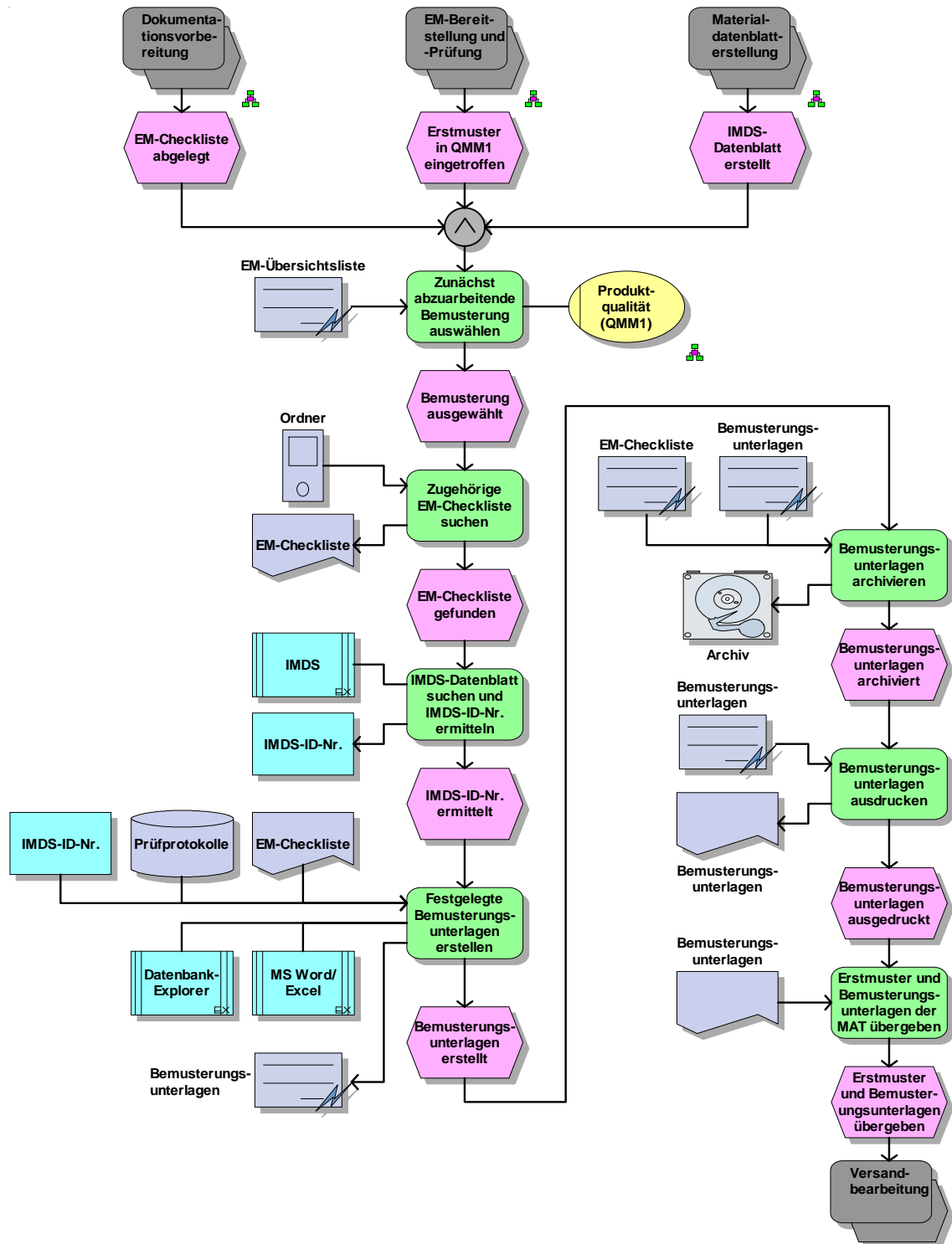


Abb. 3.7: eEPK zur Erstellung der Bemusterungsunterlagen (Ist-Zustand)

Des Weiteren muss im IMDS mit Hilfe der RB-Sachnummer das zugehörige Materialdatenblatt gesucht und dessen eindeutige IMDS-ID-Nr. ermittelt werden. Wurden die EM-Checkliste sowie das IMDS-Datenblatt gefunden, kann basierend auf den festgelegten Vereinbarungen die Erstellung der Bemusterungsunterlagen beginnen.

Hierbei werden unter Verwendung von MS Word- bzw. MS Excel-Formularen die vom Kunden gewünschten Dokumente erstellt, wobei ein Großteil der hierfür benötigten Informationen aus der ausgedruckten EM-Checkliste übernommen wird. Darüber hinaus müssen die Ergebnisse der durchgeführten EM-Prüfung dokumentiert werden. Auf die gespeicherten Prüfprotokolle wird dabei über eine Anwendung namens Datenbank-Explorer zugegriffen. Die dafür notwendigen Informationen (RB-Sachnummer, Prüfdatum, -stand und -nummer) werden von der Beschriftung der Erstmuster abgelesen.

Nach der Erstellung der vereinbarten Dokumentation werden alle im Rahmen der Bemusterung entstanden bzw. hinzugefügten Dokumente und Aufzeichnungen (wie bspw. Prozessablaufdiagramme, Prozessfähigkeitsnachweise, Prüfmittellisten, Spezifikationen und Zertifikate) sowie die ursprünglich per E-Mail empfangende und noch immer in MS Outlook vorliegende EM-Checkliste mit dem Stand des Bemusterungszeitpunktes elektronisch archiviert.²⁷ Abschließend werden die Erstmuster zusammen mit den ausgedruckten Bemusterungsunterlagen der MAT übergeben, welche diese an den Kunden versendet.

3.4 Schwachstellenanalyse

Die Schwachstellenanalyse ist die gezielte Untersuchung eines Prozesses zur Ermittlung von Schwachstellen mit dem Ziel den Prozess zu verbessern. Mögliche Schwachstellen sind bspw.:

- Zeitverzögerungen bzw. lange Durchlaufzeiten,
- organisatorische Brüche (häufiger Wechsel von verantwortlichen Organisationseinheiten im Prozess),
- Systembrüche (Anzahl der eingebundenen Informationssysteme im Prozess),
- Medienbrüche (Anzahl der Wechsel zwischen manueller und DV-gestützter Prozessbearbeitung),
- Datenredundanzen
- und Mehrfacherfassung bzw. Doppelarbeiten (vgl. Scheer 2002, S. 150 f.).

²⁷ In der Regel werden produkt- und prozessrelevante Dokumente, Aufzeichnungen und Daten mindestens für die Zeit aufbewahrt, in der für das Produkt Liefervereinbarungen bestehen.

Im Folgenden werden einige Schwachstellen der zuvor beschriebenen Prozesse aufgeführt. Dabei sollen im Besonderen die Mängel, die im Zusammenhang mit der EM-Checkliste stehen, betrachtet werden.

- Gegenwärtig werden diverse Varianten von EM-Checklisten verwendet, die sich in erster Linie in ihrem Layout unterscheiden.²⁸ Dieser Umstand erschwert die Arbeit mit der EM-Checkliste und führt zu Zeitverzögerungen bei der Bemusterung, da bspw. für einen bestimmten Arbeitsschritt benötigte Informationen je nach Variante der verwendeten EM-Checkliste an unterschiedlichen Stellen innerhalb des Dokumentes zu finden sind. Darüber hinaus erfordert die Einarbeitung neuer Mitarbeiter einen höheren Zeitaufwand.
- Eine EM-Checkliste wird in der Regel von zwei Vertriebsmitarbeitern ausgefüllt, was ebenfalls zu Zeitverzögerungen führt, da z. B. Wartezeiten entstehen und möglicherweise Rücksprachen zwischen den beteiligten Mitarbeitern notwendig sind. Zudem werden die EM-Checklisten durch das Hin- und Hersenden innerhalb des EA-Vertriebs redundant an unterschiedlichen Arbeitsplätzen vorgehalten.
- Beim Ausfüllen der EM-Checkliste werden teilweise auch Informationen aufgenommen, die bei der Bemusterung im Werk nicht benötigt werden. Des Weiteren werden in einigen EM-Checklisten bestimmte Informationen mehrfach erfasst. Es kommt auch vor, dass EM-Checklisten unvollständig oder fehlerhaft ausgefüllt sind. Dieser Umstand macht Rücksprachen mit dem jeweiligen Vertriebsmitarbeiter erforderlich, was wiederum zu Neueingaben führt und den Ablauf der Bemusterung verzögert.
- EM-Checklisten werden vom EA-Vertrieb nicht immer an alle für die Bemusterung zuständigen Abteilungen und Stellen geschickt, da nicht jeder Vertriebsmitarbeiter diese auch kennt. Für die Sachbearbeiter der QMM1 bedeutet dies Mehrarbeit, weil sie in diesem Fall die EM-Checklisten weiterleiten müssen.
- In der QMM1 und MSB/H werden die in elektronischer Form vorliegenden EM-Checklisten ausgedruckt und ausgewählte Informationen aus diesen per Hand in die EM-Übersichtsliste bzw. die Erstmuster-Datenbank übertragen (Medienbrüche). Diese Mehrfacherfassung führt nicht nur zu Zeitverzögerungen sondern auch zu Datenredundanzen. Darüber hinaus ist die manuelle Übernahme der Informationen fehleranfällig. Zusätzlich wird Platz für die physische Ablage der ausgedruckten EM-Checklisten gebraucht.

²⁸ Es existieren momentan ca. zwölf verschiedene Varianten von EM-Checklisten.

- Die Suche nach einer abgelegten EM-Checkliste erfordert in der QMM1 mehrere Schritte. Es muss zuerst in der EM-Übersichtsliste, z. B. über die RB-Sachnummer, nach dem jeweiligen Kundenwunschtermin gesucht werden. Mit diesem und dem Namen des zuständigen Sachbearbeiters kann dann in dem jeweiligen physischen Ordner auf die gesuchte EM-Checkliste zugegriffen werden.
- Bei der Erstellung der Bemusterungsunterlagen werden die benötigten Informationen aus der EM-Checkliste per Hand in die zu erstellenden Dokumente übertragen.

4 Soll-Konzept

In diesem Kapitel wird das zu entwickelnde webbasierte Informationssystem vorgestellt, wobei die Systemanforderungen formuliert und die Datenbasis des zukünftigen Websystems spezifiziert werden. Die Planungsgrundlage für das fachliche Soll-Konzept bilden dabei die aus der Ist-Analyse gewonnenen Informationen. Im Rahmen dieser Entwicklungsphase wird die Frage nach dem „Was“ beantwortet und schließlich ein möglicher Soll-Zustand für den Ablauf der Bemusterung in der RB Elektronik GmbH beschrieben und dargestellt.

4.1 Zielbestimmung

Das übergeordnete Ziel besteht darin, so viele der aufgezeigten Schwachstellen wie möglich zu beseitigen. Um dies zu gewährleisten, soll ein webbasiertes Informationssystem mit der Bezeichnung Initial Sample Documentation Order System (ISDOS) entwickelt werden. Dieses Websystem soll es ermöglichen, dass zukünftig alle für die Bemusterung benötigten Informationen in einheitlicher Form über das firmeninterne Netzwerk erfasst und zentral in einer Datenbank verwaltet werden können. Im Rahmen des Entwicklungsvorhabens sind die momentanen EM-Checklisten genau zu untersuchen und Informationen zu identifizieren, welche während der Bemusterung nicht benötigt werden. Des Weiteren müssen bisher nicht aufnehmbare Informationen, die in aktuell gültigen Normen (s. Kapitel 2) gefordert werden, mit dem Websystem erfassbar sein. Grundsätzlich ist das System so allgemein wie möglich zu entwickeln, damit dieses auch an anderen Fertigungsstandorten der RB GmbH eingesetzt werden kann.

4.2 Anforderungen

Anforderungen legen die zu erfüllenden Eigenschaften und zu erbringenden Leistungen eines Systems fest (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 32). Sie sind so zu formulieren, dass nur das „Was“ und nicht das „Wie“ beschrieben wird. Des Weiteren ist zu beachten, dass Anforderungen eindeutig, prüfbar und widerspruchsfrei sind. In der Literatur werden verschiedene Typen von Anforderungen definiert und voneinander abgegrenzt. Ein Ansatz ist bspw. die Unterteilung in funktionale, qualitative, plattformbezogene und prozessbezogene Anforderungen (vgl. Dumke/Lothar/Wille/Zbrog 2003, S. 60).²⁹ Grundsätzlich können zwei Hauptgruppen unterschieden werden, die funktionalen und die nicht-funktionalen Anforderungen. Funktionale Anforderungen legen fest,

²⁹ Für weiterführende Informationen sei an dieser Stelle auf die angegebene Literaturquelle verwiesen.

über welche Dienste ein System verfügen soll. Nicht-funktionale Anforderungen beziehen sich nicht direkt auf die Funktionalität eines Systems sondern bspw. auf Qualitätsmerkmale, Randbedingungen, Leistungsanforderungen und Systemschnittstellen (vgl. Kappel/Pröll/ Reich/Retschitzegger 2004, S. 40).

Im Folgenden werden die in Abstimmung mit dem Auftraggeber und den zukünftigen Nutzern erarbeiteten Anforderungen an das Websystem beschrieben. An dieser Stelle sei gesagt, dass einige nicht-funktionale Anforderungen (z. B. Serverplattform, Datenbank, Entwicklungstechnologie) vom Auftraggeber vorgegeben wurden.

4.2.1 Funktionale Anforderungen

Für die Beschreibung der funktionalen Anforderungen werden im Rahmen dieser Arbeit Anwendungsfälle³⁰ (engl. use cases) verwendet. Anwendungsfälle beschreiben die Funktionalität eines Systems aus der Sicht seiner Benutzer. Sie erläutern, wie ein System benutzt wird, indem sie die Interaktion zwischen den Systembenutzern und dem System selbst beschreiben (vgl. Fowler 2004, S. 121).

Nachfolgend sind die jeweiligen Anwendungsfälle mit einer Bezeichnung und einer kurzen Beschreibung aufgeführt. Im Anschluss wird eine erste Systemübersicht in Form eines vereinfachten Anwendungsfalldiagramms gegeben (s. Abb. 4.1).

- **Bestellung erfassen**

Das webbasierte Informationssystem soll es einem Mitarbeiter des EA-Vertriebs ermöglichen, alle für die Bemusterung benötigten Informationen über eine einheitliche Eingabemaske zu erfassen, wobei gewährleistet sein muss, dass auch alle Informationen aufnehmbar sind, die in aktuell gültigen Normen gefordert werden.

Beim Erfassen der Bestellung soll ein Vertriebsmitarbeiter mit ISDOS die Möglichkeit haben, ein leeres Formular oder eine bereits gespeicherte Bestellung als Vorlage verwenden zu können. Letzteres wird eine vorherige Suche erforderlich machen.

Die getätigten Eingaben des Formulars sollen zentral in einer Datenbank gespeichert werden, wobei der Bearbeitungsstatus der Bestellung auf „angelegt“ zu setzen ist. Die so angelegten Bestellungen sollen zukünftig als Auslöser der

³⁰ Dieses Konzept hielt 1995 Einzug in die UML.

Bemusterung dienen und u. a. die Art und den Umfang der anstehenden Prüfungen und Dokumentation festlegen.

- **Bestellung öffnen**

Das webbasierte Informationssystem ISDOS muss einem Benutzer die Möglichkeit bieten, in der Datenbank gespeicherte Bestellungen zu öffnen. In diesem Zusammenhang ist eine vorherige Suche erforderlich. Ist diese erfolgreich, soll ein Benutzer je nachdem, welcher Benutzergruppe er angehört und welche Rechte mit dieser verbunden sind, geöffnete Bestellungen ansehen, ändern, stornieren und löschen können.

So soll bspw. ein Systembenutzer, welcher der Benutzergruppe „Werk“ angehört, die Möglichkeit haben, Bestellungen anzusehen und zu ändern, um fehlerhafte Eingaben korrigieren bzw. Ergänzungen vornehmen zu können. Benutzer der Benutzergruppe „EA-Vertrieb“ sollen dagegen vorhandene Bestellungen ansehen, ändern, stornieren und löschen können. Dem IMDS-Beauftragten soll es ausschließlich möglich sein, Bestellungen anzusehen.

- **Bestellung suchen**

Ein Benutzer des Websystems soll die Möglichkeit haben, in der Datenbank gespeicherte Bestellungen zu suchen. Hierfür soll eine Standardsuche über RB-Benennung, RB-Sachnummer, RB-Änderungsnummer, Name und Sachnummer des Kunden sowie über den eindeutigen Identifikator (ID) der Bestellung zur Verfügung gestellt werden.

Liefern die Suchkriterien der Standardsuche nicht das gewünschte Ergebnis, soll dieses durch die Möglichkeit einer erweiterten Suche eingrenzbar sein. Hierbei soll die Suche über Name und Abteilung des Ausstellers, Kundenwerk und Kunden-Auftragsnummer, EVA-Auftragsnummer und Bearbeitungsstatus der Bestellung sowie über die Art des Bemusterungsverfahrens und die Vorlagestufe möglich sein.

Das Suchergebnis soll in tabellarischer Form angezeigt werden, wobei als Spaltennamen die Suchkriterien der Standardsuche verwendet werden. Jede Zeile soll in der letzten Spalte ein kleines Symbol enthalten, mit welchem der Benutzer, je nachdem für welchen Zweck die Suche aufgerufen wurde, durch Anklicken eine gewünschte Bestellung öffnen bzw. als Vorlage verwenden kann.

- **Bestellung versenden**

Das Websystem muss einem Mitarbeiter des EA-Vertriebs die Möglichkeit bieten, eine vollständig ausgefüllte Bestellung nach dem Erfassen bzw. Öffnen an den zuständigen Fertigungsstandort übermitteln zu können. Hierbei sollen fest im Websystem hinterlegte Verteilerlisten sicherstellen, dass alle an der Bemusterung beteiligten Abteilungen des Werks sowie der IMDS-Beauftragte informiert werden. Darüber hinaus sollen, zusätzlich zur ausgewählten nicht veränderbaren Verteilerliste, weitere E-Mail-Adressen eingegeben werden können.

Der in der Datenbank gespeicherte Bearbeitungsstatus der Bestellung ist nach dem erfolgreichen Versenden der E-Mail auf „ans Werk übermittelt“ zu setzen. Die E-Mail soll als Betreff den Namen des Websystems, die eindeutige ID, unter welcher die Bestellung in der Datenbank gespeichert ist, die RB-Benennung sowie die RB-Sachnummer enthalten. Der Inhalt der E-Mail soll aus einem Verweis (engl. link) bestehen, mit dem die Bestellung geöffnet werden kann. Zudem soll ein Benutzer die Möglichkeit haben, der E-Mail eine kurze Mitteilung hinzuzufügen.

Weiterhin soll gewährleistet sein, dass ein Vertriebsmitarbeiter eine unvollständig ausgefüllte Bestellung zur weiteren Bearbeitung an einen anderen Vertriebsmitarbeiter versenden kann. In diesem Fall wird der Bearbeitungsstatus der Bestellung nicht verändert. Die versendete E-Mail ist dabei ebenso aufgebaut, wie zuvor beschrieben.

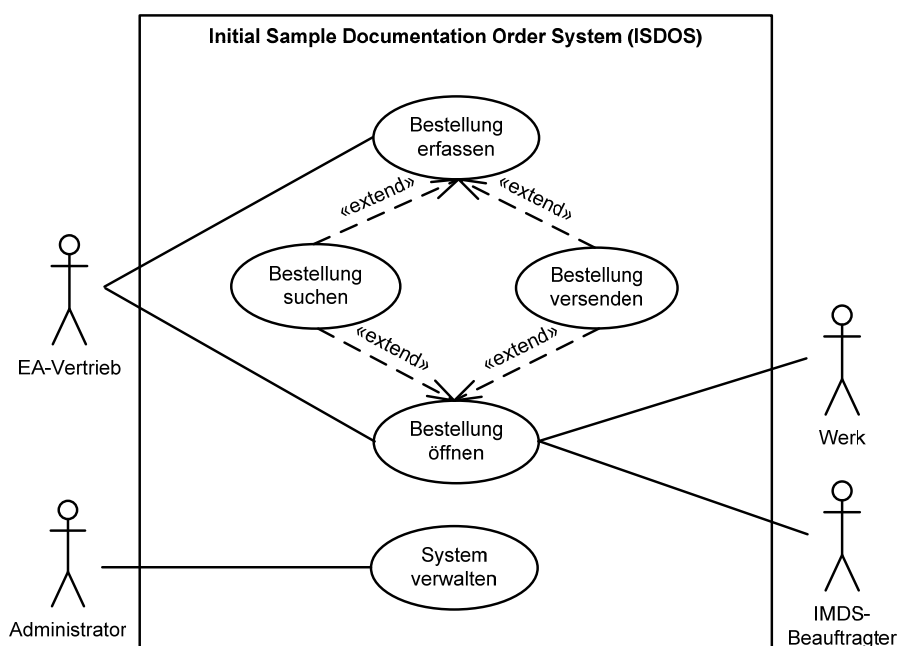


Abb. 4.1: Systemübersicht als Anwendungsfalldiagramm

- **System verwalten**

Der administrative Bereich des Websystems soll es einem Administrator ermöglichen, die in der Datenbank gespeicherten Benutzerrollen³¹ mit ihren Rechten, die Verteilerlisten sowie die Kundendaten verwalten zu können. Hierbei muss für jeden der drei Punkte gewährleistet sein, dass neue Einträge angelegt sowie vorhandene Einträge angezeigt, geändert und gelöscht werden können.

4.2.2 Nicht-funktionale Anforderungen

Die nicht-funktionalen Anforderungen werden nachfolgend in Form einer Anforderungsliste aufgeführt, wobei jede Anforderung über eine Bezeichnung verfügt.

- **Sicherheit:** Das Websystem muss vor unerlaubtem Zugriff geschützt werden. Als Schutzmaßnahme sind Kennwörter zu verwenden, mit denen sich die Benutzer am Informationssystem anmelden müssen. Spezielle Verschlüsselungsmechanismen für die Datenübertragung sind nicht erforderlich.
- **Bedienbarkeit:** Die Webanwendung soll eine einfach zu bedienende Benutzeroberfläche mit übersichtlich strukturierten Ein- und Ausgabemasken besitzen.
- **Erlernbarkeit:** Ein beliebiger Benutzer soll ohne großen Lernaufwand mit ISDOS arbeiten können.
- **Mehrsprachigkeit:** Das webbasierte Informationssystem soll weltweit eingesetzt werden und daher mehrsprachig sein, wobei zuerst die Sprachen Deutsch und Englisch zu unterstützen sind. Zwischen den Sprachen soll während der Nutzung jederzeit gewechselt werden können.
- **Erweiterbarkeit:** Das System soll leicht erweiterbar und wartbar sein. So sollen u. a. neue Funktionen und weitere Sprachen leicht hinzugefügt werden können. Die im Unternehmen geltenden Codekonventionen müssen daher unbedingt eingehalten und der Programmcode ausführlich dokumentiert werden.
- **Übertragbarkeit:** Es muss gewährleistet sein, dass ISDOS ohne hohen Änderungsaufwand von einer Computerplattform auf eine andere portiert werden kann, damit dieses u. a. auch in anderen RB-internen Werken einsetzbar ist.

³¹ Benutzerrollen können als konzeptionelle Weiterentwicklung von Benutzergruppen verstanden werden. In Abschnitt 4.3.1 wird auf das Rollenkonzept ausführlicher eingegangen.

- **Fehlertoleranz:** Mögliche Laufzeitfehler müssen abgefangen, durch eine zentrale Fehlerbehandlung ausgewertet und dem Benutzer erläutert werden können.
- **Zuverlässigkeit:** Die Webanwendung und sämtliche Daten sollen jederzeit verfügbar sein, damit eine reibungslose Arbeit gewährleistet werden kann.
- **Hilfefunktion:** An noch genau zu spezifizierenden Stellen sollen einem Benutzer hilfreiche Informationen angezeigt werden können, die ihn bspw. über die Bedeutung eines Eingabefeldes informieren.
- **Entwicklungstechnologien:** Die Webanwendung soll mit der serverseitigen Technologie ASP.NET³², in der aktuellen Version 2.0, und der Programmiersprache Visual Basic (VB) 2005 entwickelt werden.
- **Webbrowser:** Die Webanwendung soll in erster Linie für den MS Internet Explorer optimiert werden, jedoch auch mit anderen Webbrowsern ohne Funktionalitätseinbußen verwendbar sein.
- **Webserver:** Als Serverplattform soll MS Windows Server 2003 mit integriertem Internet Information Server (IIS), in der aktuellen Version 6.0, zum Einsatz kommen.
- **Datenhaltung:** Als Datenbanksystem (DBS)³³ soll Oracle 9i eingesetzt werden.
- **Datenzugriff:** Die Erstmuster-Datenbank der MSB/H sowie ein Anwendungssystem der QMM1³⁴ sollen Zugriff auf einen ausgewählten Bereich der Datenbasis des zu entwickelnden Websystems haben, damit es für diese Systeme möglich ist, bestimmte Daten einer EM-Bestellung importieren zu können.

4.3 Datenmodellierung

In diesem Abschnitt wird anhand der gesammelten Anforderungen das Datenbankmodell³⁵ spezifiziert, welches zur Erfassung und Darstellung der Informationsstruktur des zukünftigen Websystems dient und noch unabhängig vom später einzusetzenden DBS ist. Hierbei gilt es, alle für die Bemusterung benötigten Informationen, wie auch die

³² .NET ist der Oberbegriff für eine Softwareentwicklungsinfrastruktur der Firma Microsoft.

³³ Ein DBS ist die Kombination eines Datenbankmanagementsystems (DBMS) mit einer Datenbank. Unter einem DBMS wird eine Software zur Verwaltung einer Datenbank verstanden (vgl. Heuer/Saake 2000, S. 8).

³⁴ Auf dieses Anwendungssystem wird in Abschnitt 4.4 ausführlicher eingegangen.

³⁵ Als Datenbankmodell wird ein System von Konzepten zur Beschreibung von Datenbanken verstanden (vgl. Heuer/Saake 2000, S. 50).

einzelnen Benutzerrollen mit ihren zugeordneten Rechten sowie die für die Mehrsprachigkeit des Websystems benötigten Übersetzungen mit ihren Eigenschaften und Beziehungen untereinander zu betrachten. Bevor das Gesamtdatenbankmodell vorgestellt wird, soll kurz separat auf das Rollenkonzept sowie das Konzept zur Umsetzung der Mehrsprachigkeit eingegangen werden.

Als klassisches Modell für den Entwurf von Datenbanken hat sich das so genannte Entity-Relationship-Modell (ERM) von Chen³⁶ mit seinen zahlreichen Variationen bewährt. Dies ist ein abstraktes Modell, in dem Informationseinheiten (Entities), beliebige Beziehungen (Relationships) zwischen diesen und Attribute modelliert werden. Neben den Basiskonzepten wurden bereits frühzeitig weitere Modellierungskonzepte, wie bspw. Schlüsselattribute, Kardinalitäten etc. vorgeschlagen. Eine Erweiterung des ERMs ist das erweiterte Entity-Relationship-Modell (eERM), ein semantisches Datenmodell, in dem anstelle eines allgemeinen Beziehungskonzeptes weitere Beziehungsarten wie etwa Spezialisierung/ Generalisierung eingeführt wurden (vgl. Heuer/Saake 2000, S. 55 ff.).

Im Rahmen dieser Arbeit wird für die Datenmodellierung das eERM von ARIS verwendet. Da das Websystem wie bereits erwähnt so zu konzipieren ist, dass dieses auch an anderen Fertigungsstandorten der RB GmbH zur Erfassung und Verwaltung der für die Bemusterung benötigten Informationen eingesetzt werden kann, besteht die Zielsetzung darin ein Datenbankmodell zu erstellen, das eine möglichst hohe Flexibilität der zukünftigen Datenbank gewährleistet.

4.3.1 Rollenkonzept

Als Rollenkonzept wird in der EDV ein Ansatz zur Benutzerverwaltung verstanden, bei dem je nach Aufgabenbereich Benutzerrollen (kurz Rollen) definiert und Systembenutzern zugewiesen werden. Benutzerrollen sind mit bestimmten Rechten verbunden, wobei ein Benutzer auch mehrere Rollen innehaben kann.

Wie in den funktionalen Anforderungen definiert, soll im Websystem die Möglichkeit bestehen Benutzerrollen zu verwalten. Daten zu einzelnen Systembenutzern werden dagegen nicht in der Datenbank gespeichert. Vielmehr sollen im Rahmen der Realisierung zuerst die Benutzerrollen „Administrator“, „EA-Vertrieb“ und „Werk“, welche im Anwendungsfalldiagramm (s. Abb. 4.1) dargestellt sind, sowie die Rolle „Gast“ angelegt werden. Letztere kann bspw. vom IMDS-Beauftragten eingenommen werden. Prin-

³⁶ Siehe Chen, P. (1976): The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems, Band 1, Nr. 1, S. 9 – 36.

ziell soll ein Benutzer eine Benutzerrolle einnehmen können, indem er sich mit dem Benutzerrollennamen und dem passenden Kennwort am Informationssystem anmeldet.

Bei diesem Entwicklungsvorhaben wird die Verwaltung von wenigen Benutzerrollen der von vielen Benutzern (und Rollen) vorgezogen, um den Aufwand für die Benutzerverwaltung so niedrig wie möglich zu halten. Dies ist sinnvoll, da die Administration zukünftig von einem Sachbearbeiter der QMM1, neben seinen regulär anstehenden Tätigkeiten, zu bewältigen sein soll. Weiterhin ist die Anzahl der möglichen Systembenutzer nur schwer abschätzbar, weil das Websystem weltweit in diversen Verkaufshäusern des EA-Vertriebs zur Bestellung von Erstmustern genutzt werden soll.

In diesem Rollenkonzept werden die Benutzerrollen durch eine eindeutige ID, einen Namen sowie ein Kennwort definiert (s. Abb. 4.2). Die Benutzerrollen stehen zudem mit einer Menge von Rechten in Beziehung, welche für diese erlaubt sind. Ein im Websystem zur Verfügung stehendes Recht setzt sich aus einem Objekt und einer Aktion zusammen, welche für das Objekt ausgeführt werden kann. Die Rechte (z. B. „Bestellung anlegen“) sollen mit einer eindeutigen ID sowie einer Bezeichnung für das Objekt (z. B. „Bestellung“) und die Aktion (z. B. „anlegen“) in der Datenbank vorgehalten werden. Im Rahmen des Systementwurfs, genauer gesagt bei der Vorstellung des Sicherheitskonzepts (s. Abschnitt 5.6), wird das Rollenkonzept ausführlicher betrachtet.

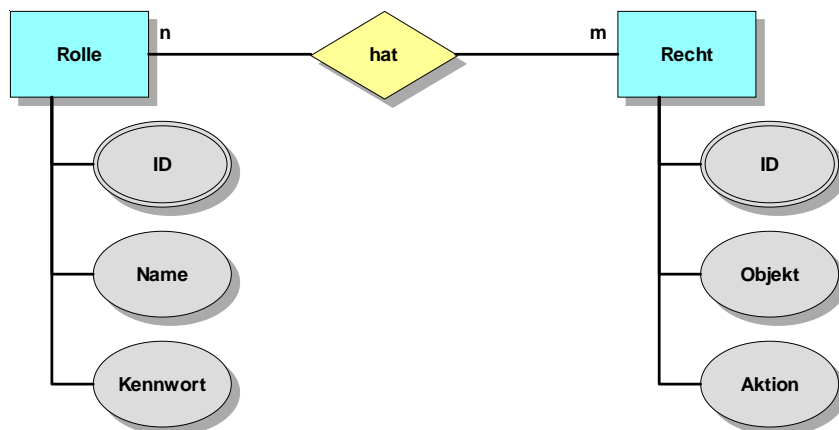


Abb. 4.2: eERM zum Rollenkonzept (Soll-Zustand)

4.3.2 Mehrsprachigkeit

Zur Umsetzung der Mehrsprachigkeit des zukünftigen Websystems sollen einige der mehrsprachigen Zeichenfolgen in der Datenbank vorgehalten werden. Neben den Übersetzungen werden auch die vom Informationssystem unterstützten Sprachen gespei-

chert. Eine Sprache wird dabei mit einer eindeutigen ID, einer Bezeichnung (z. B. „Deutsch“) und ihrem Kulturkürzel (z. B. „de-DE“) in der Datenbank vorgehalten (s. Abb. 4.3). Die in den verschiedenen Sprachen vorliegenden Zeichenfolgen sollen jeweils mit einer eindeutigen ID gespeichert werden und immer mit einer im Web-system zur Verfügung stehenden Sprache in Beziehung stehen. Im Rahmen des System-entwurfs wird genauer auf die Umsetzung der Mehrsprachigkeit eingegangen (s. Abschnitt 5.5.3).

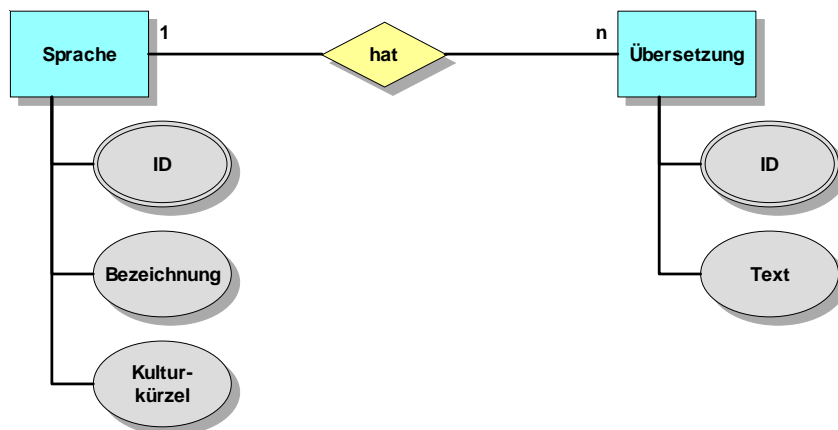


Abb. 4.3: eERM zur Umsetzung der Mehrsprachigkeit (Soll-Zustand)

4.3.3 Gesamtkonzept

In Abb. 4.4 ist das Gesamtdatenbankmodell des zu entwickelnden Websystems dargestellt. Zur Reduzierung der Komplexität wurde auf die Angabe der Attribute der einzelnen Entitytypen verzichtet.³⁷ Im Vordergrund stehen vielmehr die Entitytypen und deren Beziehungen untereinander, welche nachfolgend kurz beschrieben werden.

Eine Bestellung von Erstmustern soll wie bisher von einem bzw. zwei Vertriebsmitarbeitern für einen Kunden und ein bestimmtes Kundenwerk erfasst werden können. Neben diesen Vertriebsmitarbeiter- und Kundendaten werden Informationen zum Ansprechpartner des Qualitätsmanagements des Kunden sowie zu den Kennzeichnungen und Aufklebern für die Erstmuster in der Datenbank gespeichert. Des Weiteren sollen die ausgewählte Verteilerliste, mit welcher die Bestellung an die für die Bemusterung zuständigen Abteilungen und Stellen übermittelt wird, sowie der Bearbeitungsstatus zusammen mit der Bestellung vorgehalten werden. Zu jeder Bestellung wird zudem die Benutzerrolle festgehalten, welche ein Benutzer innehatte, der zuletzt Änderungen an

³⁷ An dieser Stelle sei auf den Systementwurf verwiesen, in welchem die Attribute der einzelnen Entitytypen ausführlich dargestellt werden.

dieser vorgenommen hat. Wie bereits aufgeführt, soll bei der Erfassung einer Bestellung das Bemusterungsverfahren festgelegt und gespeichert werden, wobei in diesem Zusammenhang zwischen dem PPF-Verfahren, dem PPAP-Verfahren oder einer kundenspezifischen Bemusterung zu wählen ist.

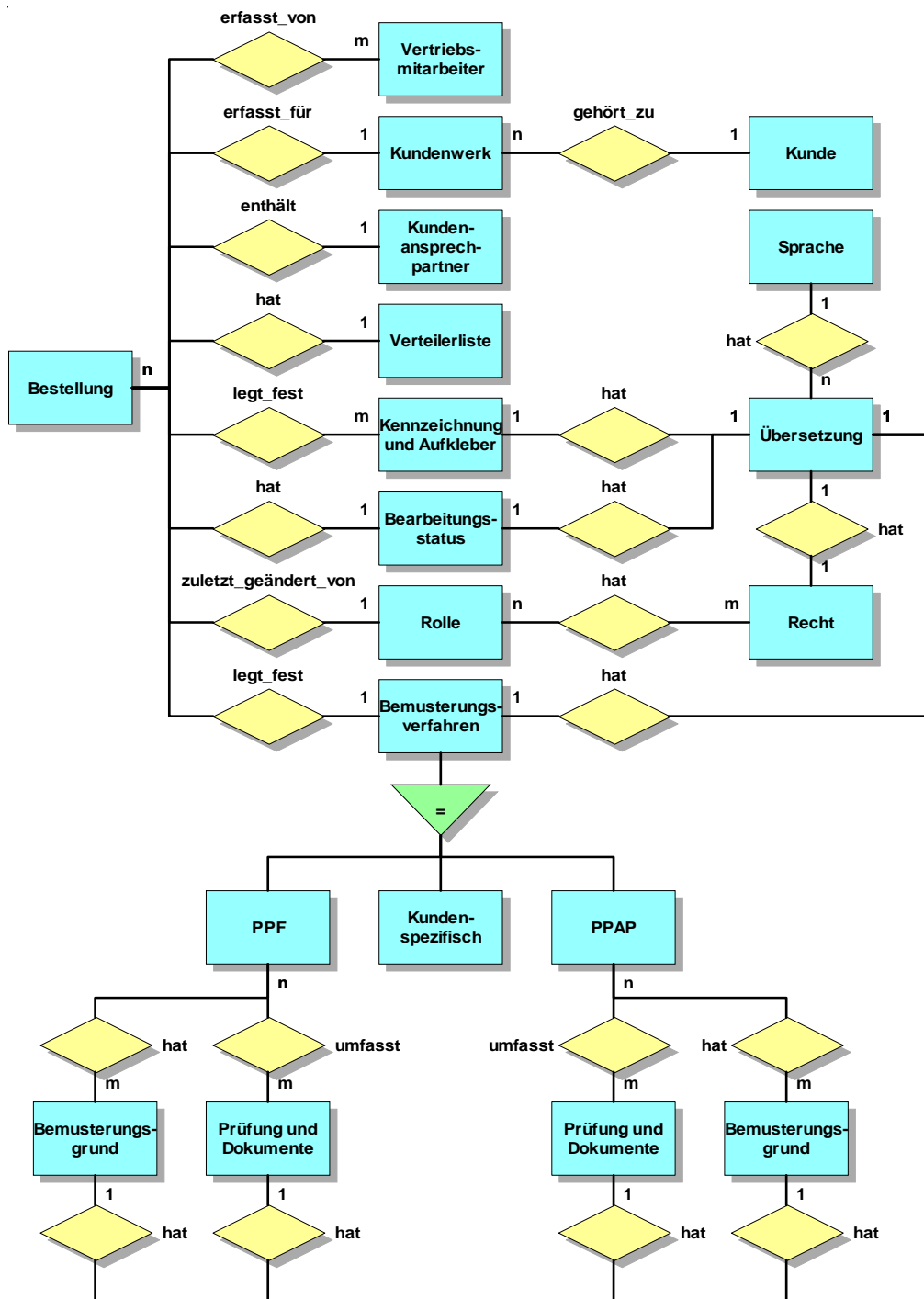


Abb. 4.4: Gesamtdatenbankmodell in eER-Notation ohne Attribute (Soll-Zustand)

Im Datenbankmodell ist diese Unterteilung als Partitionierung dargestellt. Die Partitionierung ist ein Sonderfall der Spezialisierung, wobei ein Entitytyp in mehrere disjunkte Entitytypen spezialisiert wird. Können alle Instanzen eindeutig einer Partition zugeteilt werden, wird dies als vollständige Partitionierung oder auch disjunkte Überdeckung bezeichnet. Für die Darstellung wird ein so genannter Typkonstruktor in Form eines Dreiecks mit einem Gleichheitssymbol verwendet, wobei die Spitze des Dreiecks auf die Untermenge zeigt (vgl. Heuer/Saake 2000, S. 75 ff.).

Darüber hinaus werden zu einem festgelegten Bemusterungsverfahren, außer dieses ist kundenspezifisch, der Bemusterungsgrund sowie die durchzuführenden Prüfungen und anzufertigenden Dokumente in der Datenbank gespeichert. Der Bemusterungsgrund sowie die Bezeichnungen der Prüfungen und Dokumente sollen hierbei wie auch der Bearbeitungsstatus, die Rechte, die Namen der Bemusterungsverfahren sowie die Beschreibungen zu den Kennzeichnungen und Aufklebern in verschiedenen Sprachen vorgehalten werden.

4.4 Neustrukturierung der Ablauforganisation

Im Folgenden wird der Soll-Zustand des Bemusterungsablaufs in der RB Elektronik GmbH beschrieben, wobei in erster Linie auf Verbesserungen gegenüber dem aktuellen Ist-Zustand eingegangen werden soll. Die einzelnen Teilprozesse werden, wie im Rahmen der Ist-Analyse, mit Hilfe des ARIS-Toolsets modelliert.

Die Bestellung der Erstmuster und der dazugehörigen Bemusterungsunterlagen wird wie bisher vom EA-Vertrieb erfasst (s. Abb. 4.5), nur werden die für die Bemusterung benötigten Informationen nicht mehr unter Verwendung von MS Word bzw. MS Excel aufgenommen. Zukünftig wird eine Bestellung mit ISDOS über ein einheitliches Formular erfasst und zentral in einer Datenbank gespeichert. Die unterschiedlichen Varianten von EM-Checklisten, welche bisher am Fertigungsstandort eingehen, gehören somit der Vergangenheit an. Eine weitere Verbesserung besteht darin, dass die erfassten Bestellungen nicht mehr redundant in Form von MS Word- bzw. MS Excel-Dokumenten an den unterschiedlichen Arbeitsplätzen der Vertriebsmitarbeiter und der an der Bemusterung beteiligten Mitarbeiter vorgehalten werden. Der grundlegende Ablauf der Erfassung und Verteilung der EM-Bestellungen ändert sich nicht. Eine Bestellung wird in der Regel auch zukünftig von einem Mitarbeiter des technischen EA-Vertriebs angelegt und von einem Mitarbeiter des kaufmännischen EA-Vertriebs abgeschlossen, allerdings wird die Bestellung nicht mehr durch MS Outlook sondern aus ISDOS heraus an das Werk übermittelt.

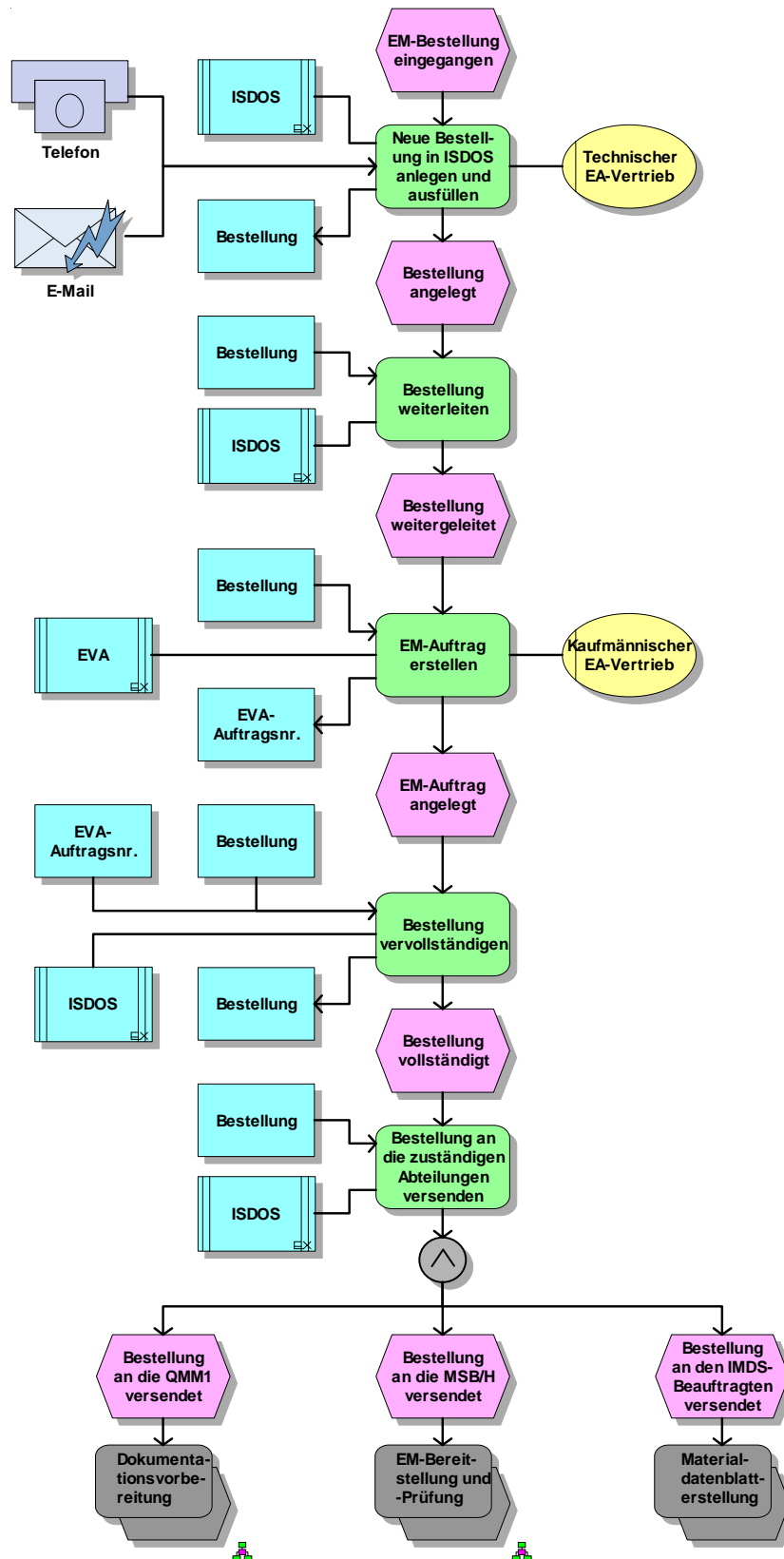


Abb. 4.5: eEPK zur Abwicklung der EM-Bestellung (Soll-Zustand)

Der Ablauf der Materialdatenblatt-erstellung bleibt fast unverändert, weshalb an dieser Stelle auf die Darstellung des Teilprozesses verzichtet wird. Der einzige Unterschied

besteht darin, dass der IMDS-Beauftragte über das zu entwickelnde Websystem auf die von ihm benötigten Daten der EM-Bestellung zugreift.

Im Gegensatz zu den zuvor beschriebenen Teilprozessen ist die Dokumentationsvorbereitung größeren Veränderungen unterworfen (s. Abb. 4.6). Die Mitarbeiter der QMM1 müssen zukünftig nicht mehr überprüfen, ob alle an der Bemusterung beteiligten Abteilungen und Stellen über die Bestellung informiert wurden, da dies durch fest in ISDOS hinterlegte Verteilerlisten gewährleistet wird. Auch müssen die Daten der Bestellung nicht mehr per Hand in die EM-Übersichtsliste übernommen und ausgedruckte EM-Checklisten in einem physischen Ordner abgelegt werden. Die EM-Übersichtsliste wird in Zukunft durch ein Anwendungssystem namens Generator abgelöst. Dieses befindet sich ebenfalls noch in der Entwicklung und wird es ermöglichen, dass ausgewählte Daten einer EM-Bestellung aus der ISDOS-Datenbank (ISDOS-DB) per Knopfdruck in die Datenbank des Generators importiert werden können. Mit dem Generator können dann z. B., wie es bisher mit der EM-Übersichtsliste möglich ist, alle noch zu bearbeitenden Bemusterungen angezeigt werden.

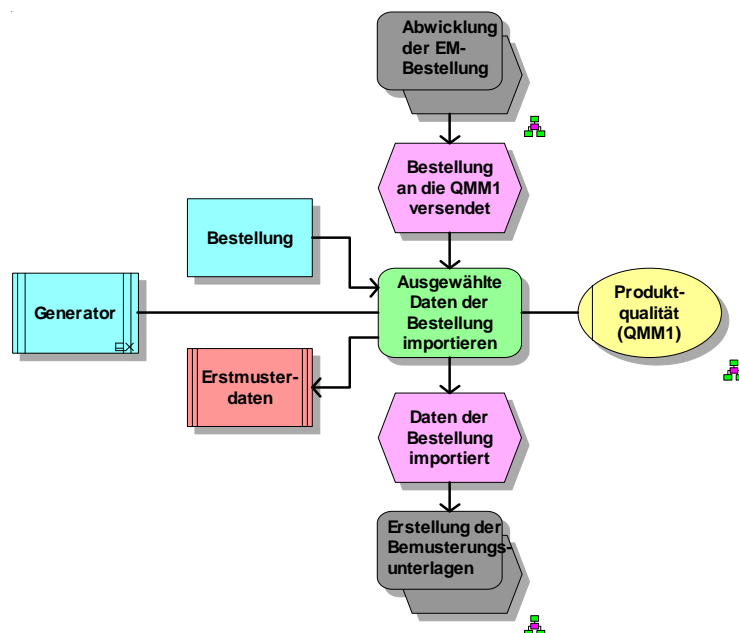


Abb. 4.6: eEPK zur Dokumentationsvorbereitung (Soll-Zustand)

Die EM-Bereitstellung und -Prüfung in der MSB/H ist ebenfalls einigen Änderungen unterworfen (s. Abb. 4.7). Die Daten der EM-Bestellung müssen auch hier zukünftig nicht mehr ausgedruckt und per Hand in den Datenbank-Explorer übertragen werden. Vielmehr soll dieser wie das Anwendungssystem der QMM1 Zugriff auf einen ausgewählten Bereich der ISDOS-DB haben, so dass die benötigten Daten auf Knopfdruck importiert werden können. Die eigentliche Bereitstellung und Prüfung der Erstmuster

verläuft wie bisher, nur werden die hierfür benötigten Informationen durch das web-basierte Informationssystem ISDOS und den Datenbank-Explorer bereitgestellt.

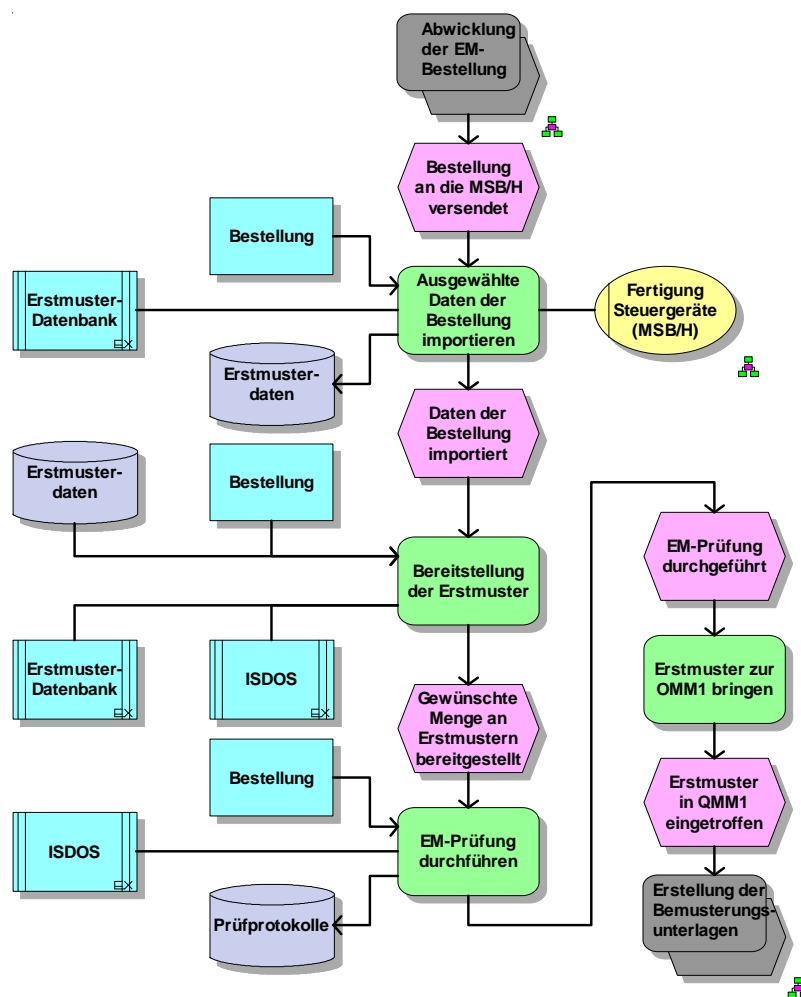


Abb. 4.7: eEPK zur EM-Bereitstellung und -Prüfung (Soll-Zustand)

Der grundlegende Ablauf der Erstellung der Bemusterungsunterlagen bleibt auch in Zukunft derselbe (s. Abb. 4.8), allerdings werden die Sachbearbeiter der QMM1 durch zwei neue Anwendungssysteme bei ihrer Arbeit unterstützt. Die EM-Übersichtsliste wird wie bereits erwähnt vom Generator abgelöst. Stehen mehrere Bemusterungen an, kann sich der zuständige Sachbearbeiter mit diesem Anwendungssystem alle Bemusterungen nach dem Zustellungsdatum geordnet anzeigen lassen und so die nächste zu bearbeitende auswählen. Die Suche nach der zugehörigen EM-Bestellung erfolgt nun nicht mehr in dem physischen Ordner, sondern mit ISDOS. Darüber hinaus werden die Sachbearbeiter zukünftig bei der Erstellung der Bemusterungsunterlagen dahingehend unterstützt, dass ein Großteil der vom Kunden gewünschten Dokumente mit dem Generator automatisch erstellt und vorausgefüllt werden. Die Voraussetzung hierfür ist, dass alle benötigten Daten der Bestellung im Rahmen der Dokumentationsvorbereitung aus der ISDOS-DB importiert worden sind. Die vom Generator erzeugten MS Word- bzw. MS

Excel-Dokumente müssen anschließend nur noch vom Sachbearbeiter vervollständigt, archiviert und ausgedruckt werden. Die Daten einer EM-Bestellung verbleiben für den gleichen Zeitraum wie die elektronisch archivierten Bemusterungsunterlagen in der ISDOS-DB.

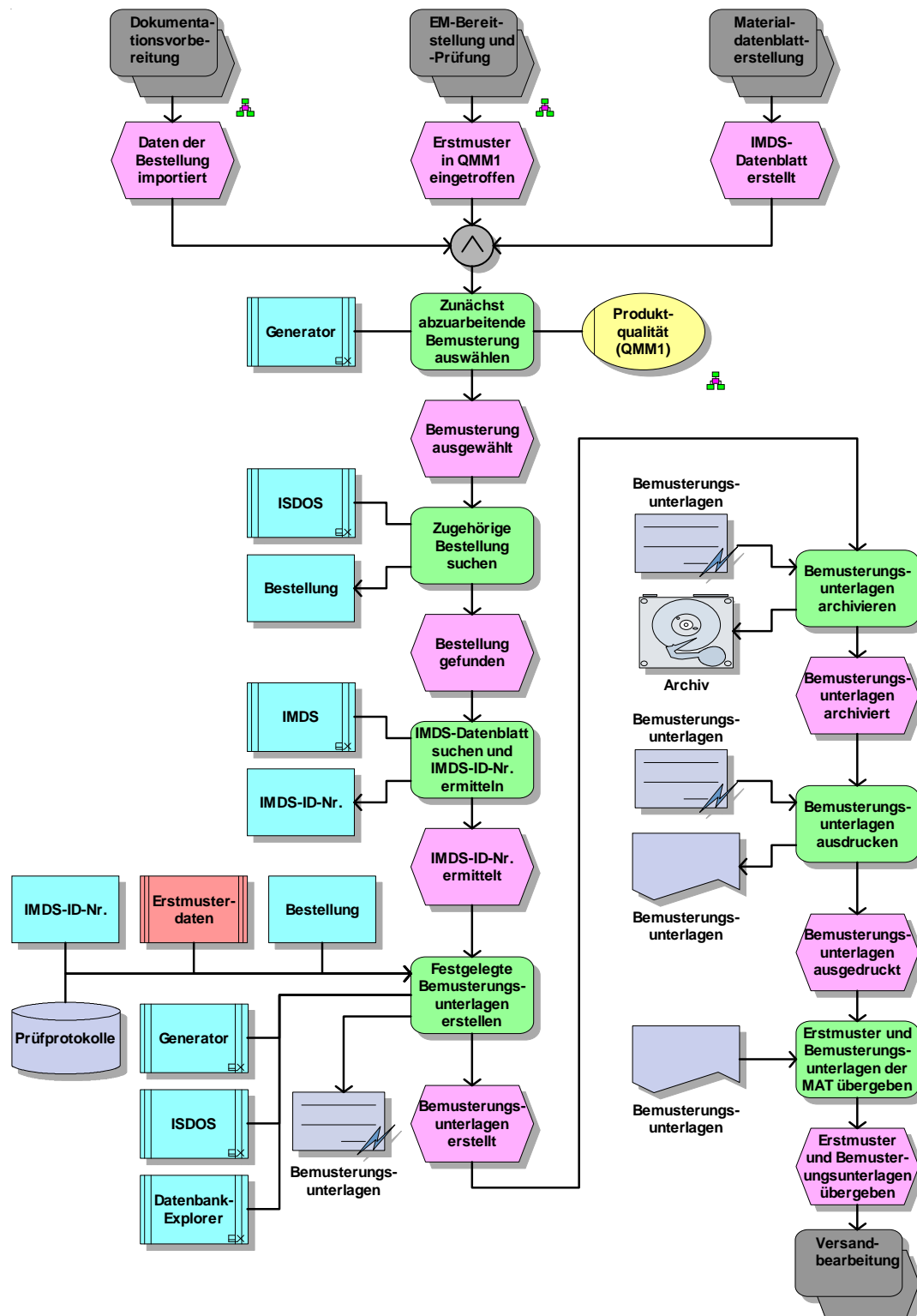


Abb. 4.8: eEPK zur Erstellung der Bemusterungsunterlagen (Soll-Zustand)

5 Systementwurf

Nachdem im vorherigen Kapitel die Anforderungen an das zu entwickelnde Websystem festgelegt wurden, kann nun mit dem Entwurf des Systems begonnen werden. Der Entwurf ist die Entwicklungsphase, in welcher das fachliche Soll-Konzept auf die konkrete, durch die hardware- und softwarebezogenen Anforderungen vorgegebene Computerplattform umgesetzt wird. An dieser Stelle ist somit die Frage nach dem „Wie“ zu beantworten und das zu entwickelnde System aus Sicht der zur Realisierung vorgesehenen Programmiersprachen etc. detailliert zu beschreiben.

Im Folgenden wird zuerst die Systemarchitektur festgelegt. Die nachfolgenden Gliederungspunkte gehen auf den Entwurf der relationalen Datenbank, der Applikation sowie der Benutzeroberfläche ein. Anschließend werden verschiedene Konzepte zur Internationalisierung und Lokalisierung von Websystemen vorgestellt, wobei letztlich das für dieses Entwicklungsvorhaben zum Einsatz kommende gesondert betrachtet wird. Im letzten Abschnitt soll die Sicherheit des Informationssystems thematisiert werden.

5.1 Gesamtarchitektur

Die grundlegende Aufgabe beim Entwurf stellt die Konstruktion der Systemarchitektur dar (vgl. Dumke/Lother/Wille/Zbrog 2003, S. 139). Die Architektur³⁸ des Websystems wird dabei in erster Linie von den funktionalen und nicht-funktionalen Anforderungen beeinflusst. Neben diesen Anforderungen sind u. a. auch Randbedingungen wie die verwendete Systemsoftware (z. B. das verwendete Betriebssystem) oder Verteilungsaspekte (z. B. die Verteilung auf unterschiedliche Standorte im Unternehmen) zu berücksichtigen (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 80).

Architekturen können in so genannten Schichtenmodellen beschrieben werden, wobei eine Schicht (engl. tier) jeweils bestimmte Funktionsbereiche eines Systems unter technologischen Aspekten zusammenfasst (vgl. Dumke/Lother/Wille/Zbrog 2003, S. 141). Für das zu entwickelnde Websystem soll eine 3-Tier-Architektur zugrunde gelegt werden, wobei diese aus den einzelnen Schichten:

- Präsentation (Webbrowser),
- Applikation (Webserver mit Applikationslogik)

³⁸ Die Architektur eines Websystems ist die software- und hardwarebezogene Struktur, welche die Komponenten dieser Struktur sowie deren Schnittstellen und Beziehungen umfasst (vgl. Dumke/Lother/Wille/Zbrog 2003, S. 141).

- und Datenhaltung (Datenbankserver) besteht.

Zur Veranschaulichung wird die Systemarchitektur in Form eines Verteilungsdiagramms (s. Abb. 5.1) der UML dargestellt. Dieses zeigt die technische Systemumgebung und ermöglicht eine erste Übersicht der Verteilung und Vernetzung von Software- und Hardwarekomponenten im zukünftigen Websystem.

Wie in Abb. 5.1 zu sehen ist, werden zwei Serverarten verwendet, wobei sich der Webserver und der Datenbankserver (DB-Server) auf verschiedenen Rechnern befinden. Der Webserver beinhaltet u. a. die ASP.NET-Webforms und die Applikationslogik, der DB-Server die relationale Datenbank (ISDOS-DB). Über einen Webbrowser auf dem Client-Rechner findet die Interaktion zwischen Benutzer und Informationssystem statt.

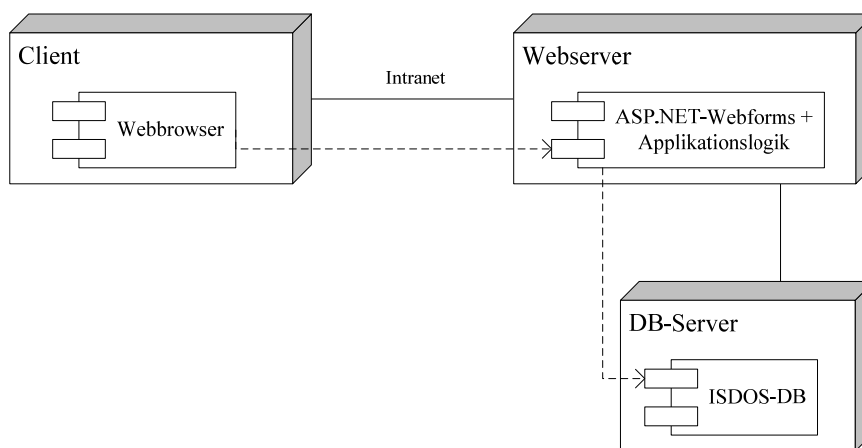


Abb. 5.1: Systemarchitektur als Verteilungsdiagramm

ASP.NET ist ein serverseitiges Programmiermodell für Webanwendungen, wobei ASP.NET-Webforms eine spezielle Technologie zur Realisierung von serverseitigen Webanwendungen mit Benutzerschnittstelle zur Darstellung im Webbrowser sind. Eine Webanwendung besteht in der Regel aus einer Abfolge von mehreren Webseiten. ASP.NET-Webforms sind dynamische Webseiten mit der Dateierweiterung `.aspx` (im Folgenden auch als ASPX-Datei bezeichnet), welche neben konventionellem HTML³⁹-Code als Ausgabeformat auch so genannte Serversteuerelemente⁴⁰ beinhalten (vgl. Schwichtenberg 2006, S. 194).

³⁹ Die Hypertext Markup Language (HTML) ist eine Auszeichnungssprache und dient der Strukturierung und Präsentation von meist statischen Inhalten wie Texten, Bildern und Verweisen in einem Webbrowser. HTML definiert eine Anzahl von so genannten Tags zur Auszeichnung (engl. markup) unterschiedlichster Semantiken (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 140 f.). Für eine ausführliche Beschreibung von HTML sei auf die zahlreich erhältlichen Literaturquellen verwiesen.

⁴⁰ Serversteuerelemente sind spezielle Tags mit dem Attribut `runat="server"`, welche nicht 1:1 an den Client übertragen, sondern auf dem Server in browserspezifische Tags umgewandelt werden. Es

5.2 Relationaler Datenbankentwurf

In diesem Abschnitt wird das konzeptionelle Datenbankmodell (s. Abschnitt 4.3), welches die Datenstrukturen des zukünftigen Websystems realisierungsunabhängig beschreibt, auf ein konkretes Datenbankmodell abgebildet. Da der Datenbestand von ISDOS in einer relationalen Datenbank verwaltet werden soll, bedeutet dies, dass aus dem konzeptionellen Modell ein Relationenmodell abzuleiten ist.⁴¹

Im Relationenmodell, dem zurzeit am weitesten verbreiteten Datenbankmodell, werden die Daten als flache (ungeschachtelte) Tabellen dargestellt. Die erste Zeile, das so genannte Relationenschema, gibt jeweils die Struktur einer Tabelle (Anzahl und Benennung der Spalten) an. Die weiteren Einträge in der Tabelle werden als Relation zu diesem Schema, die einzelnen Zeilen als Tupel und die Spaltenüberschriften als Attribute bezeichnet, wobei jedem Attribut ein Wertebereich (meist Standarddatentypen wie Integer oder String) zugeordnet ist. Eine Relation wird nicht nur mittels Relationenschema, also der Menge von Attributen, sondern auch mit einem Relationennamen deklariert. Beziehungen zwischen beliebigen Datensätzen (Tupeln) werden ausschließlich über Wertegleichheit der Datenfelder ausgedrückt (vgl. Heuer/Saake 2000, S. 11 und S. 106).

Bevor nachfolgend die dem Websystem zugrunde liegende Tabellenstruktur vorgestellt wird, soll kurz die theoretische Vorgehensweise beim relationalen Datenbankentwurf beschrieben werden.

5.2.1 Grundlegende Vorgehensweise

Die Vorgehensweise beim logischen Datenbankentwurf basiert grundsätzlich auf zwei Schritten (vgl. Heuer/Saake 2000, S. 178, S. 190 ff. und S. 238 ff.):⁴²

Der erste Schritt besteht aus der Abbildung des konzeptionellen Modells in ein Zioldatenbankmodell. Bei der Transformation eines ERM ins Relationenmodell wird jeder Entity- und Beziehungstyp jeweils auf ein Relationenschema abgebildet. Die Schlüssel- und Nichtschlüsselattribute werden bei Entitytypen den jeweiligen Relationenschemata unverändert zugeordnet. Sind mehrere Schlüssel⁴³ vorhanden, wird ein Primärschlüssel⁴⁴

wird erst zur Laufzeit entschieden, ob das Ausgabeformat reines HTML oder HTML mit Scriptcode (Dynamic HTML (DHTML)) ist (vgl. Schwichtenberg 2006, S. 198 f.).

⁴¹ Die Datenmodellierung auf der Grundlage eines bestimmten Datenbankmodells wird als logische Modellierung bezeichnet.

⁴² Im Rahmen dieser Arbeit kann nicht auf alle Aspekte dieser Thematik detailliert eingegangen werden. Vielmehr soll ein kurzer Überblick gegeben werden. Für nähere Informationen sei an dieser Stelle auf die angegebene Literatur verwiesen.

⁴³ Ein Schlüssel ist eine minimale Attributmenge, deren Werte eine Zeile einer Tabelle eindeutig identifizieren (vgl. Heuer/Saake 2000, S. 115).

gewählt. Bei Beziehungstypen werden nicht nur die zugehörigen Attribute sondern auch die Primärschlüssel der beteiligten Entitytypen in das Relationenschema übernommen. Die Kardinalitäten der abgebildeten Beziehungen werden in der relationalen Darstellung durch Wahl der Schlüssel beim zugehörigen Relationenschema ausgedrückt.⁴⁵ In bestimmten Fällen besteht die Möglichkeit, die Relationenschemata von Entity- und Beziehungstypen miteinander zu verschmelzen.⁴⁶ Bei der Abbildung von Beziehungstypen sind die Primärschlüssel der beteiligten Entitytypen im Relationenschema der Beziehung als Fremdschlüssel⁴⁷ auszuweisen. Zwischen den Relationenschemata werden somit diverse Fremdschlüsselbedingungen eingeführt.

Der zweite Schritt beinhaltet im Allgemeinen eine Optimierungsaufgabe, bei welcher unterschiedliche Optimierungsziele miteinander konkurrieren können (z. B. redundanzfreie Speicherung durch Aufteilung auf mehrere Relationen gegenüber schnellerem Zugriff bei redundanter Speicherung). Die Verbesserung des relationalen Schemas (also der Sammlung von Relationen) durch die Minimierung von Redundanzen wird als Normalisierung bezeichnet. Bei diesem Vorgang werden die vorhandenen Relationen meist auf drei so genannte Normalformen hin überprüft und diese ggf. durch Umstrukturierungen in den Relationen herbeigeführt. Allgemein können die Normalformen als formale Zwischenschritte auf dem Weg zu einem redundanzfreien Schema verstanden werden.

5.2.2 Abbildung auf das relationale Modell

Unter Berücksichtigung der zuvor beschriebenen theoretischen Vorgehensweise wurde das eERM aus Abschnitt 4.3.3 auf insgesamt 34 Relationen abgebildet. In Abb. 5.2 und 5.3 sind die einzelnen Tabellen mit deren Namen und Attributen in Form eines Datenbankmodellidiagramms dargestellt.⁴⁸

Die Tabelle TBL_ORDER bildet den Kern des relationalen Datenbankmodells, mit der nahezu alle Tabellen entweder direkt oder indirekt verbunden sind, meist über 1:n-Beziehungen. Die einzelnen Beziehungen sind in den Abbildungen als Pfeile darge-

⁴⁴ Ein Primärschlüssel ist ein ausgezeichnete Schlüssel (vgl. Heuer/Saake 2000, S. 115).

⁴⁵ Bei einer m:n-Beziehung werden bspw. beide Primärschlüssel zusammen Schlüssel im neuen Relationenschema (vgl. Heuer/Saake 2000, S. 195).

⁴⁶ Bei einer 1:n-Beziehung muss kein extra Relationenschema angelegt werden. Vielmehr kann das Entity-Relationenschema der n-Seite in das Relationenschema der Beziehung integriert werden (vgl. Heuer/Saake 2000, S. 195).

⁴⁷ Ein Fremdschlüssel ist eine Attributmenge, welche in einer anderen Tabelle Schlüssel ist (vgl. Heuer/Saake 2000, S. 115).

⁴⁸ Im Rahmen des relationalen Datenbankentwurfs sind für die Tabellen- und Attributnamen ausschließlich englische Bezeichnungen eingeführt worden.

stellt, wobei ein an einer Beziehung teilnehmendes Attribut einer Tabelle als Primärschlüssel, das andere als Fremdschlüssel gekennzeichnet ist. Die im eERM vorhandenen n:m-Beziehungen wurden im Rahmen des relationalen Datenbankentwurfs jeweils durch die Bildung einer Tabelle für den Beziehungstyp aufgelöst, wobei die Primärschlüssel der beteiligten Entitytypen zusammen den Schlüssel dieser Tabelle bilden. Ein Beispiel hierfür ist die Tabelle TBL_ROLE_HAS_RIGHT, auf welche im Rahmen der Ausführungen zum Sicherheitskonzept (s. Abschnitt 5.6.2) genauer eingegangen wird. Für den Entitytyp Vertriebsmitarbeiter wurden dagegen zwei Tabellen eingeführt. Die erste Tabelle enthält Daten zum Vertriebsmitarbeiter des technischen (TBL_SALES_TECHNICAL) und die zweite zum Mitarbeiter des kaufmännischen Verkaufsbereichs (TBL_SALES_COMMERCIAL). Gleiches gilt für den Entitytyp Kennzeichnungen und Aufkleber, welcher ebenfalls auf zwei Tabellen, eine für die Kennzeichnung und Aufkleber der Verpackung (TBL_LABEL_PACKAGE) und eine für die der Erstmuster (TBL_LABEL_PRODUCT), aufgeteilt wurde. Die letztere Aufteilung soll gewährleisten, dass zukünftige Erweiterungen bzw. Anpassungen, z. B. nur für die Kennzeichnungen und Aufkleber der Produkte, mit geringem Aufwand umsetzbar sind. Darüber hinaus sind im Rahmen der Normalisierung bestimmte Attribute der für die Entitytypen Bestellung, Recht, PPF und PPAP eingeführten Tabellen TBL_ORDER, TBL_RIGHT, TBL_REPORT_PPA und TBL_REPORT_PPAP in separate Tabellen ausgelagert worden, welche zu den Tabellen, aus denen sie stammen, in Beziehung stehen. Die meisten dieser Tabellen sind zudem mit der Übersetzungstabelle (TBL_I18N) verbunden, auf welche im Abschnitt 5.5.3 ausführlicher eingegangen wird. Dies ermöglicht es, dass die Inhalte dieser Tabellen leicht in Abhängigkeit von der im Websystem geltenden Spracheinstellung ausgelesen und an die im Rahmen der Implementierung zur Anwendung kommenden ASP.NET-Serversteuerelemente, wie bspw. Auswahl-, Listen- und Optionsfelder angebinden werden können.⁴⁹ Die Daten zu einer gespeicherten EM-Bestellung können schließlich durch eine Datenbankabfrage mit mehreren Verbundbedingungen (Join-Operationen) ermittelt werden. Damit ein schneller Zugriff auf eine Bestellung gewährleistet ist, sind im Rahmen der Normalisierung nicht alle Attribute der Tabelle TBL_ORDER ausgelagert worden, deren Auslagerung möglich wäre.

Darüber hinaus konnten nicht alle Konzepte des eERM im relationalen Datenbankmodell abgebildet werden. So ist etwa die vollständige Partitionierung von Bemusterungsverfahren in PPF- und PPAP-Verfahren sowie kundenspezifische Bemusterungen, wobei eine Bemusterung nur nach einem der drei Verfahren stattfinden kann, mit Schlüsseln und Fremdschlüsseln nicht darstellbar, was bedeutet, dass die hierarchische Struktur nachgebildet werden musste.

⁴⁹ Die Serversteuerelemente sowie die Vorgehensweise bei der Datenbindung werden im Kapitel zur Realisierung des Websystems näher betrachtet.

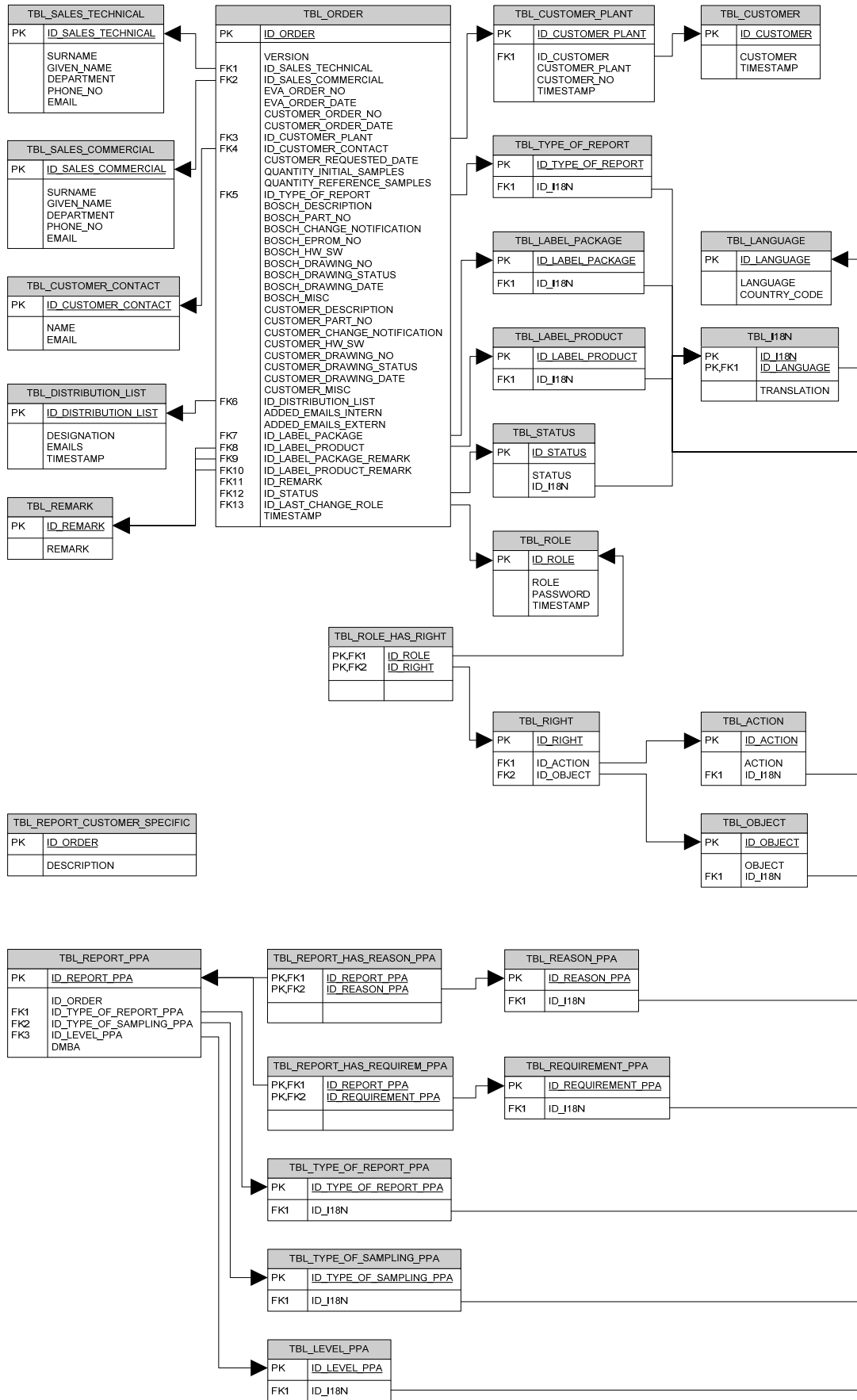


Abb. 5.2: Datenbankmodellldiagramm – Teil 1

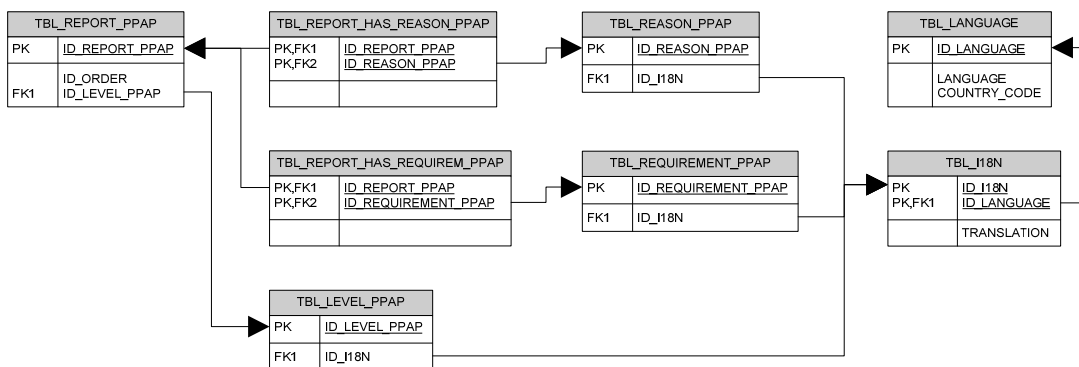


Abb. 5.3: Datenbankmodellldiagramm – Teil 2

In diesem Zusammenhang galt es zu entscheiden, ob (vgl. Alpar/Grob/Weimann/Winter 2005, S. 309 f.):

1. alle Entitytypen der Hierarchie in einer Tabelle zusammengefasst werden,
2. nur Tabellen für die untergeordneten Entitytypen gebildet werden, in welche die Attribute des übergeordneten Entitytyps „hineinvererbt“ werden, oder
3. eine Tabelle für jeden Entitytyp der Hierarchie gebildet wird, zwischen denen bestimmte Abhängigkeiten abzubilden sind.

Ein Nachteil, den alle Realisierungsmöglichkeiten gemeinsam haben, besteht darin, dass das Modellierungskonzept der vollständigen Partitionierung verloren geht. Im Rahmen dieses Entwicklungsprojekts ist die Entscheidung auf die dritte Alternative gefallen, da bspw. im ersten Fall in der erzeugten Tabelle ein jeder Datensatz für eine gewisse Anzahl von Attributen keine Werte haben würde. Die zweite Realisierungsmöglichkeit wurde nicht gewählt aufgrund der fehlenden Tabelle für den übergeordneten Entitytyp und des verloren gehenden Zusammenhangs zwischen den „speziellen“ Tabellen. Dieser Zusammenhang besteht auch bei der dritten Alternative im Datenbankmodell nicht. Da in diesem Fall jedoch eine Tabelle für den übergeordneten Entitytyp gebildet wird, ist es möglich, die bestehenden Abhängigkeiten zwischen den Tabellen im Programmcode der Webanwendung herzustellen.

So werden in dieser Arbeit der übergeordnete Entitytyp durch die Tabelle TBL_TYPE_OF_REPORT und die untergeordneten Entitytypen durch die drei Tabellen TBL_REPORT_PPA, TBL_REPORT_PPAP und TBL_REPORT_CUSTOMER_SPECIFIC repräsentiert. Beim Speichern einer EM-Bestellung wird die Bemusterungsverfahrensart, welche zuvor aus der Tabelle TBL_TYPE_OF_REPORT ausgelesen wurde, zusammen mit den Informationen der Bestellung gespeichert. Die Vereinbarungen zum gewählten Bemusterungsverfahren werden hierbei in einer der drei „speziellen“ Tabel-

len, gemeinsam mit der eindeutigen ID der Bestellung, abgelegt. Für den Zugriff auf die Daten eines festgelegten Bemusterungsverfahrens einer Bestellung sind wiederum zwei Schritte erforderlich. Zuerst muss die Art des Bemusterungsverfahrens ermittelt werden. Anschließend wird mit der ID der Bestellung auf den Datensatz, in der zu dem Bemusterungsverfahren gehörenden Tabelle, zugegriffen.

5.3 Entwurf der Applikation

In diesem Abschnitt wird ein kurzer Überblick zu den einzelnen Dateien gegeben, aus denen die zukünftige Webanwendung besteht. Auf die Verzeichnisstruktur zur Ablage dieser Dateien soll jedoch erst im Rahmen der Realisierung näher eingegangen werden. Darüber hinaus werden im Folgenden einige Klassen, die bestimmte Teile der Funktionalität bereitstellen, kurz vorgestellt. Eine ausführliche Betrachtung der Methoden und Attribute der einzelnen Klassen findet an dieser Stelle aus Platzgründen nicht statt. Vielmehr soll der Grundgedanke der Implementierung vermittelt werden.

5.3.1 Dateien

Bei der Vorstellung der Gesamtarchitektur (s. Abschnitt 5.1) wurde bereits erwähnt, dass die Webanwendung mittels ASP.NET-Webforms realisiert wird. Der Programmcode einer ASPX-Datei soll hierbei nicht im HTML-Code verschachtelt, sondern in einer separaten Hintergrundcodedatei mit der Dateierweiterung `.aspx.vb` vorgehalten werden. Diese Trennung von Gestaltung (HTML-Code) und Anwendungslogik (VB-Code) wird als Code-Behind-Modell bezeichnet (vgl. Schwichtenberg 2006, S. 44). Dieses Konzept trägt dazu bei, dass die Übersichtlichkeit erheblich gesteigert wird und ein strukturierter Programmcode entsteht. Darüber hinaus soll die Webanwendung aus eigenständigen Codedateien (`.vb`), JavaScript-Dateien (`.js`), Webkonfigurationsdateien (`.config`), Ressourcendateien (`.resx`), Stylesheet-Dateien (`.css`) und Grafiken (`.gif`) bestehen. Im weiteren Verlauf der Arbeit wird auf die verschiedenen Dateitypen im Zusammenhang mit ihrer jeweiligen Verwendung ausführlicher eingegangen.

5.3.2 Klassen

Bei dem zu entwickelnden Webprojekt wird die gesamte Applikationslogik in Hintergrundcodedateien sowie in eigenständigen, von den Webforms unabhängigen Codedateien vorgehalten. Eine Hintergrundcodedatei enthält dabei eine partielle Klasse, wel-

che die Programmierlogik, in der Regel Ereignisbehandlungsroutinen sowie beliebige Methoden und Attribute, für eine zugehörige ASPX-Datei bereitstellt. Eine solche partielle Klasse erbt üblicherweise von der Klasse `System.Web.UI.Page`⁵⁰ bzw. von einer von dieser Klasse abgeleiteten Klasse. Bei einer partiellen Klasse handelt es sich um keine vollständige Klassendefinition, da sie nur einen Teil des gesamten Programmcodes enthält. Erst beim Kompilieren, also bspw. beim ersten Ausführen bzw. Aufrufen einer Webseite, wird die ASPX-Datei mit der zugehörigen Hintergrundcodedatei von der ASP.NET-Laufzeitumgebung zu einer einzigen Klasse zusammengeführt (vgl. Microsoft Deutschland GmbH 2005b). Jede ASP.NET-Webseite ist somit zur Laufzeit die Instanz einer Klasse, welche daher auch als Seitenklasse bezeichnet wird. Da ein jedes Webform (direkt oder indirekt) von der `Page`-Klasse abgeleitet ist, stehen in allen Routinen innerhalb der Seitenklasse alle Methoden und Attribute der Klasse `Page` zur Verfügung (vgl. Schwichtenberg 2006, S. 488 ff.).

Im Rahmen dieses Entwicklungsvorhabens werden diverse Webforms erstellt, wobei die partiellen Klassen in den zugehörigen Hintergrundcodedateien wesentliche Teile der Funktionalität des Systems bereitstellen. So sollen bspw. jeweils ein Webform für das Erfassen, Öffnen, Versenden und Suchen von EM-Bestellungen sowie jeweils eines zum Anzeigen, Anlegen, Ändern und Löschen von Benutzerrollen, Verteilerlisten und Kundendaten umgesetzt werden. Die Startseite der Webanwendung soll ebenfalls als Webform realisiert werden und zugleich den Anmeldedialog bereitstellen.

Um Codedopplungen zu vermeiden und so die Wiederverwertbarkeit sowie eine leichte Erweiterbarkeit und Wartbarkeit des Programmcodes gewährleisten zu können, sollen zudem eigenständige Codedateien erstellt werden. Diese Dateien enthalten Hilfsklassen, die jeweils einen eigenen Funktionsbereich abdecken. Solche Bereiche sind bspw. die Datenbankkommunikation, die Authentifizierung und Autorisierung sowie die Mehrsprachigkeit. Die Hilfsklassen können dabei auch untereinander auf die implementierten Methoden zugreifen, was eine effiziente Mehrfachnutzung desselben Programmcodes ermöglicht. Darüber hinaus werden so die wichtigen Kernfunktionen gegenüber den Seitenklassen gekapselt, was ein wesentliches Grundprinzip der objektorientierten Programmierung darstellt.

Im Folgenden wird mit Hilfe eines Klassendiagramms der UML (s. Abb. 5.4) der Grundgedanke der Implementierung verdeutlicht.⁵¹ Im Mittelpunkt steht hierbei bei-

⁵⁰ Die `Page`-Klasse ist die Basisklasse für Webseiten in ASP.NET-Webforms. Sie ist Teil einer umfangreichen Klassenbibliothek, der so genannten .NET Framework Class Library (FCL), und im Namensraum `System.Web.UI` zu finden. Dieser gruppiert alle Klassen, die mit der Gestaltung von web-basierten grafischen Benutzeroberflächen zu tun haben (vgl. Schwichtenberg 2006, S. 512 ff.).

⁵¹ An dieser Stelle sei darauf hingewiesen, dass dieser Abschnitt den Klassenentwurf beschreibt, welcher von der tatsächlichen Implementierung abweichen kann.

spielhaft das Webform, mit welchem zukünftig der Name, das Kennwort und die Zugriffsrechte einer Benutzerrolle geändert werden können. Ein Teil der in diesem Zusammenhang benötigten Funktionalität wird durch die partielle Klasse `role_edit` bereitgestellt. Wie in dem Klassendiagramm zu sehen, ist die Funktionalität auf verschiedene Ereignisbehandlungsroutinen aufgeteilt, wobei die so genannte Seitenereignisroutine `Page_Load` bei jedem Seitenaufruf ausgeführt wird. Die drei Serversteuerereignisse werden dagegen jeweils beim Betätigen einer auf der Webseite befindlichen Schaltfläche ausgelöst. Der Teil der Funktionalität, welcher sowohl in dem Webform zum Ändern als auch in denen zum Anzeigen, Anlegen und Löschen der Benutzerrollen benötigt wird, soll in der Klasse `clsUtilRole` vorgehalten werden. Dies betrifft z. B. die Programmierlogik zum Holen und Anzeigen der aktuellen Daten einer Benutzerrolle. Darüber hinaus werden im Rahmen dieses Entwicklungsvorhabens weitere Klassen implementiert, auf deren Methoden von jedem Webform und jeder Klasse aus zugegriffen werden kann. Die Klasse `clsDBManager` stellt hierbei Methoden und Attribute zur Kommunikation mit der Datenbank und die Klasse `clsI18N` Methoden zur Umsetzung der Mehrsprachigkeit bereit. Die Klasse `clsAuthFilter` kapselt die Funktionalität, welche zur Überprüfung der Authentifizierung und Autorisierung eines Systembenutzers benötigt wird.

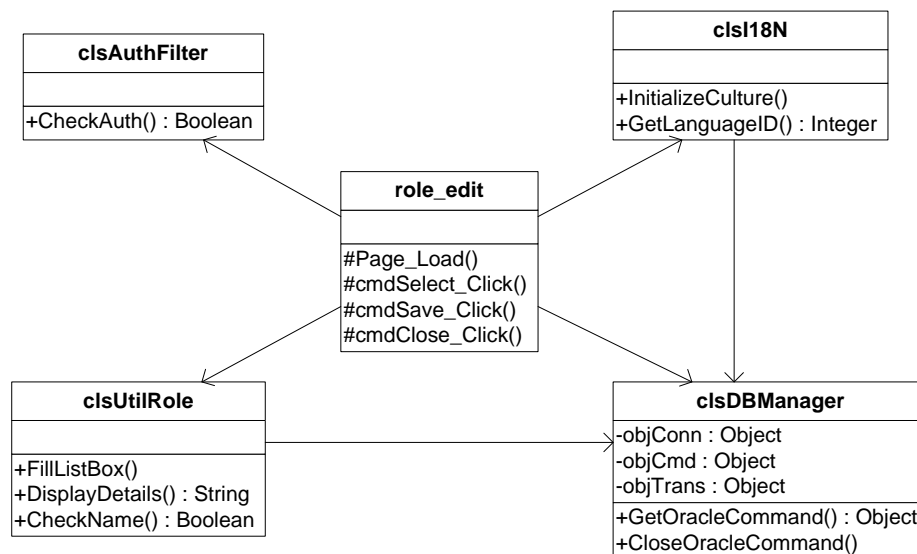


Abb. 5.4: Ausschnitt aus dem Klassendiagramm der Webanwendung

5.4 Entwurf der Benutzeroberfläche

Eine Benutzeroberfläche ist eine aus Softwarekomponenten bestehende Schnittstelle zwischen Benutzer und Informationssystem (vgl. Mertens 2001, S. 67). Benutzeroberflächen legen das Erscheinungsbild und die Bedienung eines Systems fest und können

grundsätzlich in alphanumerische (z. B. kommandoorientierte) und grafische Benutzeroberflächen (engl. Graphical User Interface (GUI)) unterschieden werden (vgl. Kruschinski 1999, S. 1 f.). Verbreitet sind heutzutage vor allem die GUIs, welche eine Benutzerinteraktion bzw. -information in Form von grafischen Elementen, wie bspw. Symbolen, Fenstern, Dialogfeldern, Schaltflächen und Menüs unter Verwendung eines Zeige- und Steuerungsinstruments, z. B. einer Maus, ermöglichen. Bei webbasierten Informationssystemen werden die GUIs mittels Webbrowser dargestellt (vgl. Mertens 2001, S. 68).

Im Folgenden werden exemplarisch einige allgemeine Gestaltungsrichtlinien für GUIs von Websystemen beschrieben, wobei in diesem Zusammenhang auf den Begriff Usability eingegangen wird. Anschließend sollen Layout und Navigationselemente der GUIs von ISDOS vorgestellt werden.

5.4.1 Allgemeine Gestaltungsrichtlinien

Der Gestaltung von GUIs muss besondere Beachtung geschenkt werden, da diese häufig entscheidend für die Akzeptanz, die Effektivität und den Erfolg des Websystems sind (vgl. Kruschinski 1999, S. 7). Die Zielsetzung besteht darin die Struktur und das Verhalten der webbasierten GUIs so zu gestalten, dass deren Bedienung intuitiv und leicht erlernbar ist, sowie benötigte Informationen auf einfache Weise zugänglich sind.⁵²

In diesem Zusammenhang wird oft der Begriff Usability verwendet, der speziell die Benutzbarkeit eines Systems betrachtet und sich nach dem Standard ISO 9126⁵³ in die Aspekte Verständlichkeit des Systems, Einfachheit seiner Bedienung und Erlernbarkeit der Systemhandhabung unterteilt (vgl. Dumke/Lothar/Wille/Zbrog 2003, S. 129). Der deutsche Terminus für den Begriff Usability ist Gebrauchstauglichkeit (vgl. Kappel/Pröll/Reich/ Retschitzegger 2004, S. 267) und wird gemäß DIN EN ISO 9241⁵⁴-11 definiert als „[d]as Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ (DIN – Deutsches Institut für Normung e. V. 1999, S. 4)

⁵² Das Fachgebiet, welches sich mit der Gestaltung von Benutzeroberflächen befasst, wird als Software-Ergonomie bezeichnet. Grundlagen hierfür kommen aus der Wahrnehmungs- und Lernpsychologie, den Arbeitswissenschaften und der grafischen Gestaltung (vgl. Böhm/Fuchs 2002, S. 141).

⁵³ Die internationale Norm ISO 9126 „Software Engineering – Product Quality“ enthält Definitionen für wesentliche Begriffe aus dem Bereich Software-Qualität.

⁵⁴ Die Normenreihe DIN EN ISO 9241 mit dem Titel "Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten" beschreibt Anforderungen an die Arbeitsumgebung, Hardware und Software.

Da die Usability einen der wichtigsten Qualitätsfaktoren für Websysteme darstellt, ist es besonders wichtig, die genannten Aspekte beim Systementwurf zu berücksichtigen. Im Folgenden werden einige allgemeingültige Gestaltungsrichtlinien, speziell für web-basierte GUIs, aufgezählt:⁵⁵

- GUIs sollten ein ansprechendes Layout haben. Eines der wichtigsten Kriterien hierfür ist die Konsistenz (vgl. Kruschinski 1999, S. 40). Dies bedeutet, dass jeder Gestaltungsaspekt (z. B. Layout, Farbschema und Zeichensatz) innerhalb eines Websystems einheitlich sein sollte (vgl. DIN – Deutsches Institut für Normung e. V. 1996, S. 6).
- Farben und Schriftarten sollen gezielt eingesetzt werden, um Informationen und Bedeutungen zu transportieren. Der Einsatz extremer Farben (z. B. mit hoher Sättigung) sollte vermieden werden (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 275). Allgemein sollte deren Gebrauch sparsam erfolgen, wobei auf einem Bildschirm nicht mehr als vier bis fünf verschiedene Farben und zwei bis drei Schriftarten zu verwenden sind (vgl. Böhm/Fuchs 2002, S. 150).
- Auf den Einsatz von Animationen sollte, sofern diese keine spezifische Bedeutung haben, verzichtet werden, da sie die Aufmerksamkeit des Benutzers auf sich ziehen und von den wesentlichen Informationen ablenken (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 279).
- Texte sollten in einer prägnanten und aufgabenbezogenen Sprache formuliert werden (vgl. Böhm/Fuchs 2002, S. 145). Wichtig ist vor allem eine sprachliche Angemessenheit, wobei nach Möglichkeit auch die jeweilige Landessprache des Benutzers angeboten werden sollte (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 276).
- Informationen und Funktionen sollten möglichst direkt ersichtlich sein (vgl. Böhm/Fuchs 2002, S. 144), wobei dem Benutzer nur relevante Informationen gezeigt werden sollten, die im Zusammenhang mit der Erledigung einer Aufgabe stehen (vgl. DIN – Deutsches Institut für Normung e. V. 1996, S. 4).
- Zur besseren Orientierung ist sicherzustellen, dass kein horizontaler Bildlauf (engl. scrolling) innerhalb der GUI vonnöten ist, da zeilenweises Lesen ein permanentes Hin- und Herschieben erforderlich machen würde. Wichtige Informationen sollten oben links zu finden sein, da sie dort am schnellsten gefunden

⁵⁵ An dieser Stelle sei darauf hingewiesen, dass die Literatur zu diesem Thema sehr umfangreich ist und hier nur ein kurzer Überblick gegeben werden kann. Für weiterführende Informationen sei auf die angegebene Literatur verwiesen.

werden. Menüs und Navigationselemente sind daher in erster Linie am linken bzw. oberen Rand zu positionieren. Generell sind überladene GUIs zu vermeiden (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 276).

- Eine durch die Navigationsstruktur bedingte Desorientierung des Benutzers muss unbedingt vermieden werden (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 277). Der Benutzer muss immer wissen, wo er sich innerhalb der Navigationsstruktur befindet, was er dort tun bzw. finden kann, woher er gekommen ist und wohin er weiter navigieren kann (vgl. Böhm/Fuchs 2002, S. 144).

Für den Entwurf gebrauchstauglicher Websysteme ist es jedoch keinesfalls ausreichend, lediglich die allgemeinen Faustregeln zu beachten, sondern vielmehr ist es notwendig, für jeden einzelnen Fall Ziele, Benutzer und Nutzungskontext konkret zu bestimmen (vgl. Kappel/Pröll/Reich/Retschitzegger 2004, S. 267).

5.4.2 Layout und Navigationselemente

Bevor in diesem Abschnitt das grundlegende Layout der Benutzeroberflächen anhand einer ersten prototypischen Realisierung vorgestellt wird, ist anzumerken, dass neben den zuvor aufgeführten allgemeinen Gestaltungsrichtlinien auch unternehmensspezifische Gestaltungsrichtlinien zu beachten sind. Diese Vorgaben zur Gestaltung von Benutzeroberflächen sind in so genannten Style Guides, welche in der Regel als einfacher Text vorliegen, formuliert und dienen zur Einhaltung der Corporate Identity (CI)⁵⁶ des Unternehmens. So existieren bspw. RB-interne Vorgaben für Layout, Farbgebung und Schriftarten, welche sicherstellen sollen, dass auch Benutzeroberflächen verschiedener Websysteme einheitlich und konsistent sind und vom Benutzer auf den ersten Blick als dem Konzern zugehörend erkannt werden können.

In Abb. 5.5 ist eine erste prototypische Realisierung der Benutzeroberfläche zu sehen. Anhand dieses Prototyps werden nachfolgend der grundlegende Aufbau sowie die einzelnen Komponenten der GUIs von ISDOS beschrieben. An dieser Stelle sei erwähnt, dass das grafische Layout der Benutzeroberflächen schlicht gehalten wird und die Funktionalität sowie die einfache Bedienbarkeit im Vordergrund stehen. Auf grafische Raffinessen wird im Rahmen dieser Entwicklung bewusst verzichtet. Ziel ist es, möglichst „schlanke“ GUIs zu entwerfen, indem z. B. die Anzahl von Grafiken auf den Webseiten niedrig gehalten sowie das Datenvolumen der Grafiken durch Reduktion von Auflösung

⁵⁶ Die CI definiert das Erscheinungsbild eines Unternehmens in der Öffentlichkeit.

oder Farbtiefe minimiert werden.⁵⁷ Wie bereits erwähnt, werden die GUIs beim Webform-Programmiermodell von ASP.NET durch eine Mischung aus HTML-Code und Serversteuerelementen realisiert.

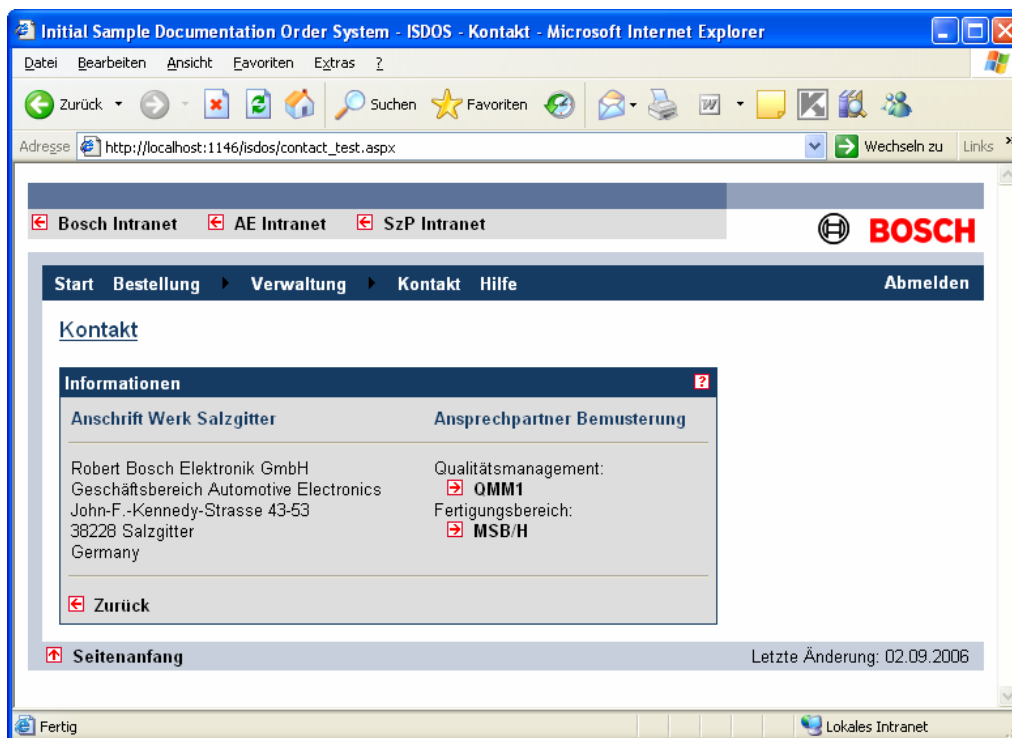


Abb. 5.5: Erste prototypische Realisierung der Benutzeroberfläche

Die Benutzeroberflächen von ISDOS sind grundsätzlich in vier Bereiche unterteilt: Kopf-, Navigations-, Inhalts- und Fußbereich. Wie in Abb. 5.5 zu sehen ist, enthält der Kopfbereich mehrere Elemente. Am rechten oberen Rand, ist das Firmenlogo zu finden. Auf gleicher Höhe befinden sich linksbündig ausgerichtet drei Verweise, wobei über diese zu den Intranet-Startseiten der RB GmbH, des Geschäftsbereichs Automobilelektronik und der RB Elektronik GmbH navigiert werden kann. Verweise sind immer durch ein vorangestelltes Pfeilsymbol und einen fetten Schriftzug gekennzeichnet. Die Pfeilrichtung zeigt dabei die Navigationsrichtung an. Zudem kann ein Verweis ausschließlich durch einen Klick auf das Pfeilsymbol aufgerufen werden.

Der Navigationsbereich enthält auf der linken Seite eine horizontale Navigationsleiste und rechts ausgerichtet einen so genannten LinkButton, mit welchem sich ein eingeloggter Benutzer vom Websystem abmelden oder ein nicht eingeloggter Benutzer den Anmeldedialog aufrufen kann. ASP.NET 2.0 stellt mit dem LinkButton-Steuerelement

⁵⁷ Dieser Ansatz berücksichtigt die unterschiedlichen technischen Gegebenheiten im Bereich der Internetanbindungen der weltweiten Verkaufshäuser und soll möglichst kurze Reaktionszeiten beim Arbeiten mit dem Websystem gewährleisten.

eine Navigationsmöglichkeit zur Verfügung, die im Webbrowser als Verweis dargestellt wird, sich jedoch wie eine Schaltfläche verhält. Zudem verfügt ASP.NET 2.0 über ein Menüsteuererelement, mit welchem eine dynamisch veränderliche Navigationsleiste mit Graufärbung der aktuell deaktivierten Menüpunkte umgesetzt werden kann. Die Navigationssituation passt sich somit den Wahlmöglichkeiten eines Benutzers an, indem nur die Aktionen auswählbar sind, die er auch ausführen darf. Bei der Navigationsleiste handelt es sich um ein so genanntes Pull-Down-Menü, welches durch Auswahl einer Option von oben nach unten aufklappt und eine Liste von weiteren Optionen anzeigt (s. Abb. 5.6).

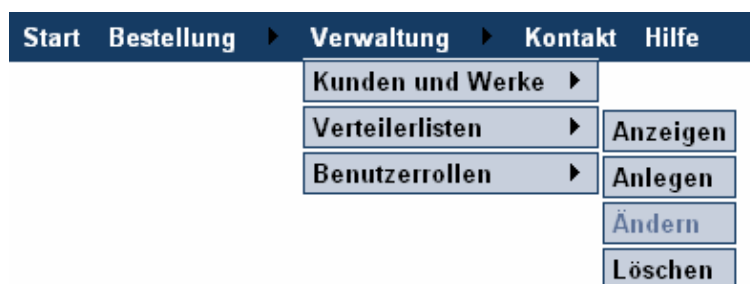


Abb. 5.6: Navigationsleiste des Informationssystems

Der Großteil der Benutzeroberflächen wird durch den Inhaltsbereich ausgemacht. Hier werden bspw. die Informationen zur Verfügung gestellt, die vermittelt werden sollen, sowie die Formulare (auch Masken genannt) dargestellt, über welche der Dialog mit dem Websystem stattfindet. Der grundlegende Aufbau des Inhaltsbereichs ist dabei immer derselbe. Die Inhalte sind grundsätzlich linksbündig ausgerichtet und alle Ein- und Ausgabemasken verfügen über eine Überschrift, die Auskunft darüber gibt, wo der Benutzer sich gerade befindet und was er dort tun kann. An wichtigen bzw. erklärungsbedürftigen Stellen des Inhaltsbereichs sind Symbole mit einem Fragezeichen zu finden, welche beim Anklicken ein Pop-Up-Fenster öffnen, das eine kurze kontextbezogene Hilfe zur Verfügung stellt. Längere Formulare mit vielen Eingabe- und Auswahlfeldern werden nicht über miteinander verknüpfte Webseiten realisiert, da dies zu zusätzlicher Navigation führen würde. Im Rahmen dieser Arbeit wird dem Einsatz von vertikalen Rollbalken gegenüber zusätzlicher Navigation der Vorzug gegeben.

Der Fußbereich enthält zwei Elemente. Auf der linken Seite befindet sich ein Verweis, der den Benutzer zum Seitenanfang führt und rechts ist das Datum der letzten Änderung des jeweiligen Webform zu finden.

Die Gestaltungsmerkmale der webbasierten GUIs werden mittels Cascading Style Sheets (CSS), einer unmittelbaren Erweiterung zu HTML, umgesetzt. Mit Hilfe dieser Technologie lassen sich HTML-Elemente, wie bspw. Tabellen, Textabsätze und Ein-

gabefelder exakt formatieren. Ein wesentliches Leistungsmerkmal von CSS besteht darin, dass die definierten Formateigenschaften, wie etwa Farbgebungen und Schriftarten in einer bzw. wenigen separaten Dateien verwaltet und in beliebig vielen ASP.NET-Webforms referenziert werden können. Erweiterungen und Änderungen an den Formatierungen lassen sich somit an zentraler Stelle durchführen. Im Rahmen dieser Arbeit werden zwei CSS-Dateien verwendet, wobei eine die RB-internen Gestaltungsvorgaben und die andere ergänzende Formateigenschaften enthält.

Darüber hinaus wird zur Durchsetzung eines einheitlichen Grundlayouts für alle Webseiten eine so genannte Masterpage verwendet. Dies ist ein spezielles Webform mit der Dateierweiterung `.master`, das als Vorlage für andere Webforms dient. Eine solche Vorlageseite kann aus einem bzw. mehreren `ContentPlaceHolder`-Elementen bestehen, welche in den von der Masterpage abgeleiteten Seiten durch `Content`-Elemente gefüllt werden (vgl. Schwichtenberg 2006, S. 552). Bei dem zu entwickelnden Webssystem wird der gesamte Kopf-, Navigations- und Fußbereich der GUIs durch eine Masterpage und der spezifische Inhaltsbereich durch diverse `ASPX`-Dateien umgesetzt. Dies hat den Vorteil, dass die Steuerelemente der Vorlageseite auf jeder Webseite erscheinen und mögliche Änderungen bzw. Erweiterungen an eben jenen Steuerelementen sowie am grundlegenden Layout an zentraler Stelle durchgeführt werden können. Für die Positionierung der Steuerelemente sowie die Strukturierung der Inhalte kommen sowohl in der Vorlageseite als auch in den Detailseiten einfache `HTML`-Tabellen zum Einsatz.

5.5 Konzept zur Internationalisierung und Lokalisierung

Bei der Entwicklung eines international einsetzfähigen Websystems sind unterschiedlichste Aspekte in der Entwurfsphase zu berücksichtigen. In diesem Zusammenhang werden oft die Begriffe Internationalisierung⁵⁸ und Lokalisierung⁵⁹ verwendet.

Lokalisierung bezieht sich dabei auf die Anpassung einer Anwendung oder eines Dokuments, um bestimmten Landessprachen bzw. kulturellen Anforderungen zu entsprechen. Der Begriff wird meist synonym für die Übersetzung der Benutzeroberflächen und Dokumentation verwendet. Sind bspw. auch Anpassungen im Bezug auf Datums- und Zeitformate, Symbole, Zeichen, Farben und Grafiken erforderlich, stellt dieser Vor-

⁵⁸ Für den Begriff Internationalisierung wird oft die Abkürzung „i18n“ verwendet, wobei „18“ der Anzahl der Buchstaben zwischen „i“ und „n“ im englischen Wort „internationalization“ entspricht (vgl. Ishida/Miller 2005).

⁵⁹ Für den Begriff Lokalisierung wird oftmals die Abkürzung „l10n“ geschrieben, wobei „10“ der Anzahl der Buchstaben zwischen „l“ und „n“ im englischen Wort „localization“ entspricht (vgl. Ishida/Miller 2005).

gang eine wesentlich komplexere Angelegenheit dar. Internationalisierung entspricht der Entwicklung einer Anwendung oder eines Dokuments derart, dass eine leichte Lokalisierung möglich ist (vgl. Ishida/Miller 2005).

Bevor im Folgenden der Lösungsansatz zur Lokalisierung des zukünftigen Websystems vorgestellt werden soll, wird auf allgemeine Empfehlungen sowie auf einige grundlegende Konzepte zur Übersetzung von Benutzeroberflächen eingegangen.

5.5.1 Allgemeine Empfehlungen

Mehrsprachigkeit stellt eine wesentliche Anforderung an heutige Websysteme dar, weshalb an dieser Stelle kurz einige Empfehlungen für die Entwicklung weltweit einsatzfähiger Anwendungssysteme aufgezählt werden (vgl. Rätzmann 2004, S. 298-323 und Microsoft Deutschland GmbH 2005a).

- Alle lokalisierbaren Ressourcen, also Benutzeroberflächenelemente wie Zeichenfolgen, Fehlermeldungen, Dialogfelder, Navigationselemente etc. sollten nicht hart codiert, sondern vom übrigen Programmcode getrennt werden.
- Zudem sollten Navigationselemente nicht als Grafikobjekte realisiert, genauer betrachtet sollten weder Bilder noch Symbole, die Text enthalten, verwendet werden.
- Auf die Verwendung von zusammengesetzten Zeichenfolgen, die während der Laufzeit aus verketteten Ausdrücken erstellt werden, ist zu verzichten, da aufgrund der Grammatik in anderen Sprachen die Reihenfolge der Ausdrücke nicht zutreffend sein kann.
- Für die Zeichenfolgen sollte auf den Benutzeroberflächen viel Platz vorhanden sein, da manche Ausdrücke in einigen Sprachen 50 bis 75 Prozent mehr Platz benötigen.
- Es ist empfehlenswert, eine professionelle Übersetzung von Sprachexperten anfertigen zu lassen und die Anwendung auf internationalen Betriebssystemen zu testen.

5.5.2 Grundlegende Konzepte

Die grundlegende Idee bei der Lokalisierung von Websystemen besteht somit in der Trennung der lokalisierbaren Zeichenfolgen vom Programmcode. Im Folgenden sollen kurz einige Ansätze hierfür vorgestellt werden.

Eine Möglichkeit besteht darin, alle zu lokalisierenden Zeichenfolgen in separate Dateien für die einzelnen Sprachen auszulagern. Diese Dateien können dann jeweils in eine separate Dynamic Link Library (DLL) kompiliert und zur Laufzeit dynamisch mit dem Programmcode verknüpft werden. Die Erstellung einer Standardtextdatei für jede Sprache ist eine weitere Alternative, wobei diese zur Laufzeit analysiert und in die Anwendung geladen werden kann. Der strukturelle Aufbau solcher Sprachdateien ist üblicherweise derselbe. So wird in diesen Dateien eine jede Zeichenfolge mit einem eindeutigen Schlüssel vorgehalten (vgl. Rätzmann 2004, S. 302 ff.). Zur Laufzeit können die benötigten Zeichenfolgen über den jeweiligen Schlüssel, in Abhängigkeit von der in der Anwendung geltenden Spracheinstellung, aus der richtigen DLL bzw. Standardtextdatei geladen und einem Benutzer angezeigt werden. An dieser Stelle sei erwähnt, dass die letztere der beiden Methoden, was den Zugriff auf die Zeichenfolgen betrifft, weniger effizient ist.

Eine andere Alternative besteht darin, die Lokalisierung mit Hilfe einer Datenbank umzusetzen. In diesem Fall wird bei jedem Aufruf einer Webseite eine Datenbankabfrage durchgeführt, mit welcher alle benötigten Zeichenfolgen, in Abhängigkeit von der geltenden Spracheinstellung ausgelesen und dargestellt werden können. Da dieser Ansatz im Rahmen dieses Entwicklungsvorhabens auch umgesetzt wird (s. Abschnitt 4.3.2), soll im nächsten Abschnitt ausführlicher auf diesen eingegangen werden.

Darüber hinaus können Benutzeroberflächen auch mit der Extensible Markup Language (XML)⁶⁰ lokalisiert werden. Hierbei sind alle Zeichenfolgen in einem zentralen XML-Dokument vorzuhalten, auf welches zur Laufzeit zugegriffen werden kann. Eine mögliche XML-Datenstruktur könnte dabei so wie in Abb. 5.7 aussehen. Durch die Unterteilung in `subgroup`-Knoten wird das XML-Dokument in sich strukturiert, so dass auch bei vielen Zeichenfolgen und Texten die Übersichtlichkeit gewährleistet bleibt. Sollte ein solches Dokument trotz dessen zu unübersichtlich sein, so kann auch für jede Sprache ein separates XML-Dokument vorgehalten werden.

⁶⁰ XML ist ein Standard zur Erstellung strukturierter, maschinen- und menschenlesbarer Dokumente und wurde bzw. wird, wie auch HTML und CSS, vom World Wide Web Consortium (W3C) entwickelt. XML ist eine textbasierte Metasprache, welche es ermöglicht, Daten bzw. Dokumente so zu beschreiben, dass diese zwischen einer Vielzahl von Anwendungen - insbesondere über das Internet - ausgetauscht und weiterverarbeitet werden können. Der Grundgedanke von XML ist dabei die konsequente Trennung von Inhalt, Struktur und Layout (vgl. Mertens 2001, S. 518).

```

<stringpool>
  <subgroup name="button">
    <text key="cmdDelete">
      <en>Delete</en>
      <de>Löschen</de>
    </text>
    <text key="cmdCancel">
      <en>Cancel</en>
      <de>Abbrechen</de>
    </text>
  </subgroup>
  <subgroup name="message">
    <text key="lblDeleteItem">
      <en>Are you sure you want to delete this item?</en>
      <de>Möchten Sie diesen Eintrag wirklich löschen?</de>
    </text>
  </subgroup>
</stringpool>

```

Abb. 5.7: Mögliche XML-Datenstruktur für die Lokalisierung

5.5.3 Lösungsansatz

ASP.NET 2.0 stellt einen eigenen Ansatz für die Lokalisierung von Websystemen zur Verfügung, bei welchem alle zu lokalisierenden Zeichenfolgen in XML-basierte Ressourcendateien mit der Dateierweiterung `.resx` ausgelagert werden. Dieses Format existiert jedoch nur zur Entwicklungszeit, da die ASP.NET-Laufzeitumgebung die Ressourcendateien in Just-in-Time-Manier in DLLs kompiliert. Dies hat den Vorteil, dass für eine neue Sprache lokalisierte Elemente einfach der Webanwendung hinzugefügt werden können, indem die neue XML-basierte Ressourcendatei auf den Webserver kopiert wird (vgl. Pattison 2006).

Grundsätzlich existieren zwei Arten von Ressourcendateien, die lokalen und globalen Ressourcen. Lokale Ressourcen können mit der Entwicklungsumgebung Visual Studio (VS) 2005 für beliebige ASP.NET-Webforms, Vorlageseiten und benutzerdefinierte Steuerelemente, die in der Entwurfsansicht geöffnet sind, automatisch generiert werden. In diesem Zusammenhang wird bspw. für jedes Webform eine eigene Ressourcendatei angelegt und in einem Unterverzeichnis des Webprojekts speziell für lokale Ressourcen (`\App_LocalResources`) abgelegt, damit diese von ASP.NET zur Laufzeit kompiliert werden können. VS 2005 füllt die Ressourcendateien auf, indem in jedem Webform nach allen lokalisierbaren Steuerelementinhalten und HTML-Elementen gesucht wird. Die Inhalte dieser Elemente werden schließlich in der Ressourcendatei gespeichert, wobei jeder Eintrag mit einem eindeutigen Schlüssel versehen wird. Diese automatisch erzeugten Ressourcendateien sind über ihre Benennungskonvention jeweils mit dem

dazugehörenden Webform verknüpft. Wird bspw. von VS 2005 eine lokale Ressource für ein Webform mit der Bezeichnung `default.aspx` generiert, trägt diese den Namen `default.aspx.resx` (vgl. Bustamante 2005). Die automatische Generierung von lokalen Ressourcen für eine jede Webform kann jedoch dazu führen, dass bestimmte Zeichenfolgen mehrfach entweder in einer Ressourcendatei oder verteilt über mehrere Ressourcendateien, vorgehalten werden. Um dies zu vermeiden, können ausgewählte oder auch alle Eigenschaften von Steuerelementinhalten und HTML-Elementen manuell in einer bzw. mehreren globalen Ressourcen gespeichert werden.

Globale Ressourcen sind nicht an bestimmte Webforms gebunden, sondern global für das gesamte Webprojekt gültig und werden in einem speziellen Verzeichnis (`\App_GlobalResources`) abgelegt. Mit globalen Ressourcen können also beliebige Zeichenfolgen wie bspw. Dialogtexte oder Fehlermeldungen vorgehalten werden, welche an unterschiedlichsten Stellen innerhalb der Webanwendung auftreten. Zur Laufzeit wird ebenfalls mit einem eindeutigen Schlüssel auf die Einträge der globalen Ressourcendateien zugegriffen (vgl. Bustamante 2005).

Zur Erstellung und Verwaltung von globalen und lokalen Ressourcen stellt VS 2005 einen Ressourcen-Editor zur Verfügung. Wie bereits erwähnt, können sowohl lokale als auch globale Ressourcendateien zur Umsetzung von mehrsprachigen Anwendungen verwendet werden. Die im Folgenden beschriebene Vorgehensweise ist dabei für lokale und globale Ressourcen dieselbe.

Grundsätzlich gilt, dass verschiedene Sprach- und Kulturkreise jeweils in einer eigenen Ressourcendatei abgelegt werden, wobei für die Unterscheidung der Dateien die Kulturkürzel laut Request for Comments (RFC) 1766 zum Einsatz kommen. Gemäß RFC 1766 wird ein eindeutiges Kulturkürzel aus einem Sprach- (zwei Kleinbuchstaben) und einem Länderkürzel (Kulturraum, zwei Großbuchstaben) zusammengesetzt (vgl. Alvestrand 1995). So entstehen Kulturkürzel, wie bspw. „de-DE“ (für Deutsch in Deutschland), „en-GB“ (für britisches Englisch) und „en-US“ (für amerikanisches Englisch). Die Kulturkürzel müssen bei den Ressourcendateien nach dem Dateinamen und vor der Endung `.resx` angegeben werden. Eine Ressourcendatei mit der Bezeichnung `texte.en-US.resx` gilt bspw. für den Kulturraum USA. Wird dagegen für eine Ressourcendatei nur ein Sprachkürzel angegeben (z. B. `texte.en.resx`), gilt diese für alle Kulturräume der jeweiligen Sprache. Ist hingegen kein Kürzel angegeben (z. B. `texte.resx`), handelt es sich um die Standardressourcendatei, welche verwendet wird, falls keine passende sprach- oder kulturspezifische Datei gefunden werden kann. Für die Erzeugung von anderen Sprachvarianten einer Ressourcendatei existiert in VS 2005 keine automatische Ableitungsfunktion. Um eine Ressourcendatei für eine andere Spra-

che zu erstellen, muss diese nicht unbedingt neu angelegt werden. So kann eine bereits vorhandene Ressourcendatei einfach kopiert und bspw. von `texte.de-DE.resx` in `texte.en-GB.resx` umbenannt werden, um in englisch lokalisierte Zeichenfolgen bereitzustellen. Das Übersetzen der Ressourceneinträge muss anschließend manuell erfolgen (vgl. Schwichtenberg 2006, S. 567 ff.). In Abb. 5.8 ist ein Ausschnitt aus einer Ressourcendatei für die deutsche Sprache zu sehen.⁶¹

```
<root>
  [...]
  <data name="LoginFailure" xml:space="preserve">
    <value>Anmeldung fehlgeschlagen.</value>
  </data>
  <data name="LoginSuccessful" xml:space="preserve">
    <value>Anmeldung erfolgreich.</value>
  </data>
  <data name="NoAccess" xml:space="preserve">
    <value>
      Sie haben nicht die notwendigen Rechte für die
      angeforderte Seite.
    </value>
  </data>
</root>
```

Abb. 5.8: Ausschnitt aus einer XML-basierten Ressourcendatei

Im Rahmen dieses Entwicklungsprojekts sollen für die Lokalisierung des Websystems ausschließlich globale Ressourcendateien zum Einsatz kommen, welche mit dem Ressourcen-Editor von VS 2005 erstellt und editiert werden. Wenn bspw. der Funktionsumfang von VS 2005 nicht zur Verfügung stehen würde bzw. Plattformunabhängigkeit eine Rolle spielen würde, dann können „normale“ XML-Dateien, wie sie in Abschnitt 5.5.2 vorgestellt wurden, einen interessanten Lösungsansatz zur Lokalisierung darstellen. Die Verwendung von XML-basierten Ressourcendateien liegt jedoch nahe, da z. B. der Zugriff auf diese bereits von ASP.NET bereitgestellt wird und so keine eigene Klasse implementiert werden muss, welche die notwendige Zugriffs- und Caching⁶²-Logik kapselt.

Die zum Einsatz kommenden Ressourcendateien werden in erster Linie Dialogtexte, Fehlermeldungen und Zeichenfolgen für die Beschriftung von Steuerelementen, wie bspw. Schaltflächen, Menüs und Verweisen enthalten. Zur Laufzeit wird dann über

⁶¹ Das `xml:space`-Attribut beschreibt hierbei, wie mit Whitespaces (Tabulator, Leerzeichen etc.) umgegangen wird. Der Wert „`preserve`“ besagt, dass vorhandene Whitespaces erhalten bleiben sollen.

⁶² Unter Caching (Zwischenspeichern) wird das Vorhalten von Daten – in diesem Fall von Zeichenfolgen – an einem anderen Ort als dem Ursprungsort verstanden, damit diese ohne erneutes Laden aus der ursprünglichen Quelle sofort zur Verfügung stehen, wenn sie wieder gebraucht werden (vgl. Schwichtenberg 2006, S. 968).

gleich bleibende Ausdrücke, auf welche im Rahmen der Realisierung genauer eingegangen werden soll, in Abhängigkeit von der geltenden Spracheinstellung auf die richtigen Ressourcendateien zugegriffen. Sollte hierbei keine Ressourcendatei für eine Sprache gefunden werden, so wird die Standardsprache angezeigt. In ASP.NET wird dieses Verhalten als Rückfallstrategie bezeichnet.

Wie bereits erwähnt, werden die im Websystem zur Verfügung stehenden Sprachen in der Datenbank, genauer gesagt in der Tabelle TBL_LANGUAGE, vorgehalten (s. Abb. 5.2). Zudem enthält auch die Datenbank in der Tabelle TBL_I18N Zeichenfolgen, die zur Lokalisierung des Websystems notwendig sind. Es handelt sich hierbei hauptsächlich um Übersetzungen für die Inhalte von Listensteuerelementen (z. B. Auswahl- und Listenfelder), welche einem Benutzer die Auswahl aus einer Menge von vorgegebenen Optionen ermöglichen. Da diese Auswahlmöglichkeiten der Steuerelemente nicht statisch durch Tags in den ASP.NET-Webforms hinterlegt, sondern in der Datenbank vorgehalten werden, liegt es nahe, die zugehörigen Übersetzungen ebenfalls dort und nicht in Ressourcendateien zu speichern. Die einzelnen Listensteuerelemente werden dann beim Aufrufen einer Webseite per Datenbindung mit den jeweiligen Auswahlmöglichkeiten befüllt. Bei den hierfür notwendigen Datenbankabfragen wird jeweils die Tabelle, welche die Auswahlmöglichkeiten für das Steuerelement enthält, über eine Verbundbedingung mit der Tabelle TBL_I18N verknüpft, wobei die ID der aktuell im Websystem geltenden Sprache (ID_LANGUAGE) als Selektionsbedingung verwendet wird.

5.6 Sicherheitskonzept

Die Sicherheit von Systemen und Daten ist die Grundlage moderner Informationsverarbeitung und entscheidet über deren erfolgreichen Einsatz (vgl. Mertens 2001, S. 411). Der Begriff Sicherheit umfasst dabei die Gesamtheit von Methoden, Verfahren, Technologien und gesetzlichen Regelungen zur Gewährleistung einer sicheren, vertraulichen⁶³ und autorisierten Systemnutzung (vgl. Dumke/Lothar/Wille/Zbrog 2003, S. 66). Dies betrifft auch Aspekte wie Zuverlässigkeit und Ausfallsicherheit und somit Maßnahmen zum Schutz vor Stromausfällen, Systemabstürzen oder Softwarefehlern, um den Verlust von Daten zu verhindern (vgl. Heuer/Saake 2000, S. 560). Es gibt eine Vielzahl von Sicherheitsmechanismen, welche in Kombination dazu geeignet sind, relevante Sicherheitsaspekte technologisch umzusetzen. An dieser Stelle seien z. B. Kryptographie, Firewalls⁶⁴, Virens Scanner und Archivierung genannt. Eine absolute Sicherheit kann es

⁶³ Vertraulichkeit bedeutet in diesem Zusammenhang, dass Informationen gegenüber Kenntnisnahme unbefugter Dritter zu schützen sind (vgl. Mertens 2001, S. 411).

⁶⁴ Eine Firewall ist ein System aus Software- und Hardwarekomponenten und dient dazu, einen sicheren Übergang zwischen einem Rechnernetz mit höherem Schutzbedarf (innen) und einem mit niedrigem

trotz ausgeklügelter Verfahren nicht geben. Vielmehr ist ein Kompromiss zwischen dem sicherheitstechnisch Notwendigen und dem wirtschaftlich Vertretbaren zu finden (vgl. Mertens 2001, S. 411 f.).

In den folgenden Abschnitten wird das Sicherheitskonzept von ISDOS vorgestellt, wobei in diesem Zusammenhang die Aspekte Authentifizierung und Autorisierung sowie Netzwerk- und Datensicherheit betrachtet werden.

5.6.1 Authentifizierung

Bei der Authentifizierung geht es darum zu überprüfen, ob eine Person, die mit dem Informationssystem arbeitet, tatsächlich diejenige ist, die sie vorgibt zu sein (vgl. Mertens 2001, S. 411). Der Nachweis erfolgt in der Regel durch eine Kombination aus Benutzernamen und Kennwort, wobei im Fall des zu entwickelnden Websystems die Anmeldung mit dem Namen der Benutzerrolle, welche ein Benutzer einnehmen möchte, und dem dazugehörigen Kennwort stattfindet. Wie bereits erwähnt, werden Daten zu einzelnen Systembenutzern nicht in der Datenbank vorgehalten (s. Abschnitt 4.3.1).

ASP.NET 2.0 stellt verschiedene Authentifizierungsverfahren zur Verfügung. Ein Verfahren ist bspw. die so genannte Windows-Authentifizierung, bei welcher sich ASP.NET ganz auf die im IIS erfolgte Authentifizierung verlässt. Wenn z. B. im IIS die Authentifizierung auf Basis der Windows-Benutzerkonten stattfindet, wird entweder die aktuelle Windows-Benutzeranmeldung vom Browser an den Webserver weitergereicht oder der Benutzer durch einen Standardanmeldedialog zur Authentifizierung aufgefordert. Der Vorteil dieses Verfahrens besteht darin, dass keine eigene Benutzerverwaltung umgesetzt werden muss und daher auch kein zusätzlicher Administrationsaufwand entsteht. Im Rahmen dieses Entwicklungsprojekts kommt die Authentifizierung mit Hilfe des Windows-Anmeldeservers jedoch nicht zum Einsatz, da ein eigenes Rollenkonzept implementiert wird. Zudem soll nicht jeder Benutzer, der über ein gültiges Windows-Benutzerkonto verfügt, auch auf das Websystem zugreifen können. Ein weiteres Verfahren, das von ASP.NET zur Verfügung gestellt wird, ist die Passport-Authentifizierung. Hierbei wird für die Authentifizierung das Microsoft Passport-Netzwerk, ein so genannter internetbasierter Single-Sign-On-Dienst verwendet. Bei diesem Authentifizierungsverfahren muss die Kombination aus Benutzernamen und Kennwort bei Passport hinterlegt und kann dann für unterschiedliche Webanwendungen zur Anmeldung genutzt werden. Auch dieser Mechanismus kommt im zukünftigen

Schutzbedarf (außen) zu ermöglichen. Hardwarekomponenten können bspw. Rechner mit Netzwerkschnittstellen (z. B. Router) sein, Softwarekomponenten z. B. Paketfilter oder Proxies (vgl. Mertens 2001, S. 200 f.).

Websystem nicht zum Einsatz, da die Benutzerrollen mit den dazugehörigen Kennwörtern in der Datenbank verwaltet werden. ASP.NET bietet noch eine weitere Alternative und zwar die formularbasierte Authentifizierung (vgl. Schwichtenberg 2006, S. 596 ff.), welche im Rahmen dieses Entwicklungsvorhabens verwendet wird. Es sei erwähnt, dass natürlich auch die Möglichkeit besteht, einen vollständig eigenen Authentifizierungsmechanismus zu implementieren. Da ASP.NET jedoch bereits Klassen, Sicherheitssteuerelemente sowie eine einfache Möglichkeit zur Konfiguration bereitstellt, wurde die Entscheidung zugunsten der formularbasierten Authentifizierung getroffen, wobei dieser Mechanismus im Rahmen der Realisierung durch eigene Implementierungen angepasst und erweitert wird.

Bei dem zu entwickelnden Websystem soll die Authentifizierung mit Hilfe eines bestimmten Webforms, welches einen Anmeldedialog bereitstellt, stattfinden. Zur Verifizierung der im Anmeldedialog eingegebenen Werte wird die Datenbanktabelle TBL_ROLE (s. Abb. 5.2) verwendet, in welcher die im Websystem zur Verfügung stehenden Benutzerrollen mit den dazugehörigen Kennwörtern gespeichert sind. Um den Missbrauch der Kennwörter zu minimieren, werden diese mittels Secure Hash Algorithm (SHA), genauer gesagt dem SHA1⁶⁵, verschlüsselt und auch so innerhalb der Datenbank abgelegt. Prinzipiell sollen alle nicht-authentifizierten (anonymen) Benutzer, die versuchen ein zugriffsgeschütztes Webform aufzurufen, auf die Anmeldeseite umgeleitet werden. Nach erfolgreicher Authentifizierung sollen sie automatisch wieder zurück auf das ursprünglich angeforderte Webform gelangen. Die Gültigkeit der Anmeldung wird dabei auf die Dauer einer Benutzersitzung begrenzt.

5.6.2 Autorisierung

Mit Autorisierung wird der Prozess bezeichnet, in welchem entschieden wird, ob einem Benutzer Zugriff auf ein von ihm angefragtes Objekt⁶⁶ gewährt wird oder nicht (vgl. Merz 2002, S. 154). Voraussetzung für die Zugriffskontrolle ist die Authentifizierung, denn nur so ist die Benutzerrolle bekannt, welche ein Benutzer innehat. So erhält ein anonymes Benutzer nur dann den verlangten Zugriff, wenn das angefragte Objekt keine

⁶⁵ SHA1 ist eine Weiterentwicklung von SHA und erzeugt für eine beliebig große Datenmenge (ausgedrückt durch eine Zeichenfolge von Bytes) einen eindeutigen Hashwert mit einer Länge von 160 Bit. Sind zwei Hashwerte identisch, so sind die entsprechenden Originaldaten ebenfalls identisch. Jede noch so minimale Änderung an den Daten führt zu einer beträchtlichen Änderung des Hashwerts. Darüber hinaus kann vom Hashwert nicht wieder auf die Originaldaten geschlossen werden (vgl. Merz 2002, S. 162 f.).

⁶⁶ Solche Objekte können bspw. Webforms, per Uniform Resource Locator (URL) adressierte Dokumente auf dem Webserver, Datenbanken, Verzeichnisse und Dateien oder Prozeduren bzw. Funktionen sein.

Anmeldung erfordert. Ist ein Benutzer erfolgreich angemeldet, darf er immer dann eine Aktion auf ein bestimmtes Objekt anwenden, wenn diese Aktion in seiner Benutzerrolle hinsichtlich dieses Objektes enthalten ist. Um dies zu gewährleisten, werden einem Benutzer bspw. nur erlaubte Aktionen im Menü sowie zulässige Schaltflächen auf bestimmten Webforms angeboten. Grundsätzlich ist für jedes angefragte Objekt bei jedem Zugriff die Berechtigung erneut zu prüfen, denn nur so ist sicherzustellen, dass ein Benutzer auch tatsächlich die erforderlichen Rechte hat, welche er für ein Objekt benötigt.

Die Zugriffskontrolle wird mit Hilfe der Rechtevergabe geregelt, wobei die im Web-system zur Verfügung stehenden Rechte in der Datenbank verwaltet werden (s. Abschnitt 4.3.1). Diese Rechte setzen sich aus je einem Objekt und einer Aktion zusammen, welche für das Objekt ausgeführt werden kann. Die Objekte sind in der Tabelle TBL_OBJECT (s. Abb. 5.2) gespeichert, wobei anfänglich die Objekte „Bestellung“, „Kunde“, „Verteilerliste“ und „Benutzerrolle“ zur Verfügung stehen. Die möglichen Aktionen „anzeigen“, „anlegen“, „ändern“, „stornieren“ und „löschen“ werden in der Tabelle TBL_ACTION vorgehalten. Die Tabelle TBL_RIGHT enthält schließlich alle Rechte, welche durch die Kombination der Objekte mit den Aktionen entstehen (z. B. „Bestellung anzeigen“, „Verteilerliste ändern“). Die Aktion „stornieren“ wird hierbei ausschließlich mit dem Objekt „Bestellung“ in Beziehung gesetzt. Somit werden insgesamt 17 Rechte in der Datenbank vorgehalten, die im administrativen Bereich von ISDOS den einzelnen Benutzerrollen zugewiesen werden können. Diese Zuordnungen sind in der Tabelle TBL_ROLE_HAS_RIGHT gespeichert.

Darüber hinaus benötigt die Webanwendung grundlegende Rechte auf dem Webserver, um dort agieren und bspw. die zur Anwendung gehörenden Dateien aus dem Dateisystem lesen zu können. Es muss zudem gewährleistet sein, dass die Webanwendung auf die Datenbank zugreifen kann, wobei diese so zu schützen ist, dass nur berechtigte Anwendungssysteme Zugriff auf die gespeicherten Daten haben. Für das zu entwickelnde Websystem soll hinsichtlich des Zugriffs auf die Datenbank das Trusted-Subsystem-Modell⁶⁷ von ASP.NET zugrunde gelegt werden. Dies bedeutet, dass die Anzahl der authentifizierten Benutzer zur Datenbank hin abnimmt und nur noch eine im Websystem hinterlegte Kombination aus Benutzername und Kennwort für die Anmeldung an der Datenbank verwendet wird (s. Abb. 5.9). Die Datenhaltungsschicht vertraut somit darauf, dass die Applikationsschicht alle notwendigen Autorisierungsüberprüfungen vorgenommen hat.

⁶⁷ Für weiterführende Informationen zu den Sicherheitsarchitekturen von ASP.NET sei an dieser Stelle auf Schwichtenberg 2006, S. 616 verwiesen.



Quelle: In Anlehnung an Schwichtenberg 2006, S. 616.

Abb. 5.9: Trichterprinzip beim Trusted-Subsystem-Modell

5.6.3 Netzwerk- und Datensicherheit

Eine Grundregel zur Sicherung von Webanwendungen besteht darin, dass Benutzer nur soviel Zugriff auf den Webserver haben, wie unbedingt zum Ausführen der Anwendung benötigt wird. Es ist somit immer ratsam eine Firewall vor dem Webserver zu platzieren, um nicht benötigte Ports für einen Zugriff zu sperren (vgl. Wenz/Hauser/Kordwig/Trennhaus 2002, S. 727). Die Sicherheitsinfrastruktur wird jedoch im Rahmen dieser Diplomarbeit nicht ausführlicher betrachtet, da diese im Verantwortungsbereich der Abteilung Zentralbereich Informationsverarbeitung (s. Abb. 3.1) liegt. Gleiches gilt für die Datensicherung, welche mittels täglicher Backups gewährleistet wird. Im Falle eines Systemausfalls kann somit das entsprechende Backup zurückgespielt werden. Letztendlich sei an dieser Stelle erwähnt, dass zur Minimierung des Sicherheitsrisikos darauf zu achten ist, dass Webserver sowie Betriebssystem mittels Sicherheitsupdates immer auf dem aktuellsten Stand gehalten werden.

6 Realisierung

Nach der Konzeption und dem Entwurf des Websystems behandelt dieses Kapitel die eigentliche Umsetzung der Arbeit. Die Beschreibung der Realisierung orientiert sich dabei an den in Abschnitt 5.1 vorgestellten Schichten. Es sei darauf hingewiesen, dass in diesem Kapitel natürlich nicht der vollständige Programmcode besprochen werden kann, da dies den Umfang der Arbeit sprengen würde. Vielmehr soll ein Eindruck von der prinzipiellen Funktionalität des Websystems vermittelt werden, wobei u. a. anhand von Programmcodebeispielen und Screenshots der Benutzeroberfläche auf Besonderheiten und zentrale Bestandteile der Anwendung eingegangen wird.

6.1 Datenhaltungsschicht

Im Rahmen der Implementierung wurde der in Abschnitt 5.2 vorgestellte relationale Datenbankentwurf unter Verwendung der Structured Query Language (SQL) in einer Oracle 9i Datenbank sowie zu Testzwecken in einer Oracle 10g Datenbank umgesetzt.

SQL ist eine vom American National Standards Institute (ANSI) und der ISO genormte relationale Datenbanksprache, welche von allen namhaften relationalen Datenbanken unterstützt wird. Sie unterteilt sich u. a. in die Data Definition Language (DDL) und die Data Manipulation Language (DML). Die DDL dient dabei in erster Linie der Definition von Tabellen, Sichten und Wertebereichen, wohingegen die DML die Manipulation von Daten ermöglicht (vgl. Heuer/Saake 2000, S. 179).

Des Weiteren kam bei der Realisierung und Administration der Datenbank der Oracle SQL Developer zum Einsatz, ein Werkzeug mit grafischer Benutzeroberfläche, welches zurzeit kostenlos von Oracle zur Verfügung gestellt wird. Es ermöglicht dem Benutzer u. a. das Bearbeiten von Datenbankobjekten, wie etwa Tabellen, Sichten, Indizes, Triggern und Sequenzen sowie das Erstellen und Testen von SQL-Statements und -Skripten (vgl. Oracle Corporation 2006).

Abbildung 6.1 zeigt die Benutzeroberfläche des Oracle SQL Developer am Beispiel einer zur Bearbeitung geöffneten Sichtdefinition. Wie der Abbildung zu entnehmen ist, wurden bei der Datenbankrealisierung eine Vielzahl von Sichten (engl. views) angelegt. Dabei wird eine Sicht über eine Datenbankabfrage definiert, welche jederzeit das Ergebnis dieser Abfrage reflektiert. Im Rahmen dieses Entwicklungsprojekts wurden bspw. oft benötigte Teilabfragen und aufwändige SQL-Abfragen mit mehreren Verbundbedingungen als Sichten realisiert.

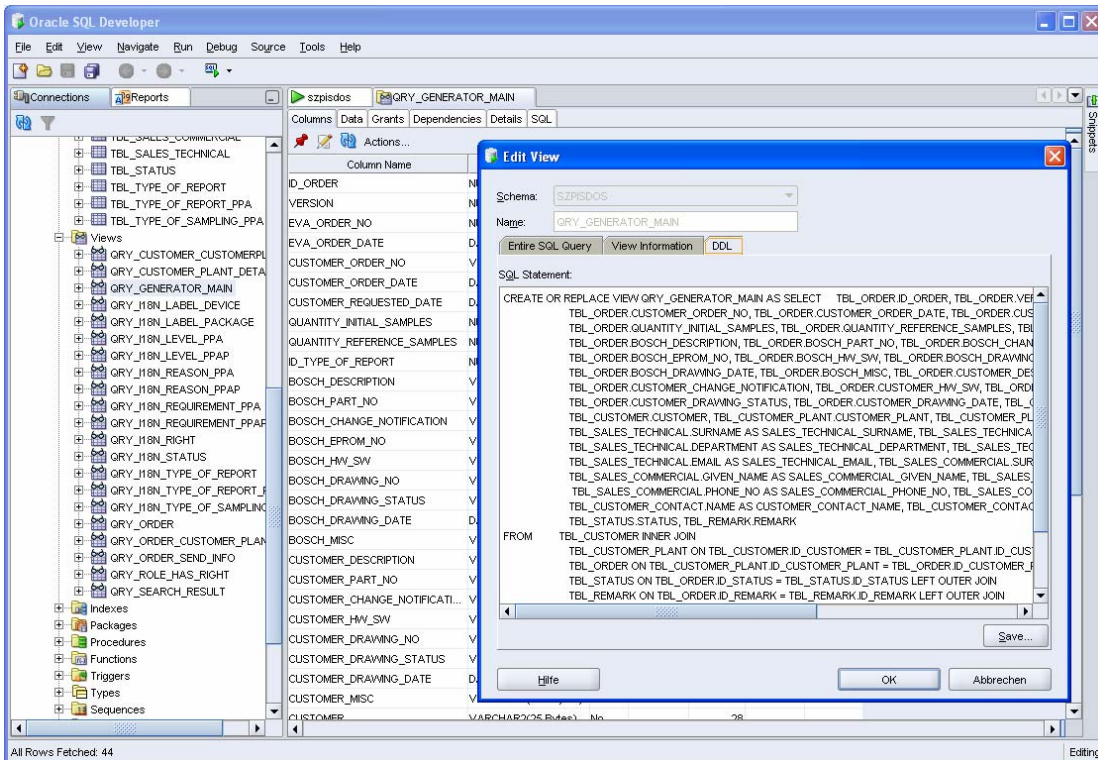


Abb. 6.1: Benutzeroberfläche des Oracle SQL Developer

Diese Vorgehensweise führt dazu, dass ein Benutzer z. B. ohne Kenntnis des Datenbankschemas auf bestimmte Informationen der Datenbank zugreifen kann, wobei ihm die Komplexität der eigentlichen Datenbankabfrage verborgen bleibt. Eine weitere Aufgabe des Einsatzes von Sichten besteht in der Strukturierung und Präsentation der in der Datenbank vorhandenen Informationsmenge, da verschiedene Benutzer bzw. Anwendungen oftmals unterschiedliche Anforderungen an die Aufbereitung der Daten haben. So ermöglicht eine Sicht z. B. dem Generator der QMM1 den Zugriff auf einen bestimmten Bereich der Datenbank, wobei nur die für dieses Anwendungssystem relevanten Informationen der EM-Bestellung durch Ausblenden der unwichtigen Daten präsentiert werden.

Darüber hinaus werden mit Hilfe von einigen Sichten ausgewählte Daten schon bei der Datenbankabfrage für das spätere Anzeigen auf der Benutzeroberfläche von ISDOS vorbereitet. Ein Beispiel hierfür ist die RB-Sachnummer, welche beim Anzeigen als 12-stellige Zeichenfolge im Format: 4 Zeichen | Punkt | 3 Zeichen | Punkt | 3 Zeichen (z. B. 0281.200.300) benötigt wird, in der Datenbank jedoch als 10-stellige Zeichenfolge ohne Punkte gespeichert ist. Abbildung 6.2 zeigt die Formatierung der RB-Sachnummer mit Hilfe einer SQL-Abfrage, die so auch als Teil einer Sicht definiert wurde. Hervorzuheben sind hierbei die Zeichenkettenfunktion `SUBSTR()`, welche dazu dient nur einen Teil einer gespeicherten Zeichenkette zurück zu geben und der Verknüpfungsoperator „||“ zum Verketteten mehrerer Zeichenfolgen. So gibt `SUBSTR(BOSCH_`

PART_NO, 1, 4) eine Zeichenkette von vier Zeichen, beginnend beim ersten Zeichen, zurück, wobei diese Zeichenkette mit dem Ergebnis von SUBSTR(BOSCH_PART_NO, 5, 3) und SUBSTR(BOSCH_PART_NO, 8, 3) verknüpft wird. Diese Formatierung findet bereits auf Datenbankebene statt, damit die Ergebnismenge der Abfrage auf die jeweilige Sicht nicht auf Applikationsebene in einer Schleife zeilenweise durchlaufen und in die benötigte Form gebracht werden muss, sondern z. B. gleich an ein zugehöriges ASP.NET-Serversteuerelement angebunden bzw. in einer Schleife ausgegeben werden kann. Diese Vorgehensweise kann einen Performance-Vorteil ergeben und ist dann besonders sinnvoll, wenn es eine große Anzahl von Datensätzen vorzubereiten gilt.

```

SELECT   SUBSTR(BOSCH_PART_NO,1,4) || '.' ||
           SUBSTR(BOSCH_PART_NO,5,3) || '.' ||
           SUBSTR(BOSCH_PART_NO,8,3) AS BOSCH_PART_NO
FROM     TBL_ORDER;

```

Abb. 6.2: SQL-Abfrage zur Ermittlung der formatierten RB-Sachnummern

6.2 Applikationsschicht

Für die Implementierung der Webanwendung kam, wie in den Anforderungen festgelegt, die objektorientierte Programmiersprache VB 2005 von Microsoft zum Einsatz. Die Programmierlogik der ASP.NET-Webforms hätte auch mit anderen .NET-Programmiersprachen, wie bspw. CSharp (C#) oder JSharp (J#) umgesetzt werden können. Die Wahl fiel jedoch auf VB 2005, da in der QMM1 bereits andere Anwendungen auf der Basis von VB 6.0 und Visual Basic for Applications (VBA) existieren und die für die spätere Wartung und Erweiterung von ISDOS zuständigen Mitarbeiter dadurch einen geringeren Aufwand bei ihrer Arbeit zu erwarten haben, wenn eine syntaktisch ähnliche Programmiersprache zu Grunde liegt. Die bestehenden Unterschiede zwischen z. B. C# und VB 2005 sind indes nur primär syntaktischer Natur und für die meisten Problemszenarien unerheblich. Um ASP.NET 2.0 als serverseitige Technologie für Webanwendungen nutzen zu können, wird darüber hinaus ein Webserver benötigt, der ASP.NET-Webforms ausführen kann. Hierfür kommt wie gefordert ein Microsoft Webserver, genauer gesagt ein IIS (auf MS Windows Server 2003) zum Einsatz (s. Abschnitt 4.2.2).

Im Folgenden wird auf die Verzeichnisstruktur sowie die Konfiguration der Webanwendung eingegangen. Anschließend sollen ausgewählte Bereiche der implementierten Applikationslogik, welche für ISDOS von genereller Bedeutung sind, näher vorgestellt werden.

6.2.1 Verzeichnisstruktur

Bei jedem Webprojekt ist es wichtig, eine zweckmäßige Dateiablage zu haben, damit ein effizientes Arbeiten möglich ist. In ASP.NET 2.0 besitzen Webanwendungen daher eine bestimmte Verzeichnisstruktur zur Ablage von Dateien, welche im Rahmen dieser Implementierung um projektspezifische Verzeichnisse und Inhalte erweitert wurde. In Abb. 6.3 ist die Verzeichnisstruktur von ISDOS dargestellt.

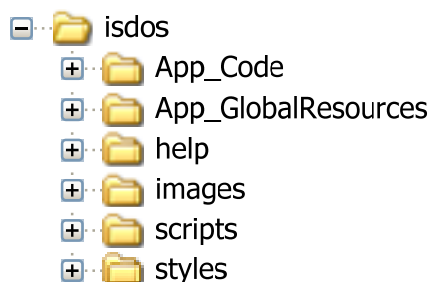


Abb. 6.3: Verzeichnisstruktur von ISDOS

Im Hauptverzeichnis, welches den Namen des Informationssystems trägt, befinden sich alle ASP.NET-Webforms inklusive der Hintergrundcodedateien. Dazu zählen bspw. die Webforms für das Erfassen, Öffnen, Suchen und Versenden von EM-Bestellungen sowie die Startseite von ISDOS, welche automatisch aufgerufen wird, wenn ein Systembenutzer dieses Verzeichnis ohne Angabe einer bestimmten Webseite über den Webbrowser anfordert. Zudem enthält das Hauptverzeichnis die Masterpage (s. Abschnitt 5.4.2) und die Webkonfigurationsdatei, auf welche im nächsten Abschnitt ausführlicher eingegangen wird.

Die Verzeichnisse `App_Code` und `App_GlobalResources` sind von der Entwicklungsumgebung VS 2005 automatisch generiert worden. Hintergrund ist, dass in ASP.NET-Webanwendungen bestimmte Datei- und Verzeichnisnamen für bestimmte Typen von Inhalten verwendet werden. So enthält das Verzeichnis mit dem Namen `App_Code` immer die Webform-unabhängigen Codedateien, welche zur Laufzeit automatisch kompiliert werden. Im Rahmen dieses Entwicklungsprojekts gehören dazu alle eigenständigen VB-Codedateien, wie bspw. die Klasse `clsDBManager` für die Datenbankkommunikation, die Klasse `clsI18N` für die Lokalisierung und die Klasse `clsAuthFilter` für die Authentifizierung und Autorisierung eines Systembenutzers. Im Verzeichnis `App_GlobalResources` werden dagegen alle sprachspezifischen globalen Ressourcendateien abgelegt, welche für die Lokalisierung des Websystems benötigt werden.

Das Verzeichnis `help` dient als Sammelbecken für alle Dateien und Unterverzeichnisse, die im Zusammenhang mit der Online-Hilfe des Informationssystems stehen. Bisher

befindet sich in diesem Verzeichnis ausschließlich ein Webform, das den Systembenutzer darüber informiert, dass die Funktionalität noch nicht umgesetzt worden ist. Im Verzeichnis `images` sind alle Grafiken wie etwa die Pfeilsymbole für die Verweise oder das Firmenlogo für den Kopfbereich gespeichert, welche in der Benutzeroberfläche von ISDOS eingebunden sind. Das Verzeichnis `scripts` enthält eine JavaScript-Datei, auf deren Funktionalität bei der Vorstellung der Präsentationsschicht näher eingegangen wird. Im letzten Verzeichnis `styles` sind zwei CSS-Dateien abgelegt, in denen die Gestaltungsmerkmale der webbasierten GUIs vorgehalten werden (s. Abschnitt 5.4.2).

6.2.2 Konfiguration

Wie bereits im letzten Abschnitt erwähnt, befindet sich im Hauptverzeichnis von ISDOS eine Webkonfigurationsdatei. Diese XML-basierte Datei hat den Namen `web.config` und besteht aus zahlreichen Konfigurationselementen, welche für die gesamte Webanwendung, also das Hauptverzeichnis und alle untergeordneten Verzeichnisse, Gültigkeit besitzen. Im Folgenden wird anhand des in Abb. 6.4 gezeigten Ausschnitts der Aufbau dieser Konfigurationsdatei vorgestellt. Eine genauere Betrachtung von einzelnen Konfigurationseinstellungen findet in den kommenden Abschnitten jeweils im Zusammenhang mit der Beschreibung ihrer Anwendung statt.

Die XML-Konfigurationsdatei von ISDOS ist in die drei Hauptelemente `appSettings`, `system.web` und `system.net` unterteilt. Innerhalb des `appSettings`-Konfigurationselements können beliebige Einstellungen in Form von Attribut/Wert-Paaren hinterlegt werden, die zur Laufzeit aus der Anwendung heraus abgefragt werden können. Im Rahmen dieses Entwicklungsprojekts wurde hier u. a. die Verbindungszeichenfolge zur Datenbank abgelegt. Der `system.web`-Abschnitt ist für die Webanwendung von besonderer Bedeutung. So werden z. B. mit Hilfe des `compilation`-Elements die Festlegungen für die automatische Kompilierung der ASP.NET-Anwendung definiert. Mit dem Attribut `debug="false"` wird dabei der Debugmodus deaktiviert und mit `explicit="true"` wird die Deklaration aller Variablen im Programmcode erzwungen. Das Attribut `strict="true"` bewirkt, dass im Programmcode keine Datentypkonvertierungen zugelassen sind, bei denen Datenverluste auftreten könnten. Innerhalb des folgenden `namespaces`-Konfigurationselements wurde der Namensraum⁶⁸ `System.Data.OracleClient` importiert, was dazu führt, dass im Programmcode der Webanwendung ein vereinfachter Zugriff auf die Klassen

⁶⁸ Zur Gewährleistung der Übersichtlichkeit sind die Klassen der FCL, welche von allen .NET-Programmiersprachen aus genutzt werden können, in 153 Namensräume (engl. namespace) eingeteilt (vgl. Schwichtenberg 2006, S. 100 f.).

dieses Namensraums möglich ist, da z. B. beim Aufruf einer Methode nicht mehr der gesamte Namensraum der Klasse vorangestellt werden muss. Weitere Namensräume wurden nicht importiert, da dies bereits in einer globalen .NET-Konfigurationsdatei geschehen ist. Zuletzt wurden innerhalb des `system.web`-Abschnitts die Einstellungen für das Sessionmanagement (s. Abschnitt 6.4) und den Authentifizierungsmechanismus (s. Abschnitt 6.6) definiert. Im dritten und letzten Hauptelement mit der Bezeichnung `system.net` wurden die Einstellungen für den Mailserver zum Versenden von E-Mails festgelegt.

Ein Vorteil der XML-basierten Webkonfigurationsdatei besteht darin, dass alle Einstellungen zentral festgelegt und auch verändert werden können. Dabei sind die Änderungen, welche zur Laufzeit durchgeführt werden, in der Regel ohne Neustart der ASP.NET-Anwendung wirksam. Die Verwendung einer solchen Konfigurationsdatei sowie das konsequente Vermeiden von festen Hinterlegungen und absoluten Pfaden im Programmcode (z. B. zu URL-Ressourcen und Dateien) führen dazu, dass ISDOS ohne hohen Änderungsaufwand von einer ASP.NET-fähigen Computerplattform auf eine andere übertragen werden und somit auch an anderen RB-internen Fertigungsstandorten zum Einsatz kommen kann.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="strConn" value="Data Source=orasa02.world;
      Persist Security Info=True;User ID=xx;Password=xx;Unicode=True"/>
  </appSettings>
  <system.web>
    <compilation debug="false" explicit="true" strict="true"/>
    <pages>
      <namespaces>
        <add namespace="System.Data.OracleClient"/>
      </namespaces>
    </pages>
    <sessionState mode="InProc" cookieless="AutoDetect" timeout="30"/>
    <authentication mode="Forms">
      <forms name="ISDOS" protection="All" cookieless="AutoDetect"
        timeout="30" path="/"></forms>
    </authentication>
  </system.web>
  <system.net>
    <mailSettings>
      <smtp deliveryMethod="Network">
        <network defaultCredentials="true" host="xx"/>
      </smtp>
    </mailSettings>
  </system.net>
</configuration>
```

Abb. 6.4: Aufbau der XML-Konfigurationsdatei von ISDOS

6.2.3 Datenbankkommunikation

Für die Kommunikation mit der Datenbank wird die zentrale Datenzugriffsschnittstelle ActiveX Data Objects .NET (ADO.NET) verwendet. ADO.NET ist Teil der FCL (Namensraum `System.Data`) und wird mit diversen Datenprovidern ausgeliefert. Im Rahmen dieser Implementierung kam der .NET Data Provider für Oracle (Namensraum `System.Data.OracleClient`) zum Einsatz. Dieser funktioniert mit Hilfe von Oracle-Client-Schnittstellen, welche durch die Installation der Oracle-Client-Software zur Verfügung gestellt werden.

Bevor die Webanwendung auf die Datenbank zugreifen kann, muss eine Datenbankverbindung mit einem `Connection`-Objekt hergestellt werden. Der anschließende Zugriff auf die Datenquelle geschieht immer mit einem `Command`-Objekt. Für die Bereitstellung dieser Objekte ist die Implementierung der bereits vorgestellten Klasse `clsDBManager` zuständig. Der Aufbau der Datenbankverbindung findet bei der Instanziierung der Klasse statt. Hierfür muss dem Konstruktor dieser Klasse die Datenbankverbindungszeichenfolge übergeben werden (s. Abb. 6.5).

```
Public Sub New(ByVal strConnString As String)
    objConn = New OracleConnection(strConnString)
End Sub
```

Abb. 6.5: Konstruktor der Klasse `clsDBManager`

Wie bereits im letzten Abschnitt erwähnt, befindet sich diese Verbindungszeichenfolge in der XML-Konfigurationsdatei und ist nicht fest im Programmcode implementiert. Um die Zeichenfolge aus der Webanwendung heraus auslesen zu können, wird die Objektmenge `AppSettings` der Klasse `System.Configuration.ConfigurationManager` genutzt (s. Abb. 6.6).⁶⁹

```
strConnString = ConfigurationManager.AppSettings("strConn")
```

Abb. 6.6: Auslesen der Datenbankverbindungszeichenfolge

Die Klasse `clsDBManager` stellt zudem die Methode `GetOracleCommand()` bereit (s. Abb. 6.7). Diese übernimmt sowohl die Instanziierung der Klasse `OracleCommand`, die Bindung des `OracleCommand`-Objekts (`objCmd`) an das `OracleConnection`-Objekt (`objConn`) über die Eigenschaft `Connection`, als auch das Öffnen der Datenbankverbindung mit `Open()`. Sollen mit dem providerspezifischen `Command`-Objekt

⁶⁹ Da der Namensraum `System.Configuration` durch die globale .NET-Konfigurationsdatei importiert wird, reicht an dieser Stelle die verkürzte Schreibweise. Findet sich in den folgenden Beispielen eine solche Kurzform der Schreibweise, ist dies auf denselben Sachverhalt zurückzuführen.

anschließend Änderungen an mehreren Datenbanktabellen auf einmal durchgeführt werden, ist es notwendig, diese Änderungen als eine logische Einheit zu betrachten, welche entweder vollständig oder gar nicht ausgeführt wird. Hierfür bieten DBMS Transaktionen an, in denen bspw. mehrere SQL-Befehle verpackt werden können. Der Beginn einer Transaktion kann dabei in der `GetOracleCommand()`-Methode mit `BeginTransaction()` markiert werden. Das `Transaction`-Objekt (`objTrans`) wird anschließend dem `OracleCommand`-Objekt zugewiesen.

```
Public Function GetOracleCommand(ByVal bTrans As Boolean) _
    As OracleCommand
    objCmd = New OracleCommand()
    objCmd.Connection = objConn
    objConn.Open()
    If bTrans = True Then
        objTrans = objConn.BeginTransaction
        objCmd.Transaction = objTrans
    End If
    Return objCmd
End Function
```

Abb. 6.7: `GetOracleCommand()`-Methode der Klasse `clsDBManager`

Auf dem von der Methode `GetOracleCommand()` zurückgegebenen providerspezifischen `Command`-Objekt können nun jegliche Formen von SQL-Befehlen zum Abfragen und Manipulieren der Datenbankinhalte ausgeführt werden. Dabei kann für das Objekt über die Eigenschaft `CommandText` z. B. eine SQL-Abfrage festgelegt werden, welche Daten liefert und mit dem Schlüsselwort `SELECT` beginnt. Die Abfrage kann dann bspw. mit der Methode `ExecuteReader()` der `OracleCommand`-Klasse zur Ausführung gebracht werden. Als Ergebnis wird ein `DataReader`-Objekt zurückgegeben, das alle Datensätze enthält, die für diese Abfrage in der Datenbank ermittelt wurden. Die Ergebnismenge des `DataReader`-Objekts kann schließlich entweder an ein ASP.NET-Serversteuerelement gebunden oder in einer Schleife durchlaufen werden. Abbildung 6.8 zeigt einen Ausschnitt aus einer Methode, mit welcher die Namen der in der Datenbank gespeicherten Benutzerrollen ausgelesen und an ein Listenfeld mit der Bezeichnung `lstUserRoleName` gebunden werden. Da an dieser Stelle die grundlegende Vorgehensweise verdeutlicht werden soll, wurde der eigentliche Programmcode aus Platzgründen verkürzt dargestellt, indem der komplette Fehlerbehandlungsteil sowie andere für dieses Beispiel unrelevante Anweisungen ausgeblendet wurden.

Die `OracleCommand`-Klasse stellt darüber hinaus noch weitere Methoden bereit. So kommt bspw. die Methode `ExecuteNonQuery()` zum Einsatz, wenn eine SQL-Anweisung zur Manipulation von Datenbankinhalten (mit den Schlüsselwörtern `INSERT`, `UPDATE` oder `DELETE`) ausgeführt werden soll.

```

Try
    objCmd = objDBManager.GetOracleCommand(False)
    objCmd.CommandType = CommandType.Text
    objCmd.CommandText = "SELECT ROLE FROM TBL_ROLE ORDER BY ROLE"

    objDataReader = objCmd.ExecuteReader()
    lstUserRoleName.DataSource = objDataReader
    lstUserRoleName.DataBind()
    objDataReader.Close()
Catch ex As Exception
    ' ...
Finally
    objDBManager.Dispose()
End Try

```

Abb. 6.8: Ausführung eines SQL-Befehls mit anschließender Datenbindung

Als Ergebnis wird hierbei keine Datenmenge, sondern nur ein Zahlenwert zurückgegeben, der angibt, wie viele Datensätze von der Operation betroffen waren. Wie bereits erwähnt, können mehrere solcher DML-Befehle auch in einer Transaktion verpackt werden. Hierbei müssen die SQL-Anweisungen einzeln abgearbeitet und u. a. mit Hilfe eines Try-Catch-Blocks auf ihre erfolgreiche Durchführung hin überprüft werden. Im Fehlerfall wird die Transaktion mittels `objCmd.Transaction.Rollback()` zurückgesetzt. Bei erfolgreicher Ausführung wird die Transaktion mit `objCmd.Transaction.Commit()` bestätigt und die Auswirkungen der Operationen werden endgültig in der Datenbank gespeichert.

Genauso, wie eine Datenbankverbindung erst kurz vor ihrem Gebrauch geöffnet werden sollte, ist es wichtig, diese zu schließen, sobald sie nicht mehr benötigt wird, anstatt auf den Garbage Collector⁷⁰ zu warten. Für das Schließen der Datenbankverbindung ist die Methode `Dispose()` in der Klasse `clsDBManager`, genauer gesagt die Anweisung `objConn.Close()`, zuständig. Im Programmcodebeispiel der Abb. 6.8 wird die `Dispose()`-Methode mit Hilfe des Objekts `objDBManager` aufgerufen. Dabei wird jedoch nicht die tatsächliche Datenbankverbindung geschlossen, sondern nur das `OracleConnection`-Objekt im Speicher freigegeben. Denn ADO.NET stellt einen so genannten Verbindungspoolingmechanismus⁷¹ zur Verfügung, der eine Wiederverwendung von Datenbankverbindungen ermöglicht. So wird mit eben jenem Mechanismus eine bestimmte Anzahl von aktiven Datenbankverbindungen in einem Pool vorgehalten. Versucht eine Anwendung eine Verbindung zur Datenbank aufzubauen, wird eine bereits geöffnete Verbindung aus diesem Pool genutzt. Voraussetzung hierfür ist, dass

⁷⁰ Ein Garbage Collector ist ein Mechanismus zur Speicherbereinigung, der automatisch alle Objekte löscht, auf welche keine Referenzen mehr zeigen.

⁷¹ Dieser Mechanismus ist voreingestellt und kann in der Datenbankverbindungszeichenfolge mit Hilfe von Parametern konfiguriert werden.

eine solche auch vorhanden ist. Wird eine Datenbankverbindung von einer Anwendung geschlossen, so geht diese für eine bestimmte Zeit in den Pool zurück und kann von anderen Prozessen verwendet werden.⁷² Für Websysteme wie ISDOS, die kontinuierlich mit der Datenbank kommunizieren, ist dieser Mechanismus von Vorteil, weil Datenbankverbindungen nicht jedes Mal erneut aufgebaut werden müssen und dadurch die Performance erhöht wird.

6.2.4 Sessionmanagement

Da das Hypertext Transfer Protocol (HTTP)⁷³ ein zustandsloses Protokoll ist, wird jede Anforderung einer Webseite (bei welcher der Client eine so genannte HTTP-Anfrage an den Webserver startet) als neue Anforderung behandelt, was z. B. dazu führt, dass die Informationen zu einer Anforderung eines Clients standardmäßig für die folgenden Anforderungen nicht verfügbar sind. Um die Zustandslosigkeit des HTTP-Protokolls zu überwinden, wurde ein Mechanismus benötigt, mit welchem bspw. Seiten- und Statusinformationen über mehrere Anforderungen derselben bzw. verschiedener Webseiten beibehalten werden können. ASP.NET stellt hierfür verschiedene Techniken zur Verfügung, wobei in diesem Abschnitt speziell auf die Funktionsweise und Umsetzung des Sessionmanagements eingegangen wird.

Sessionmanagement kann als ein Mechanismus zur Zustandsverwaltung auf Benutzerebene verstanden werden, welcher die Datenweitergabe zwischen verschiedenen Webseiten für eine Benutzersitzung (Session) ermöglicht (vgl. Schwichtenberg 2006, S. 541). Hierfür bekommt ein Client am Anfang einer Benutzersitzung vom Webserver eine eindeutige Sitzungsnummer (Session ID) zugewiesen. Zudem wird dem Client über die Sitzungsnummer ein so genannter Sitzungszustand zugeordnet, welcher eine Instanz der `HttpSessionState`-Klasse ist und das Speichern von sitzungsspezifischen Informationen als Attribut/Wert-Paare ermöglicht.⁷⁴ Bei der nächsten Anforderung einer Webseite kann die ASP.NET-Sitzungsverwaltung den jeweiligen Benutzer anhand der Sitzungsnummer eindeutig identifizieren.

Für die Speicherung und Weiterreichung der Sitzungsnummer zwischen den jeweiligen Seitenübergängen können verschiedene Techniken eingesetzt werden. Im Rahmen die-

⁷² Wurden alle Verbindungen geschlossen und beendet, gibt es keinen Pool.

⁷³ Das HTTP ist ein standardisiertes Protokoll zur Übertragung von Webseiten und anderen Daten im World Wide Web (WWW).

⁷⁴ Die Sitzungsvariablenwerte werden für die gesamte Dauer der Benutzersitzung beibehalten und sind nur innerhalb der Sitzung sichtbar.

ser Arbeit erfolgt die Weitergabe der Session ID in erster Linie per Cookie⁷⁵, wobei dieser als Textdatei auf dem Client-Rechner abgelegt und bei jeder Seitenanforderung an den Webserver gesendet wird. Falls der Client-Browser keine Cookies unterstützt bzw. diese nicht akzeptiert, wird die Session ID automatisch bei jedem Seitenübergang in der URL eingebaut und zum Client übermittelt (URL-Rewriting). Zur Umsetzung dieses Mechanismus war mit ASP.NET keine Programmierung notwendig. Es musste lediglich die Einstellung `cookieless="AutoDetect"` im `sessionState`-Abschnitt der XML-Konfigurationsdatei festgelegt werden (s. Abb. 6.4).⁷⁶ Zudem wurden mit Hilfe des `sessionState`-Elements zwei weitere Konfigurationseinstellungen für das Sessionmanagement vorgenommen. Die Einstellung `mode="InProc"` bewirkt hierbei, dass die Sitzungstabelle, welche die Zuordnung zwischen den Sitzungsnummern und den dort gespeicherten Attribut/Wert-Paaren enthält, im Hauptspeicher des Webserver vorgehalten wird. Das `timeout`-Attribut ermöglicht die Festlegung der Leerlaufdauer einer Benutzersitzung in Minuten. Verstreicht die dort vorgegebene Zeitspanne, ohne dass vom Client-Rechner eine weitere Seitenanforderung eingeht, wird die Benutzersitzung automatisch beendet.

Dieser `timeout`-Wert muss bei einer Webanwendung immer mit Bedacht gewählt werden. Es gilt hierbei, einen Kompromiss zwischen Benutzerfreundlichkeit und Sicherheit zu finden. Wird ein zu hoher Wert gewählt und der Benutzer verlässt seinen Arbeitsplatz ohne die Benutzersitzung zu beenden, könnte diese z. B. von einem unbefugten Benutzer zur Manipulation von Daten missbraucht werden. Ist der `timeout`-Wert dagegen zu klein, wird die Benutzersitzung schon nach wenigen Minuten Inaktivität abgebrochen und der Benutzer zur Startseite umgeleitet.⁷⁷ Dies kann unvorteilhaft sein, wenn bspw. ein Benutzer eine Bestellung anlegt, jedoch nicht sofort alle für die Eingabe erforderlichen Informationen zur Verfügung hat. Bei der Implementierung von ISDOS wurde die Zeitspanne, nach welcher eine Benutzersitzung ungültig wird, auf 30 Minuten festgelegt.

Für das Speichern und Auslesen der Variablenwerte wird die `Session`-Objektmenge verwendet. Abbildung 6.9 zeigt anhand eines Ausschnitts aus dem Programmcode, wie der Name einer Benutzerrolle im Sitzungszustand gespeichert und wieder ausgelesen wird. Um sicherzustellen, dass sich der gewünschte Wert im Sitzungszustand befindet, wird vor dem Zugriff überprüft, ob dieser überhaupt vorhanden ist. Zudem wird beim

⁷⁵ Cookies sind clientseitig gespeicherte Daten, welche temporär oder permanent sein können und Webseiten beim Austausch zwischen Browser und Webserver begleiten.

⁷⁶ Wird z. B. ausschließlich die Verwendung von URL-Rewriting gewünscht, ist die Einstellung `cookieless="UseUri"` vorzunehmen.

⁷⁷ Die Weiterleitung des Benutzers zur Startseite hängt zudem mit dem implementierten Authentifizierungsmechanismus zusammen, welcher in Abschnitt 6.2.6 ausführlicher betrachtet wird.

Auslesen der Sitzungsvariable, wie dem Programmcodebeispiel zu entnehmen ist, der Typ `Object`, dem alle Variablenwerte angehören, in den Typ `String` umgewandelt.

```

Session("UserRole") = strUserRole           ' Speichern

If Session("UserRole") IsNot Nothing Then
    strUserRole = CType(Session("UserRole"), String) ' Auslesen
End If

```

Abb. 6.9: Speichern und Auslesen des Namens einer Benutzerrolle

An dieser Stelle sei erwähnt, dass in der konkreten Webanwendung noch weitere Mechanismen zur Zustandsverwaltung zum Einsatz kommen. Hierzu zählen bspw. versteckte Felder, so genannte `HiddenFields`, welche im Browser nicht sichtbar dargestellt werden, sowie der von ASP.NET zur Verfügung gestellte Seitenzustand, der auch als `ViewState` bezeichnet wird. Beide Techniken werden in erster Linie zur Speicherung von beliebigen seitenspezifischen Informationen verwendet. Darüber hinaus kommen auch `Cookies` zur Speicherung von sitzungübergreifenden Daten und so genannte Abfragezeichenfolgen, welche an das Ende einer URL angehängt werden, zur Anwendung. Abfragezeichenfolgen beginnen dabei immer mit einem Fragezeichen und können mehrere, über `&`-Zeichen miteinander verknüpfte Attribut/Wert-Paare enthalten. Mit dieser einfachen Möglichkeit wird bspw. in einem Anwendungsfall die ID einer Bestellung von einer Webseite zu einer anderen übergeben und dort verarbeitet. Vertrauliche sowie für die Funktionalität der Webanwendung unverzichtbare Daten werden jedoch nicht mit den vier eben vorgestellten Techniken gespeichert bzw. übergeben, da böswillige Benutzer so vorgehaltene Informationen mehr oder weniger einfach manipulieren könnten. Für solche Daten wird der Sitzungszustand verwendet. Ein Nachteil bei der Verwendung des Sitzungszustands besteht darin, dass die Variablenwerte der Benutzersitzung solange im Hauptspeicher des Webservers verbleiben, bis sie entweder entfernt oder ersetzt werden, was bei zunehmender Auslastung des Webservers die Leistung verringern kann. Daher wurde im Rahmen der Implementierung darauf geachtet, keine unnötig großen Informationsmengen im Sitzungszustand zu speichern sowie Sitzungsvariablen sofort zu löschen, sofern diese nicht mehr gebraucht werden.

6.2.5 Lokalisierung

Für die Lokalisierung von ISDOS kommen, wie in Abschnitt 5.5.3 vorgestellt, globale Ressourcendateien sowie die Datenbanktabelle `TBL_I18N` zum Einsatz. Des Weiteren werden die im Websystem zur Verfügung stehenden Sprachen in der Datenbank, genauer in der Tabelle `TBL_LANGUAGE`, vorgehalten (s. Abb. 5.2). Der Inhalt der zuletzt

genannten Tabelle wird bei jeder Anforderung der Startseite von ISDOS aus der Datenbank ausgelesen und an ein Auswahlfeld gebunden. Ein Benutzer hat somit die Möglichkeit, jederzeit die für die Webanwendung geltende Spracheinstellung mit Hilfe dieses Auswahlfeldes zu ändern.⁷⁸ Damit die ausgewählte Sprache auch auf allen folgenden Webseiten bekannt ist, werden das Kulturkürzel sowie die ID der Sprache, welche zuvor aus der Datenbank ausgelesen wurden, im Benutzersitzungszustand gespeichert. Beendet ein Benutzer eine Browsersitzung und ruft anschließend die Webanwendung erneut auf, so sind seine letzten Spracheinstellungen nicht mehr vorhanden und für die Lokalisierung würde die Defaultsprache Englisch verwendet werden. Um die Benutzerfreundlichkeit zu erhöhen, wäre es von Vorteil die benutzerspezifische Spracheinstellung sitzungübergreifend zu speichern. Genau dies geschieht mit Hilfe eines Cookies. Nach getätigter Sprachauswahl wird das jeweilige Kulturkürzel aus dem Auswahlfeld nicht nur im Sitzungszustand sondern auch in einem Cookie abgelegt. Dabei wird die Gültigkeitsdauer des Cookies auf 31 Tage festgelegt (s. Abb. 6.10). Sind im Browser des Clients Cookies deaktiviert oder werden diese nicht unterstützt, kann die Spracheinstellung bei erneutem Aufruf der Webanwendung nicht wieder hergestellt werden, was dazu führt, dass zur Lokalisierung die Defaultsprache zum Einsatz kommt.

```
Response.Cookies("Lang").Value = ddlLanguage.SelectedValue.ToString
Response.Cookies("Lang").Expires = DateTime.Now.AddDays(31)
```

Abb. 6.10: Speichern der Spracheinstellung in einem Cookie

Nachdem eine Sprache ausgewählt bzw. beim ersten Aufrufen der Webanwendung aus dem Cookie ausgelesen wurde, muss die Sprachinformation dem aktuellen Thread zugewiesen werden.⁷⁹ Dies muss zu einem frühen Zeitpunkt der Webseitengenerierung, entweder in der Ereignisbehandlungsroutine `InitializeCulture()` oder im Konstruktor der Seitenklasse geschehen, weshalb die Webseite nach der Sprachauswahl auch neu zu laden ist. Zudem muss die Sprachinformation in jedem angeforderten ASP.NET-Webform erneut gesetzt werden. Damit der Programmcode für die Methode `InitializeCulture()` nicht in jedem Webform vorgehalten wird, ist im Rahmen der Implementierung die bereits vorgestellte Klasse `clsI18N` umgesetzt worden. Diese Klasse erbt von der Basisklasse `Page` (s. Abschnitt 5.3.2) und enthält die Methode `InitializeCulture()`, wobei diese die von der Basisklasse geerbte Methode gleichen Namens überschreibt. Schließlich musste ein jedes Webform nur noch von der Klasse `clsI18N` anstatt von `Page` abgeleitet werden.

⁷⁸ Zur Auswahl stehen hierbei momentan die Sprachen Englisch und Deutsch.

⁷⁹ ASP.NET verfügt über einen so genannten Threadpool, aus welchem bei jeder Anforderung einer Webseite ein Thread geholt und diesem die Anforderung zugewiesen wird. Ein solcher Thread bleibt dann für die gesamte Dauer der Anforderung von dieser Seite belegt.

Abbildung 6.11 zeigt eine leicht gekürzte Version der `InitializeCulture()`-Methode. Wie dem Programmcodeausschnitt zu entnehmen ist, wird in dieser Methode zuerst überprüft, ob die Sprachinformation im Benutzersitzungszustand vorhanden ist. Wenn dies der Fall ist, wird das Kulturkürzel der Sprache ausgelesen und über eine Instanz der Klasse `CultureInfo` (Namensraum `System.Globalization`) dem aktuellen Thread (Namensraum `System.Threading.Thread.CurrentThread`) zugewiesen. Die Einstellung `CurrentUICulture` ist dabei für die Auswahl der richtigen Ressourcendateien und `Culture` u. a. für die Formatierung von Datums- und Zeitwerten zuständig. Existieren keine Sprachinformationen im Benutzersitzungszustand wird überprüft, ob ein Cookie mit der benutzerspezifischen Sprache vorhanden ist. Sollte dies der Fall sein, wird der Inhalt des Cookies dem aktuellen Thread zugewiesen und anschließend im Sitzungszustand gespeichert. Zudem wird mit der `GetLanguageID()`-Methode der Klasse `clsI18N` die ID der Sprache aus der Datenbank ermittelt und ebenfalls im Sitzungszustand abgelegt.

```
Protected Overrides Sub InitializeCulture()
    If Session("Lang") IsNot Nothing Then
        Threading.Thread.CurrentThread.CurrentUICulture = _
            New Globalization.CultureInfo(Session("Lang").ToString())
        Threading.Thread.CurrentThread.CurrentCulture = _
            New Globalization.CultureInfo(Session("Lang").ToString())
    Else
        If Request.Cookies("Lang") IsNot Nothing Then
            Threading.Thread.CurrentThread.CurrentUICulture = _
                New Globalization.CultureInfo(Request.Cookies("Lang").Value)
            Threading.Thread.CurrentThread.CurrentCulture = _
                New Globalization.CultureInfo(Request.Cookies("Lang").Value)
            Session("Lang") = Request.Cookies("Lang").Value
            Session("LangID") = GetLanguageID(Request.Cookies("Lang").Value)
            ' ...
        End If
    End If
End Sub
```

Abb. 6.11: `InitializeCulture()`-Methode der Klasse `clsI18N`

Im Rahmen der Implementierung wurden acht globale Ressourcendateien, vier für die deutsche und vier für die englische Sprache, angelegt. Prinzipiell hätten auch alle Einträge zu einem Sprach- und Kulturraum jeweils in einer Ressourcendatei vorgehalten werden können. Um jedoch eine einfache Wartbarkeit und Erweiterbarkeit dieser Dateien zu gewährleisten, wurden die einzelnen Spracheinträge aufgeteilt. So existieren pro Sprach- und Kulturraum jeweils eine Ressourcendatei für die Beschriftungen von Schaltflächen-, Text- und Navigationssteuerelementen sowie eine Ressourcendatei für längere Zeichenfolgen, z. B. Dialogtexte und Fehlermeldungen.

Die einzelnen Inhalte der Ressourcendateien werden zur Laufzeit, in Abhängigkeit von der geltenden Spracheinstellung ausgelesen und den entsprechenden Steuerelementen zugewiesen. Dies geschieht auf zwei verschiedenen Wegen. Einmal aus dem Programmcode heraus, wenn z. B. Zeichenfolgen erst zur Laufzeit zusammengesetzt bzw. Fehlermeldungen ausgegeben werden sollen. Wie der Abb. 6.12 zu entnehmen ist, erfolgt der Zugriff auf eine Ressourcendatei dabei ohne Kulturkürzel (da dies im aktuellen Thread bekannt ist) und Dateierweiterung. In diesem Programmcodeausschnitt wird auf die Ressourcendatei zugegriffen, welche alle Spracheinträge für die Textstueerelemente enthält. Der Zugriff erfolgt über die Klasse `Resources` mit Hilfe des eindeutigen Schlüssels des Spracheintrags (`LastChange`). Weiterhin wird das letzte Änderungsdatum der angeforderten Webseite ermittelt, wobei der Datumswert automatisch an das Format des aktuell gültigen Kulturraums angepasst wird. Letztendlich werden die beiden Zeichenfolgen miteinander verknüpft und dem Steuerelement zugewiesen.

```
lblLastChangeDate.Text = Resources.labels.LastChange & ": " & _
    System.IO.File.GetLastWriteTime(Request.PhysicalPath).Date
```

Abb. 6.12: Auslesen eines Spracheintrags aus einer Ressourcendatei

Die zweite Lokalisierungsform soll an dieser Stelle nicht ausführlicher vorgestellt werden. Es sei nur gesagt, dass diese in erster Linie für Zeichenfolgen zum Einsatz kommt, die sich nicht erst zur Laufzeit ergeben. Dabei wird bspw. die statische Texteigenschaft eines Steuerelements nicht aus dem Programmcode der Hintergrundcodedatei heraus, sondern innerhalb der ASPX-Datei bei der Deklaration des Steuerelements durch einen speziellen Ausdruck festgelegt. Der Zugriff auf den jeweiligen Spracheintrag der Ressourcendatei erfolgt ebenfalls mit Hilfe des eindeutigen Schlüssels.

Abschließend wird kurz auf die Lokalisierung des Websystems mit Hilfe der Datenbank eingegangen. Wie bereits erwähnt, werden alle hierfür benötigten Zeichenfolgen in der Tabelle `TBL_I18N` vorgehalten (s. Abb. 5.2), zur Laufzeit aus der Datenbank ausgelesen und in erster Linie an Listenstueerelemente, wie z. B. Auswahl- und Listenfelder gebunden. Um die Zeichenfolgen für ein bestimmtes Listenstueerelement in der richtigen Sprache auslesen zu können, ist in der Regel eine Datenbankabfrage mit einer Verbundbedingung notwendig. So wird in einer Abfrage bspw. die Tabelle `TBL_STATUS`, welche die Auswahlmöglichkeiten für ein spezielles Listenstueerelement enthält, mit der Tabelle `TBL_I18N` über die gemeinsamen Attributwerte für die Spalte `ID_I18N` verknüpft. Die Ergebnismenge besteht aus allen für das Steuerelement zur Verfügung stehenden Zeichenfolgen, weshalb letztendlich eine Selektion über die ID der aktuell gültigen Sprache (`ID_LANGUAGE`) durchzuführen ist. An dieser Stelle wird deutlich, warum zusätzlich zum Kulturkürzel die eindeutige ID der Sprache im Sitzungszustand

gespeichert ist. Wäre dies nicht der Fall müsste zusätzlich die Tabelle TBL_LANGUAGE in den Verbund aufgenommen werden, damit letztendlich eine Selektion über das Kulturkürzel möglich wäre. Zur weiteren Vereinfachung wurde im Rahmen der Realisierung für jede Datenbankabfrage, die zur Ermittlung von lokalisierten Zeichenfolgen dient, eine Sicht definiert. Insgesamt sind 13 Sichten in der Datenbank angelegt worden, wobei deren Namen, wie der Abb. 6.1 zu entnehmen ist, mit „QRY_I18N“ beginnen. Ein in Abschnitt 6.1 noch nicht erwähnter Vorteil dieser Vorgehensweise besteht darin, dass die in den Sichten definierten Abfragen vom DBMS bereits bei ihrer Erstellung auf ihre syntaktische Korrektheit überprüft und intern vom Abfrageoptimierer vereinfacht werden und somit bei der eigentlichen Ausführung der Datenbankabfrage diesbezüglich kein zusätzlicher Aufwand notwendig ist.

6.2.6 Sicherheitsmechanismen

Um das Websystem vor unberechtigtem Zugriff zu schützen, ist es unumgänglich, dass sich ein Benutzer mit Hilfe eines Anmeldedialogs authentifiziert (s. Abschnitt 5.6.1). Der Anmeldedialog, welcher die Eingabe eines Benutzerrollennamens und eines Kennworts erfordert, ist dabei auf der Startseite von ISDOS realisiert.⁸⁰ Die Benutzerrollen und Kennwörter wurden erstmalig durch den Entwickler angelegt. Eine Änderung ist jederzeit über die Webanwendung möglich, kann aber nur von einem Benutzer vorgenommen werden, der über die notwendige Berechtigung verfügt.

Für die Überprüfung der im Anmeldedialog eingegebenen Werte ist die Seitenklasse der Startseite zuständig. Hierbei wird mit einer Datenbankabfrage ermittelt, ob die Kombination aus Benutzerrolle und Kennwort in der Tabelle TBL_ROLE existiert, was natürlich voraussetzt, dass die Namen der Benutzerrollen in der Tabelle eindeutig sind. Um eine schnelle Suche gewährleisten zu können, ist im Rahmen der Realisierung der Datenbank für die Tabellenspalte ROLE ein Index⁸¹ angelegt worden, welcher bewirkt, dass eine gewünschte Benutzerrolle nicht mehr sequentiell gesucht, sondern mit logarithmischem Aufwand gefunden werden kann. Darüber hinaus werden bei der Abfrage nicht die eigentlichen Kennwörter miteinander verglichen, sondern kryptische Zeichenketten, da das eingegebene Kennwort noch vor der Abfrage mittels SHA1 verschlüsselt wird und auch die Kennwörter so innerhalb der Datenbank gespeichert sind.

Wird mit der Datenbankabfrage genau ein Datensatz gefunden, der die Suchkriterien erfüllt, ist nachgewiesen, dass die Kombination aus Benutzerrolle und Kennwort korrekt

⁸⁰ Ein Screenshot der Startseite wird im Rahmen der Vorstellung der Präsentationsschicht gezeigt.

⁸¹ Ein Index ist eine spezielle Zugriffsstruktur, wobei in den meisten DBS in der Regel Baumstrukturen zur Anwendung kommen.

ist. In diesem Fall wird für den Benutzer mit Hilfe der `SetAuthCookie()`-Methode der Klasse `FormsAuthentication` (Namensraum `System.Web.Security`) ein so genanntes Authentifizierungsticket mit dem Namen der Benutzerrolle erzeugt (s. Abb. 6.13) und an den Client gesendet. Dieses Ticket wird dann bei jeder weiteren Anforderung einer Webseite an den Webserver übermittelt, damit sich ein Benutzer beim Aufruf einer zugriffsgeschützten Webseite nicht erneut authentifizieren muss. Wie auch beim Sessionmanagement funktioniert diese Übertragung bei der formularbasierten Authentifizierung mit und ohne Einsatz von Cookies. Falls ein Client-Browser keine Cookies unterstützt bzw. diese nicht akzeptiert, wird auch hier die Authentifizierungsinformation in der URL an den Webserver übermittelt.

```
FormsAuthentication.SetAuthCookie(strUserName, false)
```

Abb. 6.13: Erzeugung eines Authentifizierungstickets⁸²

Die Konfigurationseinstellungen für die formularbasierte Authentifizierung wurden ebenfalls in der zentralen Webkonfigurationsdatei, genauer gesagt innerhalb des `authentication`-Abschnitts, vorgenommen (s. Abb. 6.4). Neben der Festlegung des konkreten Authentifizierungsverfahrens durch das Attribut `mode="Forms"` wurden mit Hilfe des `forms`-Elements unterschiedliche Einstellungen definiert. Die Attribute `cookieless="AutoDetect"` und `timeout="30"` haben dabei die gleiche Aufgabe und Bedeutung, wie bei der Konfiguration für das Sessionmanagement und werden daher nicht erneut beschrieben (s. Abschnitt 6.2.4). Es wurden außerdem noch drei weitere Konfigurationseinstellungen vorgenommen. So wurde mit Hilfe des `name`-Attributs ein eindeutiger Name für den Authentifizierungscookie und durch das `path`-Attribut der Pfad für diesen von der Webanwendung erzeugten Cookie definiert. Die letzte Konfigurationseinstellung `protection="All"` legt fest, dass zum Schutz des Cookies sowohl eine Datenüberprüfung als auch eine Verschlüsselung durchgeführt wird, so dass der Inhalt eines verschlüsselten Cookies z. B. nicht während der Übertragung geändert werden kann. Auf die in diesem Zusammenhang zum Einsatz kommenden Algorithmen wird an dieser Stelle nicht eingegangen. Vielmehr soll im Folgenden die Umsetzung der Zugriffskontrollmechanismen näher betrachtet werden.

Wie bereits im Abschnitt 5.6.2 vorgestellt, sind den Benutzerrollen bestimmte Rechte zugeordnet, welche in der Datenbank verwaltet werden. Bevor geprüft werden kann, ob ein authentifizierter Benutzer auf ein von ihm angefragtes Objekt zugreifen darf oder nicht, müssen die Rechte, welche mit seiner Benutzerrolle verbunden sind, aus der Da-

⁸² Mit Hilfe des zweiten Parameters der `SetAuthCookie()`-Methode könnte ein permanenter Cookie gespeichert werden. Im Rahmen der Implementierung wurde jedoch die Standardeinstellung beibehalten und ein sitzungsbasierter Cookie verwendet.

tenbank ausgelesen werden. Die Rechte werden jedoch nicht vor jeder Überprüfung neu ermittelt, sondern nur einmal direkt nach der erfolgreichen Anmeldung, genauer gesagt nach dem Setzen des Authentifizierungstickets. Für die hierfür notwendige Datenbankabfrage, bei der fünf Tabellen⁸³ über Verbundbedingungen miteinander zu verknüpfen sind, wurde im Rahmen der Datenbankrealisierung eine Sicht definiert. Nachdem die Rechte ermittelt wurden, werden diese gemeinsam in einer Zeichenkette gespeichert, wobei die einzelnen Rechte über das spezielle Zeichen „|“ voneinander getrennt sind. Anschließend wird diese Zeichenkette im Benutzersitzungszustand abgelegt, welcher wie bereits erwähnt dieselbe Gültigkeitsdauer wie das Authentifizierungsticket besitzt.

Für das Informationssystem ISDOS wurden verschiedene Zugriffskontrollmechanismen implementiert. So wird bspw. bei fast jedem Webseitenaufruf (ausgenommen beim Aufruf der Startseite und der Webseite mit den Kontaktdaten) nicht nur überprüft, ob ein Benutzer authentifiziert ist, sondern auch das benötigte Recht für den Zugriff auf diese Webseite hat. Für diese Überprüfung ist die zentrale Klasse `clsAuthFilter`, genauer die Methode `CheckAuthenticationAndAuthorisation()` zuständig, welche immer beim Laden einer zugriffsgeschützten Webseite aufgerufen wird. Als Parameter werden der Methode dabei die Instanz der jeweiligen Seitenklasse (`objPage`) und das für die Webseite notwendige Recht (`strRight`) übergeben. In Abb. 6.14 ist ein Ausschnitt aus dieser Methode dargestellt. Wie dem Programmcode zu entnehmen ist, wird zuerst mit Hilfe der `IsAuthenticated`-Eigenschaft überprüft, ob eine Authentifizierung stattgefunden hat. Ist der Benutzer bspw. nicht authentifiziert, werden die URL der angeforderten Webseite sowie die Abfragezeichenfolgen ermittelt und im Sitzungszustand zwischengespeichert. Anschließend wird der Benutzer zur Startseite (`default.aspx`) umgeleitet.

```

If objPage.User.Identity.IsAuthenticated = False Then
    ' ...
    strTemp = objPage.Request.Url.LocalPath & "?" & _
              objPage.Request.QueryString.ToString
    objPage.Session("RedirectURL") = strTemp
    objPage.Response.Redirect("default.aspx")
Else
    ' ...
    strTemp = objPage.Session("Rights").ToString.ToLower
    If strTemp.Contains(strRight.ToLower & "|") = False Then
        objPage.Response.Redirect("error_eval.aspx?ErrCode=NoAccess")
    End If
End If

```

Abb. 6.14: Überprüfung der Authentifizierung und Autorisierung in der Klasse `clsAuthFilter`

⁸³ Die Rede ist von den fünf Tabellen: `TBL_ROLE`, `TBL_ROLE_HAS_RIGHT`, `TBL_RIGHT`, `TBL_ACTION` und `TBL_OBJECT` (s. Abb. 5.2).

Hat er sich dort erfolgreich authentifiziert, wird der Benutzer wieder zurück auf die eigentlich angeforderte Webseite geleitet, wo die Methode `CheckAuthenticationAndAuthorisation()` erneut ausgeführt wird. Gemäß dem Fall, dass er nun authentifiziert ist, wird ermittelt, ob das der Methode übergebene Recht in der Zeichenkette vorkommt, welche im Benutzersitzungszustand gespeichert ist und alle Rechte des Benutzers enthält. Ist diese Überprüfung erfolgreich abgeschlossen, wird die Methode beendet und die angeforderte Webseite geladen. Verfügt der Benutzer dagegen nicht über das benötigte Recht für diese Webseite, so wird er auf die zentrale Fehlerbehandlungsseite (`error_eval.aspx`) umgeleitet. Diese wird im Rahmen der Arbeit jedoch nicht näher betrachtet. Es sei nur erwähnt, dass mit Hilfe der Fehlerbehandlungsseite ein zuvor abgefangener Laufzeitfehler oder eine anderweitige Fehlersituation ausgewertet und dem Benutzer anschließend eine aussagekräftige Fehlermeldung angezeigt wird. Solche Fehler können bspw. aufgrund unauthorisierter Zugriffe auf Webseiten oder auch wegen unzulässiger Benutzereingaben (z. B. durch das Anlegen eines bereits in der Datenbank existierenden Kunden) auftreten.

Die `CheckAuthenticationAndAuthorisation()`-Methode der Klasse `clsAuthFilter` stellt folglich sicher, dass alle Benutzer, welche mit ISDOS arbeiten, auch authentifiziert und dazu autorisiert sind. Ruft ein nicht-authentifizierter Benutzer eine zugriffsgeschützte Webseite bspw. direkt über die Eingabe der Webadresse oder über die im Browser gespeicherten Favoriten auf, so wird auch hier der Benutzer automatisch zur Startseite umgeleitet.

Darüber hinaus wird bereits beim Laden der Menüleiste des Informationssystems auf ähnliche Weise wie durch die `CheckAuthenticationAndAuthorisation()`-Methode überprüft, ob ein Benutzer die benötigten Rechte zum Aufrufen der dort hinterlegten Webseiten hat. Hierbei werden in der Hintergrundcodeklasse der zentralen Masterpage die einzelnen Menüpunkte in einer `For-Each`-Schleife durchlaufen, wobei ebenfalls mit Hilfe der Anweisung `strTemp.Contains(strRight.ToLower & "|")` ermittelt wird, ob das zu einem Menüpunkt gehörende Recht in der im Benutzerzustand gespeicherten Zeichenkette vorhanden ist. Hat ein Benutzer nicht das benötigte Recht, so wird der jeweilige Menüpunkt deaktiviert und ausgegraut. Eine ähnliche Vorgehensweise wird auch auf einigen Webseiten von ISDOS zur Ausblendung von bestimmten Schaltflächen verwendet. Somit stehen ausgewählte Schaltflächen nur den Benutzern zur Verfügung, die auch das notwendige Recht zum Ausführen der dort hinterlegten Funktionalität haben.

6.3 Präsentationsschicht

Die Benutzeroberfläche von ISDOS wird in einem Browser dargestellt, der in der Lage sein muss JavaScript⁸⁴ auszuführen. Wie bereits erwähnt, wird bei der Anforderung einer Webseite eine HTTP-Anfrage vom Browser an den Webserver gesendet, welcher diese an das jeweilige ASP.NET-Webform weiterleitet. Der Webserver, genauer gesagt das ASP.NET Page Framework, führt schließlich das Webform bestehend aus ASPX-Datei und Hintergrundcodedatei aus. In diesem Zusammenhang werden die ASP.NET-Serversteuerelemente, die in der ASPX-Datei enthaltenen sind, in HTML-Tags umgewandelt. So wird z. B. aus dem Serversteuerelement `<asp:Textbox ...>` das HTML-Tag `<input type=text ...>` erzeugt. Die generierte Webseite (HTTP-Antwort) enthält schließlich nur noch HTML, JavaScript, CSS und Zustandsdaten (z. B. Cookies), wenn diese an den Browser übertragen wird. Zusätzlich zu dem von ASP.NET automatisch erzeugten JavaScript-Code wurden im Rahmen der Realisierung eigene JavaScript-Funktionen implementiert. Diese werden in der Datei mit der Bezeichnung `utilities.js` vorgehalten, welche in verschiedenen ASPX-Dateien eingebunden ist.

Auf die Darstellung von Details zur Implementierung der JavaScript-Funktionen und der ASPX-Dateien soll an dieser Stelle verzichtet werden. Im Folgenden wird vielmehr das entwickelte Informationssystem anhand von ausgewählten Screenshots vorgestellt, wobei im Wesentlichen die für den Benutzer sichtbaren Funktionen und Eigenschaften beschrieben werden. Der grundlegende Aufbau der Benutzeroberfläche wird in diesem Zusammenhang nicht mehr betrachtet, da dies bereits in Abschnitt 5.4.2 geschehen ist.

6.3.1 Startseite

Nachdem das webbasierte Informationssystem, z. B. durch Eingabe einer Webadresse im Browser oder durch Aufruf eines im Browser gespeicherten Favoriten, aufgerufen wurde, erscheint in der Regel die Startseite, da alle nicht-authentifizierten Benutzer, auch wenn diese die Startseite nicht direkt aufgerufen haben, dorthin umgeleitet werden. Abbildung 6.15 zeigt diese Seite, die im Inhaltsbereich einen Anmeldedialog enthält. Zusätzlich ist es einem Benutzer an dieser Stelle möglich mit Hilfe eines Auswahlfeldes eine andere Sprache für die Lokalisierung von ISDOS festzulegen.

⁸⁴ JavaScript ist eine standardisierte, clientseitige Scriptsprache, welche in jeder Betriebssystemumgebung ausgeführt werden kann, sofern ein Browser installiert ist, der ein solches Script interpretieren kann (vgl. Lubkowitz 2005, S. 373 f.). Ist dies der Fall, so kann JavaScript z. B. auf alle Elemente einer im Browser geladenen Webseite zugreifen bzw. dessen Elemente ändern.

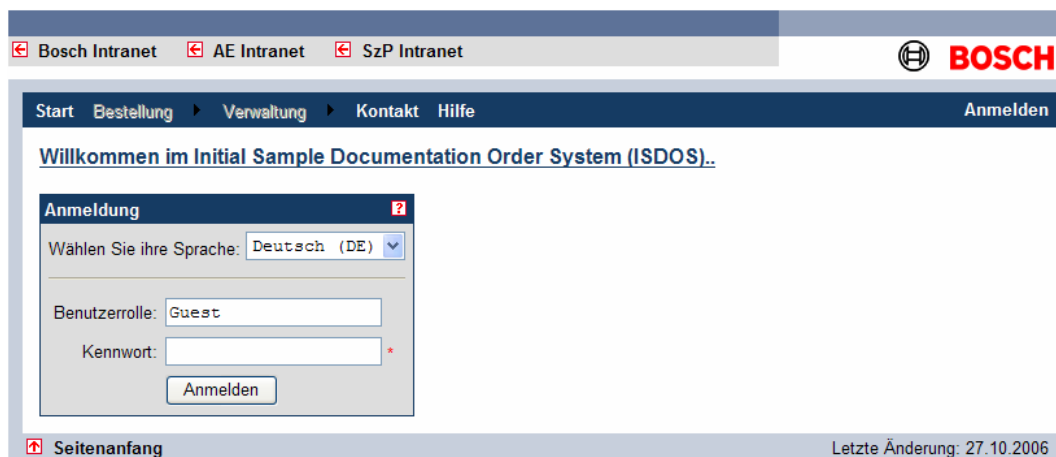


Abb. 6.15: Startseite mit Anmeldedialog

Wie in dem Screenshot zu erkennen, sind die Menüpunkte „Bestellung“ und „Verwaltung“ deaktiviert und ausgegraut, sofern ein Benutzer nicht authentifiziert ist. Um sich im Websystem anzumelden, muss ein Benutzer den Namen der Benutzerrolle, welche er innehaben möchte, sowie das dazugehörige Kennwort eingeben und anschließend die Schaltfläche „Anmelden“ betätigen. Sind beide Eingabefelder ausgefüllt, wird das Eingabeformular an den Webserver übertragen und dort verarbeitet. Ist dagegen eines oder sind beide Eingabefelder leer, so wird die Übertragung des Formulars unterbunden und hinter dem jeweils betroffenen Eingabefeld erscheint ein rotes Sternchen. Die Validierung der Eingabefelder ist mittels JavaScript realisiert, was den Vorteil hat, dass die Fehlermeldung, in diesem Zusammenhang ein rotes Sternchen, auf dem Client ohne Kommunikation mit dem Webserver erzeugt wird, was zusätzlichen Netzverkehr und Zeitaufwand einspart. Ohne den Einsatz von JavaScript müsste das Eingabeformular zur Überprüfung auf Vollständigkeit zunächst an den Webserver übermittelt und im Fehlerfall zusammen mit den Fehlermeldungen wieder zurück übertragen werden.

An dieser Stelle sei gesagt, dass in diesem Websystem fehlende sowie ungültige Benutzereingaben (z. B. fehlerhafte E-Mail-Adresse, falscher Datentyp), welche Laufzeitfehler hervorrufen können, in der Regel schon clientseitig mittels JavaScript abgefangen werden. Handelt es sich dagegen z. B. um unzulässige Benutzereingaben (etwa wenn ein Benutzer versucht einen bereits in der Datenbank gespeicherten Kunden erneut anzulegen) oder unautorisierte Zugriffe auf eine Webseite, können diese natürlich nicht mittels JavaScript behandelt werden. In diesen Fällen kommt wie bereits erwähnt die serverseitige Fehlerbehandlungsseite zur Anwendung.

Hat sich ein Benutzer erfolgreich auf der Startseite authentifiziert, werden die beiden Eingabefelder sowie die Schaltfläche zum Anmelden ausgeblendet. Stattdessen erscheint, wie in der Abb. 6.16 zu sehen, die Benutzerrolle, welche der Benutzer eingenommen hat. Das Auswahlfeld für die Sprache bleibt weiterhin sichtbar, so dass der

Benutzer auf der Startseite jederzeit die Spracheinstellung ändern kann. Darüber hinaus sind in der Menüleiste nun alle Menüpunkte freigegeben, auf welche der angemeldete Benutzer zugreifen darf. Wie dem Screenshot entnommen werden kann, darf ein als Gast angemeldeter Benutzer gespeicherte EM-Bestellungen öffnen, jedoch weder Bestellungen neu anlegen noch irgendwelche administrativen Funktionen ausführen.⁸⁵

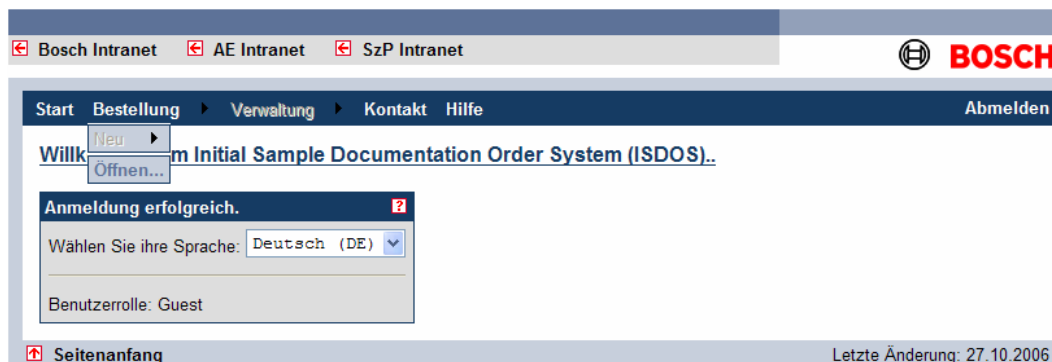



Abb. 6.16: Startseite nach erfolgreicher Anmeldung

6.3.2 Bestellformular

Das Formular zur Erfassung einer Bestellung kann, sofern der Benutzer das benötigte Recht dazu hat, über die Menüleiste aufgerufen werden. Hierbei hat der Benutzer die Möglichkeit ein leeres Formular zu öffnen oder eine bereits gespeicherte EM-Bestellung zu suchen und als Vorlage zu verwenden.

Wie den Abb. 6.17 und 6.18 entnommen werden kann, ist das Eingabeformular für das Erfassen einer Bestellung in mehrere Bereiche unterteilt. Die unterschiedlichen Längen der enthaltenen Eingabefelder sind darin begründet, dass die Länge eines jeden Feldes der maximal in diesem Feld aufnehmbaren Anzahl von Zeichen entspricht. Diese Anzahl ist wiederum abhängig von der Zeichenanzahl, welche im dazugehörenden Feld in der Datenbank gespeichert werden kann. Für mehrzeilige Eingabefelder kann bei der Deklaration des Steuerelements zwar die Länge und die Zeilenanzahl, jedoch nicht die maximale Anzahl der aufnehmbaren Zeichen festgelegt werden. Daher wurde eine JavaScript-Funktion implementiert, die nach jedem Eintrag eines Benutzers in ein solches Eingabefeld dafür sorgt, dass alle Zeichen des Feldinhalts abgeschnitten werden, die einen festgelegten Wert überschreiten.

⁸⁵ Zudem kann ein Benutzer, der als Gast authentifiziert ist, die geöffnete Bestellung nur ansehen, da für diese Benutzerrolle die Schaltflächen zum Ändern, Stornieren und Löschen ausgeblendet sind.

Bosch Intranet AE Intranet SzP Intranet 

Start Bestellung Verwaltung Kontakt Hilfe Abmelden

[Bestellung](#) > [Neu](#) > [Leeres Formular](#)

Technischer Verkaufsbereich ?

Nachname: Müller Abteilung: GS/SDC3
 Vorname: Frank Telefonnr.: 0711-822-46789
 E-Mail: Frank.Mueller-de.bosch.com
 Die E-Mail-Adresse ist nicht korrekt!

[Seitenanfang](#)

Kaufmännischer Verkaufsbereich ?

Nachname: Schmidt Abteilung: GS/SDC2
 Vorname: Peter Telefonnr.: 0711-822-46567
 E-Mail: *

[Seitenanfang](#)

Kunde ?

Kunde und Werk: Porsche | Stuttgart | 761198
 Kundenwuschtermin: 29 02 2007 Das Datum ist fehlerhaft!
 Ansprechpartner (Name):
 Ansprechpartner (E-Mail):

[Seitenanfang](#)

Bestellung ?

EVA-Auftragsnr.: 377789 Kundenauftragsnr.: 89660-G02G60
 Die EVA-Auftragsnr. muss eine 7-stellige Zahl sein!
 EVA-Auftragsdatum: 13 12 2006 Kundenauftragsdatum: 12 12 2006
 Anzahl Erstmuster: 4 Anzahl Referenzmuster: 1

[Seitenanfang](#)

Produkt ?

Lieferant / Produktionsstandort:	Kunde:
Benennung: EDC16CP318.30	Benennung:
Sachnr.: 0261.201.466	Sachnr.: A 285 150 03 79
Änderungsnr.: 1039.S19.44	Änderungsnr.:
Die Eingabe muss eine 12-stellige Zeichenfolge sein!	
EPROM-Nr.:	Datenstand:
Datenstand:	Datenstand:
Zeichnungsnr.: 0261.A00.171	Zeichnungsnr.:
Zeichnungsstand: 004	Zeichnungsstand:
Zeichnungsdatum: 18 04 2006	Zeichnungsdatum:
Sonstiges: TKU RB: 0261.K00.550	Sonstiges:

[Seitenanfang](#)

Abb. 6.17: Formular zur Erfassung einer Bestellung – Teil 1

Kennzeichnungen und Aufkleber ?

Verpackung: Bosch Kundenspezifisch

Bemerkung:

Produkt: Bosch Kundenspezifisch

Bemerkung:

Seitenanfang

Bemusterungsverfahren ?

PPF (VDA) PPAP (QS9000) Kundenspezifisch

Beim Ändern des Bemusterungsverfahrens werden die aktuellen Eingaben unter "PPF (nach VDA)" verworfen.

Seitenanfang

PPF (VDA) ?

Bericht PPF Bericht sonstige Muster DmbA

Vorlagestufe:

Bemusterung Nachbemusterung Neubemusterung

Neuteil Werkzeugänderung / -korrektur
 Produktänderung (Spezifikationsänderung) Änderung von Zukaufteilen
 Produktionsverlagerung Änderung von Lieferanten
 Änderung von Produktionsprozessen Sonstiges - Bitte unten spezifizieren
 Aussetzen der Fertigung länger als 12 Monate

Prüfungen / Dokumentation

<input checked="" type="checkbox"/> 01 Maßprüfung	<input type="checkbox"/> 09 EMV - Prüfung	<input type="checkbox"/> 17 Prüfmittelliste
<input type="checkbox"/> 02 Funktionsprüfung	<input type="checkbox"/> 10 Zuverlässigkeitsprüfungen	<input type="checkbox"/> 18 Prüfmittelfähigkeitsnachweis
<input type="checkbox"/> 03 Werkstoffprüfung	<input type="checkbox"/> 11 Design - FMEA	<input type="checkbox"/> 19 EU-Sicherheitsdatenblatt
<input type="checkbox"/> 04 Haptikprüfung	<input type="checkbox"/> 12 Konstruktionsfreigabe	<input checked="" type="checkbox"/> 20 Materialdatenblatt / IMDS
<input type="checkbox"/> 05 Akustikprüfung	<input type="checkbox"/> 13 Prozess - FMEA	<input type="checkbox"/> 21 Transportmittel / Verpackung
<input type="checkbox"/> 06 Geruchsprüfung	<input checked="" type="checkbox"/> 14 Prozessablaufdiagramm	<input type="checkbox"/> 22 Zertifikate
<input type="checkbox"/> 07 Aussehenprüfung	<input type="checkbox"/> 15 Produktionslenkungsplan	<input type="checkbox"/> 24 Prozessabnahme
<input type="checkbox"/> 08 Oberflächenprüfung	<input checked="" type="checkbox"/> 16 Prozessfähigkeitsnachweis	<input type="checkbox"/> 24 Sonstiges - Bitte unten spezifizieren

Seitenanfang

Bemerkungen ?

[Zur Startseite](#)

Seitenanfang
Letzte Änderung: 30.10.2006

Abb. 6.18: Formular zur Erfassung einer Bestellung – Teil 2

Da auch mit ISDOS eine Bestellung in der Regel von zwei Vertriebsmitarbeitern aufgenommen wird, enthält das Formular im oberen Teil jeweils einen separaten Bereich für die Eingabe der Vertriebsmitarbeiterdaten. Im ersten Bereich können die Daten des technischen, im zweiten die des kaufmännischen Vertriebsmitarbeiters erfasst werden.

Die E-Mail-Adresse eines Mitarbeiters ist hierbei von besonderer Bedeutung, weil diese RB-intern für jeden Mitarbeiter eindeutig ist. Sollen Daten zu einem Vertriebsmitarbeiter gespeichert werden, so ist die E-Mail-Adresse immer mit aufzunehmen. Um dies sicherzustellen, ist eine JavaScript-Funktion realisiert worden, welche überprüft, ob das Eingabefeld für die E-Mail-Adresse nicht leer ist, sobald mindestens eines der anderen vier Eingabefelder einen Eintrag enthält. Des Weiteren wird in diesem Zusammenhang mittels JavaScript die Korrektheit der eingegebenen E-Mail-Adresse geprüft.

In den folgenden beiden Bereichen des Formulars können mittels diverser Eingabe- und Auswahlfelder Daten zum Kunden und zur Bestellung erfasst werden. Dies sind bspw. Kundennamen, -werk, -wunschtermin, EVA-Auftragsnummer sowie die Anzahl der Erst- und Referenzmuster. Referenzmuster sind spezielle Muster, die nach der Bemusterung zwecks möglicher Rückfragen im Werk verbleiben. In der Regel wird zusätzlich zu den Erstmustern ein Referenzmuster bestellt, weshalb dieser Wert auch beim Öffnen des Formulars voreingestellt ist. Darüber hinaus werden die eingegebenen Datumswerte wie auch im gesamten Formular durch eine JavaScript-Funktion auf ihre Richtigkeit geprüft. Daneben findet eine Überprüfung der Eingabefelder für die EVA-Auftragsnummer sowie die Anzahl der Erst- und Referenzmuster statt, welche sicherstellt, dass die EVA-Auftragsnummer eine 7-stellige Zahl und die jeweils bestellte Musteranzahl ein Zahlenwert ist. Wird in einem dieser Felder eine ungültige Benutzereingabe vorgenommen, ist sofort nach dem Verlassen des betreffenden Feldes eine aussagekräftige Fehlermeldung sichtbar.

Weiterhin können mit dem Formular Informationen zum Produkt festgehalten werden, wobei der betreffende Bereich zweigeteilt ist. Auf der linken Seite werden die Produktdaten des Lieferanten (in diesem Fall von RB), auf der rechten die Daten des Kunden erfasst. Auch an dieser Stelle sind diverse JavaScript-Funktionen hinterlegt. So wird z. B. überprüft, ob die Einträge für die RB-Sach-, RB-Änderungs- und RB-Zeichnungsnummer jeweils 12-stellige Zeichenfolgen sind. Sollte dies nicht der Fall sein, wird auch hier eine Fehlermeldung ausgegeben. Darüber hinaus sorgt eine JavaScript-Funktion beim Verlassen des jeweiligen Eingabefeldes dafür, dass die eingegebenen Nummern richtig formatiert werden. Wie bereits erwähnt, hat eine RB-Sachnummer das Format: 4 Zeichen | Punkt | 3 Zeichen | Punkt | 3 Zeichen (z. B. 0281.200.300), dies gilt ebenfalls für die RB-Änderungs- und RB-Zeichnungsnummer. Da diese Nummern meist nur aus Zahlenwerten bestehen, kann ein Benutzer diese bequem mit Hilfe des Ziffernblocks der Tastatur eingeben, wobei die Punkte nachträglich automatisch eingefügt werden. Zudem gewährleistet die JavaScript-Funktion, dass nicht erlaubte Zeichen entfernt werden. Wie bei den drei Nummern findet auch eine clientseitige Formatierung

des Eintrages für die RB-Benennung statt, wobei in diesem Zusammenhang beim Verlassen des Eingabefeldes alle Leerzeichen entfernt werden.

Wie bereits beschrieben, müssen Erstmuster besonders gekennzeichnet werden. Dies gilt auch für die Verpackung der Erstmuster und wird in der Regel durch Aufkleber gewährleistet. Der hierbei zum Einsatz kommende Aufkleber wird in diesem Formular mit Hilfe eines Optionsfeldes festgelegt. Ist z. B. ein kundenspezifischer Aufkleber gewünscht, kann zusätzlich eine Bemerkung in einem Eingabefeld aufgenommen werden.

Im darauf folgenden Bereich des Formulars kann das Bemusterungsverfahren festgelegt werden, wobei hier zwischen dem PPF-, PPAP- und einem kundenspezifischen Verfahren gewählt werden kann. Je nach erfolgter Auswahl erscheint anschließend ein anderer Block, der ausgefüllt werden muss. In dem in Abb. 6.18 dargestellten Formular wurde das PPF-Verfahren gewählt. Die Abb. 6.19 und 6.20 zeigen jeweils den Block, welcher zu sehen wäre, würde das PPAP-Verfahren bzw. das kundenspezifische Verfahren ausgesucht werden.

PPAP (QS9000)	
Grund der Vorlage	
<input type="checkbox"/> Erstmalige Vorlage	<input type="checkbox"/> Änderung zur optionalen Konstruktion oder Werkstoff
<input type="checkbox"/> Technische Änderung(en)	<input type="checkbox"/> Änderung von Unterlieferant oder Lieferquelle des Werkstoffes
<input type="checkbox"/> Werkzeug: Verlagerung, Ersatz, Überholung oder sonstiges	<input type="checkbox"/> Änderung im Bearbeitungsprozess
<input type="checkbox"/> Korrektur eines Fehlers	<input type="checkbox"/> Teile werden an einem zweiten Standort hergestellt
<input type="checkbox"/> Werkzeug für mehr als ein Jahr inaktiv	<input type="checkbox"/> Sonstiges - Bitte unten spezifizieren
Vorlagestufe	
<input checked="" type="radio"/> 1. Stufe - Bestätigung und für ausgewiesene Aussehensteile Bericht zur Freigabe des Aussehens.	
<input type="radio"/> 2. Stufe - Bestätigung mit Musterteilen und eingeschränkte unterstützende Daten werden dem Kunden vorgelegt.	
<input type="radio"/> 3. Stufe - Bestätigung mit Musterteilen und umfassende unterstützende Daten werden dem Kunden vorgelegt.	
<input type="radio"/> 4. Stufe - Bestätigung und andere Forderungen wie sie vom Kunden festgelegt wurden.	
<input type="radio"/> 5. Stufe - Bestätigung mit Musterteilen und vollständige unterstützende Daten, die am Produktionsstandort des Lieferanten bewertet werden.	
Forderungen	
<input type="checkbox"/> 01 Design-Aufzeichnungen	<input type="checkbox"/> 10 Aufzeichnungen von Material- und Leistungstests
<input type="checkbox"/> - für geschützte Komponenten	<input type="checkbox"/> 11 Untersuchungen zur Kurzzeitfähigkeit des Prozesses
<input type="checkbox"/> - für alle anderweitigen Komponenten	<input type="checkbox"/> 12 Dokumentation eines qualifizierten Laboratoriums
<input type="checkbox"/> 02 Design-Änderungsdokumente	<input type="checkbox"/> 13 Bericht zur Freigabe des Aussehens
<input type="checkbox"/> 03 Technische-Freigabe durch den Kunden	<input type="checkbox"/> 14 Muster-Serierteile
<input type="checkbox"/> 04 Design - FMEA	<input type="checkbox"/> 15 Referenzmuster (Bezugsmuster)
<input type="checkbox"/> 05 Prozessflussdiagramme	<input type="checkbox"/> 16 Spezifische Prüfmittel
<input type="checkbox"/> 06 Prozess - FMEA	<input type="checkbox"/> 17 Unterlagen über Einhaltung von kundenspezifischen Forderungen
<input type="checkbox"/> 07 Prüfplan	<input type="checkbox"/> 18 Teilevorgabe-Bestätigung
<input type="checkbox"/> 08 Analyse von Messsystemen	<input type="checkbox"/> - Checkliste für Forderungen an verfahrenstechnische Produkte
<input type="checkbox"/> 09 Messergebnisse	

Abb. 6.19: Eingabeblock für das PPAP-Verfahren

A screenshot of a software window titled "Kundenspezifisch". The window has a dark blue title bar with a small red icon on the right. Below the title bar, there is a light gray area containing the text "Beschreibung / Formulare:" followed by a large, empty white text input field with a vertical scrollbar on the right side.

Abb. 6.20: Eingabeblock für ein kundenspezifisches Bemusterungsverfahren

Inhalt und Aufbau der Eingabeblocke zum PPF- und PPAP-Verfahren orientieren sich dabei stark an den Empfehlungen und Forderungen des VDA Bd. 2 bzw. des Referenzhandbuchs PPAP, sodass in dem jeweiligen Eingabeblock über Kontrollkästchenlisten, Options- und Auswahlfelder die Notwendigkeit des Bemusterungsverfahrens sowie der Umfang der durchzuführenden Prüfungen und Dokumentation normgerecht festgelegt werden kann. Im Folgenden wird der Eingabeblock für die Erfassung der Vereinbarungen zum PPF-Verfahren näher betrachtet. Aus Platzgründen können jedoch nicht alle Zusammenhänge und Auswirkungen der in diesem Block getätigten Eingaben erklärt werden.⁸⁶ Aus selbigem Grund wird auf eine Beschreibung der beiden Eingabeblocke zum PPAP-Verfahren und zur kundenspezifischen Bemusterung verzichtet. Es sei jedoch gesagt, dass für das PPAP-Verfahren der Auslöser der Bemusterung, die Vorlagestufe sowie die Forderungen und für das kundenspezifische Verfahren eine Beschreibung z. B. des Verfahrens und der nötigen Formulare aufgenommen werden können.

Ist als Bemusterungsverfahren das PPF-Verfahren ausgewählt worden, kann in einem ersten Schritt über ein Optionsfeld die Berichtsart, also die Art der anzufertigenden Dokumentation, festgelegt werden. Sind bspw. Erstmuster Gegenstand der Bemusterung, ist hier „Bericht PPF“ zu wählen. Sollen dagegen so genannte „sonstige Muster“ bemustert werden, also Produkte, die nicht vollständig unter serienmäßigen Bedingungen hergestellt wurden (z. B. Erprobungsmuster), so ist an dieser Stelle „Bericht sonstige Muster“ auszuwählen. Zudem können über das daneben liegende Kontrollkästchenfeld Dokumente zu einem Produkt als Dokumente mit besonderer Archivierung (DmbA) gekennzeichnet werden, was bedeutet, dass die erstellte Dokumentation auch nach Ablauf der Aufbewahrungszeit nicht vernichtet werden darf. In einem weiteren Schritt können Vorlagestufe und Art der Bemusterung festgelegt werden. Als Neubemusterung wird hierbei die Anwendung des PPF-Verfahrens für Requalifikationsprüfungen von Prozessen und Produkten bezeichnet. Ist die Bemusterungsart erfasst, kann über die darunter liegende Kontrollkästchenliste der Auslöser für das PPF-Verfahren festgehalten werden. Abschließend ermöglicht eine weitere Kontrollkästchenliste die Festlegung des Bemusterungsumfangs, wobei mit dieser die durchzuführenden Prüfungen und die dem Kunden vorzulegenden Dokumente und Aufzeichnungen

⁸⁶ An dieser Stelle sei auf den VDA Bd. 2 verwiesen (s. VDA – Verband der Automobilindustrie e. V. 2004).

erfasst werden können. Prinzipiell können bei der Bemusterung alle in Zeichnungen und Spezifikationen enthaltenden Produktmerkmale (Maß, Funktion, Werkstoff, Haptik, Akustik, Geruch, Aussehen, Oberfläche etc.) geprüft werden, sofern diese anwendbar sind. Obwohl für die Produkte der RB Elektronik GmbH in der Regel nicht alle Prüfungen zur Anwendung kommen, werden im realisierten Bestellformular dem Benutzer alle im VDA Bd. 2 geforderten Prüfungen zur Auswahl gestellt. Dies lässt sich damit begründen, dass ISDOS, wie in den Anforderungen definiert, auch an anderen RB-internen Fertigungsstandorten einsetzbar sein soll und somit bei der Entwicklung darauf geachtet wurde, das Bestellformular wie auch alle anderen Ein- und Ausgabeformulare so allgemein wie möglich zu gestalten.

Der letzte Bereich des Formulars ermöglicht es einem Benutzer abschließende Bemerkungen aufzunehmen. Sind alle Eingaben getätigt und alle clientseitigen Gültigkeitsprüfungen erfolgreich abgeschlossen, kann der Benutzer durch Drücken der Schaltfläche „Speichern“ den Inhalt des Formulars in der Datenbank speichern. Können die Daten nicht gespeichert werden, so wird der Benutzer zur zentralen Fehlerbehandlungsseite umgeleitet, wo ihm eine passende Fehlermeldung präsentiert wird. Andernfalls wird er darüber informiert, dass die Bestellung erfolgreich gespeichert wurde. Gleichzeitig bekommt der Benutzer eine Auflistung von Möglichkeiten präsentiert, die ihm zeigt, was er jetzt tun kann (s. Abb. 6.21). So hat der Benutzer bspw. die Möglichkeit, sofern alle für die Bemusterung benötigten Informationen erfasst wurden, die Bestellung an das Werk zu übermitteln. Ist eine Bestellung unvollständig, kann er einen anderen Mitarbeiter über die gespeicherte Bestellung in Kenntnis setzen, so dass dieser die Bestellung weiter bearbeiten kann. Um diesen Abschnitt nicht übermäßig auszuweiten, wird die Webseite zum Versenden der notwendigen E-Mail an dieser Stelle nicht beschrieben.



Abb. 6.21: Information über die erfolgreiche Speicherung einer Bestellung

Des Weiteren soll das Formular zum Öffnen einer gespeicherten Bestellung aus demselben Grund nicht betrachtet werden. Es sei lediglich darauf hingewiesen, dass dieses ebenso aufgebaut ist, wie das in Abb. 6.17 und 6.18 dargestellte. Die Unterschiede bestehen darin, dass bei einer in diesem Formular geöffneten Bestellung im oberen Teil die eindeutige ID sowie der Bearbeitungsstatus der Bestellung angezeigt werden und neben der Schaltfläche zum Speichern, je nach Rechten des Benutzers, weitere Schaltflächen zum Stornieren bzw. Löschen der Bestellung vorhanden sind.

6.3.3 Suchfunktion

Die Suchfunktion von ISDOS kommt zum Einsatz, wenn ein Benutzer eine gespeicherte Bestellung bspw. zur Ansicht öffnen oder als Vorlage zur Erfassung einer neuen Bestellung verwenden möchte. Wird das Suchformular z. B. zum Öffnen einer Bestellung aufgerufen, so ist dies auch aus der Überschrift des Formulars ersichtlich (s. Abb. 6.22). In beiden Fällen steht einem Benutzer zunächst die Standardsuche zur Verfügung, wobei er über einen Verweis im unteren Bereich der Suchmaske in die erweiterte Suche wechseln kann. Hierbei werden die oberen sechs Suchkriterien der Standardsuche, um weitere Kriterien ergänzt.

The screenshot shows a web interface for an extended search. At the top, there are navigation links for 'Bosch Intranet', 'AE Intranet', and 'SzP Intranet', and the Bosch logo. Below that is a breadcrumb trail: 'Start > Bestellung > Verwaltung > Kontakt > Hilfe' and a 'Abmelden' button. The main content area is titled 'Bestellung > Öffnen...' and contains a form titled 'Erweiterte Suche'. The form has the following fields:

- RB-Benennung:
- RB-Sachnr.:
- RB-Änderungsnr.:
- ISDOS-Nr.:
- Kunde:
- Kundensachnr.:
- Nachname des Ausstellers:
- Kundenwerk:
- Abteilung des Ausstellers:
- Kundenauftragsnr.:
- EVA-Auftragsnr.:
- Bemusterungsverfahren:
- Bearbeitungsstatus:
- Vorlagestufe:

At the bottom of the form, there is a hint: 'Hinweis: ? - Platzhalter für ein Zeichen * - Platzhalter für eine beliebige Zeichenfolge'. Below the form, there are three buttons: 'Erweiterte Suche', 'Zuletzt bearbeitete Bestellung', and 'Suche starten >'. The footer of the page includes 'Seitenanfang' and 'Letzte Änderung: 30.10.2006'.

Abb. 6.22: Suchformular für die erweiterten Suche

Wie der Abb. 6.22 zu entnehmen ist, können Suchkriterien sowohl über Eingabe- als auch Auswahlfelder festgelegt werden. Sollte ein Benutzer nicht genau wissen, wonach er sucht, hat er zudem die Möglichkeit spezielle Platzhalter in den Eingabefeldern zu

benutzen. Hierbei gilt ein Fragezeichen („?“) als Platzhalter für ein Zeichen und ein Stern („*“) als Platzhalter für eine beliebige Zeichenfolge. Dies ermöglicht es einem Benutzer bspw. nach allen EM-Bestellungen zu suchen, die sich auf all jene Produkte beziehen, welche von der RB-Benennung her mit der Zeichenfolge „EDC“ beginnen. Des Weiteren kann ein Benutzer im unteren Bereich des Suchformulars ohne Auslösen der eigentlichen Suche auf die von ihm zuletzt bearbeitete Bestellung zugreifen. Diese Funktionalität wurde mit Hilfe eines Cookies realisiert, in welchem die ID der betreffenden Bestellung sitzungsübergreifend vorgehalten wird. Ist mindestens ein Suchkriterium vom Benutzer festgelegt worden, so kann über die Schaltfläche „Suche starten“ die Suche in der Datenbank angestoßen werden.

Auf einer neuen Webseite wird das Suchergebnis einer Anfrage in tabellarischer Form angezeigt, wobei die einzelnen Spaltennamen den Suchkriterien der Standardsuche entsprechen (s. Abb. 6.23).⁸⁷ Zudem befindet sich am Ende einer jeden Zeile ein kleines Symbol, mit dem die gewünschte Bestellung durch Anklicken geöffnet werden kann.

ISDOS-Nr.	RB-Sachnr.	RB-Benennung	RB-Änderungsnr.	Kunde	Kundensachnr.	
641	0261.201.209	ECUM7.9.41		Toyota	89661-02E90	
681	0261.201.209	ECUM7.9.41		Toyota	89661-02E90	
661	0261.201.371	ECUM7.9.41		Toyota	89660-02F00	
961	0261.201.383	ME770		Smart	A132 153 02 79	
1021	0261.201.383	Steuengerät		Smart	A132 153 09 79	
901	0261.201.463	Me2.7.2	1039.S19.424	DCC	A 275 150 09 79	
902	0261.201.464	Me2.7.2	1039.S19.449	DCC	A 285 150 03 79	
701	0261.208.840	ME7.7.1(GMT)		Mitsubishi	1860A705	
905	0261.209.246	Me9.7	1039.S20.026	DCC	A 237 153 46 79	
941	0261.209.247	Me9.7	1039.S20.026	DCC	A 237 153 47 79	
904	0261.209.248	Me9.7		DCC	A 273 153 45 79	
1001	0261.209.255	Me9.7		AMG	A 156 150 16 79	
903	0261.209.272	Me9.7		AMG	A 156 150 35 79	

Ergebnisseite: 1 2

Zurück

Seitenanfang Letzte Änderung: 30.10.2006

Abb. 6.23: Ergebnisanzeige nach erfolgreicher Suche⁸⁸

Zur besseren Übersicht werden große Ergebnismengen nicht in Form einer einzelnen Liste dargestellt, sondern auf relativ kurze Seiten gleicher Größe aufgeteilt. Der Zugriff auf die jeweiligen Ergebnisseiten erfolgt über die sich unterhalb der Ergebnismenge

⁸⁷ Können keine Bestellungen gefunden werden, welche die festgelegten Kriterien erfüllen, so erhält der Benutzer eine kurze Mitteilung.

⁸⁸ Dieses Suchergebnis bezieht sich nicht auf die in Abb. 6.22 dargestellte Suchanfrage.

befindenden Seitennummern. Um die Übersichtlichkeit weiter zu erhöhen, wird mittels JavaScript und CSS dafür gesorgt, dass beim Überfahren der Ergebniszeilen mit dem Mauszeiger die jeweils aktuelle Zeile komplett farbig hervorgehoben wird.

6.3.4 Administrationsbereich

Der administrative Bereich von ISDOS ist über den Hauptmenüpunkt „Verwaltung“ zugänglich. Dieser enthält wiederum drei Menüpunkte, über welche die in der Datenbank gespeicherten Kundendaten, die Benutzerrollen mit ihren Rechten sowie die Verteilerlisten verwaltet werden können. Hierfür stellt jeder der drei Menüpunkte vier Untermenüpunkte bereit, mit denen jeweils ein Formular zum Anzeigen, Anlegen, Ändern oder Löschen aufgerufen werden kann (s. Abb. 5.6). Jeder dieser vier Untermenüpunkte ist mit einem bestimmten Recht verbunden und kann wie bereits beschrieben nur dann ausgewählt werden, wenn ein Benutzer das benötigte Recht dazu hat. Verfügt der Benutzer nicht wenigstens über das Recht einen der vier Untermenüpunkte aufzurufen, ist der gesamte übergeordnete Menüpunkt deaktiviert. Sollten auf dieser Ebene alle drei Menüpunkte nicht auswählbar sein, so ist auch der Hauptmenüpunkt „Verwaltung“ nicht zugänglich.⁸⁹

Da sich die Formulare zum Anzeigen, Anlegen, Ändern und Löschen gleichen, werden in diesem Abschnitt exemplarisch nur die Formulare zum Anlegen von Kundendaten, zum Ändern von Benutzerrollen sowie zum Löschen von Verteilerlisten vorgestellt.

Wie der Abb. 6.24 zu entnehmen ist, befindet sich im oberen Teil des Formulars zum Anlegen von Kundendaten ein Listenfeld, das alle aktuell in der Datenbank gespeicherten Kunden mit den dazugehörigen Kundenwerken und eindeutigen Kundennummern enthält. Unterhalb des Listenfeldes kann der Benutzer zwischen zwei Möglichkeiten des Kundendatenanlegens wählen. Je nachdem welcher LinkButton hier angeklickt wird, erscheint im unteren Bereich des Formulars entweder ein Eingabe- oder ein Auswahlfeld zum Festlegen des Kunden. In diesem Beispiel wurde der LinkButton mit der Beschriftung „Kunden auswählen und Werk anlegen“ gedrückt, sodass ein Benutzer anschließend einen bereits in der Datenbank gespeicherten Kunden auswählen und das Kundenwerk sowie die eindeutige Kundennummer eingeben kann. Durch Anklicken der Schaltfläche „Speichern“ können die Eingaben des Formulars in der Datenbank gespeichert werden. Bevor das Formular jedoch an den Webserver übertragen werden kann, wird mittels JavaScript geprüft, ob die beiden Eingabefelder nicht leer sind.

⁸⁹ Dasselbe Prinzip wird auch im Hauptmenüpunkt „Bestellung“ angewandt.

Bosch Intranet AE Intranet SzP Intranet **BOSCH**

Start Bestellung > Verwaltung > Kontakt Hilfe Abmelden

Verwaltung > Kunden und Werke > Anlegen

Kunden und Werke

Aktueller Stand:

AMG Affalterbach 761138
AMG Maranello 127483
Audi Ingolstadt 249374
Audi Neckarsulm 134865
Audi Wolfsburg 231243
Bentley Crewe 247355
BMW Dingolfing 126475

Neuen Kunden und Werk anlegen
 Kunden auswählen und Werk anlegen

Kunden auswählen und Werk anlegen:

Kunde: Audi

Kundenwerk: *

Kundennr.: 2316

Die Kundennummer muss eine 6-stellige Zahl sein!

Speichern

Seitenanfang Letzte Änderung: 26.10.2006

Abb. 6.24: Formular zum Anlegen von Kundendaten

Darüber hinaus findet eine Überprüfung der eingegebenen Kundennummer statt um sicherzustellen, dass diese eine 6-stellige Zahl ist. Können die Kundendaten nach erfolgreicher clientseitiger Gültigkeitsprüfung in der Datenbank gespeichert werden, wird der neue Eintrag in das Listenfeld aufgenommen und selektiert. Andernfalls wird der Benutzer zur Fehlerbehandlungsseite weitergeleitet, wo er eine Fehlermeldung angezeigt bekommt.

Wie in der Abb. 6.25 zu sehen ist, befindet sich auch im Formular zum Ändern von Benutzerrollen ein Listenfeld, in dem alle aktuell in der Datenbank vorgehaltenen Benutzerrollen angezeigt werden. Wird das Formular aufgerufen, sieht ein Benutzer zunächst ausschließlich das Listenfeld und die daneben befindliche Schaltfläche „Auswählen“. Erst wenn der Benutzer eine Benutzerrolle im Listenfeld selektiert und die Schaltfläche betätigt, wird der untere Teil des Formulars geladen und der ausgewählte Eintrag kann bearbeitet werden. Das Listenfeld bleibt weiterhin sichtbar, sodass ein Benutzer jederzeit eine andere Benutzerrolle auswählen kann. Im unteren Bereich des Formulars hat ein Benutzer nun die Möglichkeit den Namen der Benutzerrolle sowie das Kennwort zu ändern. Unterhalb der Eingabefelder können schließlich die Berechtigungen für die Benutzerrolle festgelegt werden. Im Screenshot sind die Rechte der Benutzerrolle „Sales“ zu sehen. Wäre stattdessen die Benutzerrolle „Admin“ ausgewählt worden, würden alle Rechte als selektiert angezeigt werden.

Bosch Intranet AE Intranet SzP Intranet BOSCH

Start Bestellung Verwaltung Kontakt Hilfe Abmelden

Verwaltung > Benutzerrollen > Ändern

Benutzerrollen

Aktueller Stand:

Ihre Änderungen:

Benutzerrolle:

Altes Kennwort:

Neues Kennwort: Kennwort bestätigen:

Bitte bestätigen Sie das neue Kennwort.

Berechtigungen:

Bestellung:

Anzeigen Anlegen Ändern Stornieren Löschen

Kunde und Werk:

Anzeigen Anlegen Ändern Löschen

Verteilerliste:

Anzeigen Anlegen Ändern Löschen

Benutzerrolle:

Anzeigen Anlegen Ändern Löschen

Seitenanfang Letzte Änderung: 26.10.2006

Abb. 6.25: Formular zum Ändern von Benutzerrollen

Durch Anklicken der Schaltfläche „Speichern“ können auch hier die getätigten Änderungen in der Datenbank gespeichert werden. An dieser Stelle wird wiederum durch JavaScript-Funktionen sichergestellt, dass das Eingabefeld für den Namen der Benutzerrolle nicht leer ist und beim Ändern des Kennworts alle drei notwendigen Eingabefelder einen Eintrag enthalten, wenn mindestens eines dieser Felder ausgefüllt worden ist. Vergisst bspw. ein Benutzer das neue Kennwort zu bestätigen, wird er durch eine Fehlermeldung darauf hingewiesen. Darüber hinaus werden die Eingabe für das neue Kennwort und die Kennwortbestätigung miteinander verglichen, wobei eine Fehlermeldung erscheint, wenn diese nicht übereinstimmen. Können die Änderungen erfolgreich in der Datenbank gespeichert werden, wird der untere Teil des Formulars geschlossen und ausschließlich der obere Teil dargestellt, wobei die zuvor geänderte Benutzerrolle selektiert ist. Andernfalls bekommt der Benutzer auf der Fehlerbehandlungsseite eine

Fehlermeldung angezeigt. Mit der Schaltfläche „Schließen“ kann ein Benutzer die getätigten Änderungen verwerfen und den unteren Teil des Formulars schließen.

Abbildung 6.26 zeigt das Formular zum Löschen von Verteilerlisten. Auch hier sieht ein Benutzer nach dem Aufrufen des Formulars zunächst nur ein Listenfeld, welches die aktuellen Verteilerlisten enthält, und darunter eine Schaltfläche zum Auswählen einer solchen. Die im Listenfeld angezeigten Verteilerlisten werden dabei mit ihrer Benennung und den dazugehörenden E-Mail-Adressen dargestellt. Damit das Listenfeld von der Breite her nicht zu groß wird, werden maximal die ersten beiden E-Mail-Adressen angezeigt. Enthält eine Verteilerliste mehr als zwei E-Mail-Adressen, so befinden sich am Ende des jeweiligen Eintrages drei Punkte. Wenn ein Benutzer eine Verteilerliste im Listenfeld selektiert und die Schaltfläche „Auswählen“ drückt, wird der untere Teil des Formulars geladen und die ausgewählte Verteilerliste mit Hilfe von zwei nicht editierbaren Eingabefeldern angezeigt. Soll der ausgewählte Eintrag doch nicht gelöscht werden, so kann über die Schaltfläche „Schließen“ der Vorgang beendet und der untere Teil des Formulars wieder geschlossen werden. Durch Drücken der Schaltfläche „Löschen“ wird die ausgewählte Verteilerliste endgültig gelöscht. Dies ist nur dann möglich, wenn keine Referenz mehr auf die zu löschende Verteilerliste innerhalb der Datenbank besteht. Ist eine derartige Referenz vorhanden, wird der Löschvorgang abgebrochen und eine erklärende Fehlermeldung angezeigt.

The screenshot shows a web interface for deleting distribution lists. At the top, there are navigation links for 'Start', 'Bestellung', 'Verwaltung', 'Kontakt', and 'Hilfe', along with a 'Abmelden' button. The current page is 'Verwaltung > Verteilerlisten > Löschen'. The main content area is titled 'Verteilerlisten' and contains a table with the following data:

Aktueller Stand:	
SzP/QMM1/MSB	abteilungsbriefkasten.qmm1@de.bosch.com; abteilungsbriefkasten.msb@de.bosch.com ...
SzP/QMM1/MSH	abteilungsbriefkasten.qmm1@de.bosch.com; abteilungsbriefkasten.msh@de.bosch.com

Below the table is an 'Auswählen' button. A confirmation message asks: 'Wollen Sie den Eintrag wirklich löschen?'. Below this, there are two input fields: 'Benennung: SzP/QMM1/MSB' and 'E-Mail-Adressen: abteilungsbriefkasten.qmm1@de.bosch.com; abteilungsbriefkasten.msb@de.bosch.com; max.mustermann@de.bosch.com'. At the bottom right of the form are 'Löschen' and 'Schließen' buttons. The footer contains 'Seitenanfang' and 'Letzte Änderung: 26.10.2006'.

Abb. 6.26: Formular zum Löschen von Verteilerlisten

7 Bewertung der Lösung

In diesem Kapitel wird das entwickelte webbasierte Informationssystem mit den anfangs in Kapitel 4 definierten Anforderungen verglichen, womit gezeigt werden soll, inwieweit die realisierte Lösung den Vorgaben entspricht.

Grundsätzlich lässt sich feststellen, dass die Webanwendung mit den vorgegebenen **Entwicklungstechnologien** ASP.NET 2.0 und VB 2005 umgesetzt werden konnte. Der Zugriff auf das realisierte Informationssystem erfolgt von einem beliebigen Client-Rechner innerhalb des firmeninternen Netzwerks. Die Benutzer benötigen hierfür lediglich ein JavaScript-fähigen **Webbrowser**, was den Vorteil hat, dass clientseitig keine Installation und Wartung von anderweitigen Programmkomponenten notwendig ist. Auf dem **Webserver**, bei welchem es sich wie gefordert um einen IIS (auf MS Windows Server 2003) handelt, muss das System dagegen einmalig eingerichtet werden. Damit ISDOS in vollem Umfang bedienbar ist, muss JavaScript im Webbrowser auch aktiviert sein. Der Einsatz von JavaScript erschien deshalb notwendig, da nur so die volle Funktionalität einiger ASP.NET-Serversteuerelemente wie etwa die des Menüsteuerelements zur Verfügung steht. Darüber hinaus wird durch die Verwendung von JavaScript, z. B. für die Formatierung und Validation von Eingabefeldinhalten, zusätzlicher Netzverkehr und Zeitaufwand eingespart sowie der Bedienkomfort der Webanwendung erhöht, da die Änderungen bzw. Fehlermeldungen in der Regel sofort nach Verlassen des betreffenden Eingabefeldes sichtbar sind.

Der Schutz von ISDOS vor unberechtigtem Zugriff ist vollständig realisiert. So wird die **Sicherheit** des Websystems zum einen dadurch gewährleistet, dass sich ein jeder Benutzer mit einer Benutzerrolle und dem passenden Kennwort authentifizieren muss. Zum anderen ist durch das implementierte Rollenkonzept sichergestellt, dass ein Benutzer die Funktionalität des Websystems nur in dem Umfang nutzen kann, wie es die mit der Benutzerrolle verbundenen Berechtigungen erlauben.

Die in den fünf Anwendungsfällen beschriebenen Anforderungen sind komplett umgesetzt worden. So ermöglicht das Websystem, dass alle für die Bemusterung benötigten Informationen in einheitlicher Form erfasst und zentral in einer Datenbank verwaltet werden können. Für die **Datenhaltung** kommt entsprechend den Vorgaben eine Oracle 9i Datenbank zum Einsatz. Bei der Umsetzung des Bestellformulars wurde besonders darauf geachtet, dass dieses übersichtlich aufgebaut ist. Hierfür sind thematisch zueinander gehörende Informationen jeweils in einem separaten Bereich zusammengefasst worden. Der Vorteil besteht darin, dass Benutzer für einen bestimmten Arbeitsschritt benötigte Informationen schneller erfassen können, da sie diese nicht unterschiedlichen Stellen innerhalb des Dokuments entnehmen müssen. Zudem werden in dem neuen

Eingabeformular (s. Abb. 6.17 und 6.18) im Gegensatz zu den alten EM-Checklisten (s. Abb. 3.2) keine Informationen mehrfach erfasst sowie für jedes auswählbare Bemusterungsverfahren ein eigener Eingabeblock zur Verfügung gestellt, welcher sich vom Inhalt und Aufbau stark an den Empfehlungen und Forderungen des VDA Bd. 2 bzw. des Referenzhandbuchs PPAP orientiert.

Wie in den Anforderungen definiert, kann für die Erfassung einer EM-Bestellung sowohl ein leeres Formular als auch eine bereits gespeicherte Bestellung als Vorlage verwendet werden. Die für den zweiten Fall erforderliche Suchfunktion, welche auch benötigt wird um eine gespeicherte Bestellung zum Anzeigen, Ändern, Stornieren oder Löschen zu öffnen, wurde ebenfalls erfolgreich realisiert.

Hat ein Benutzer alle für die Bemusterung benötigten Informationen erfasst, kann er diese über eine fest im Websystem hinterlegte Verteilerliste an den Fertigungsstandort übermitteln. Ist die Bestellung dagegen unvollständig, kann er einen anderen Mitarbeiter über die gespeicherte Bestellung in Kenntnis setzen, sodass dieser die Bestellung weiter bearbeiten kann. In beiden Fällen wird eine E-Mail versendet, die einen Verweis enthält, mit dem die Bestellung geöffnet werden kann.

Auch der administrative Bereich des Websystems ist vollständig einsatzbereit. So können Benutzerrollen mit ihren Berechtigungen, Verteilerlisten sowie Kundendaten sowohl angezeigt, angelegt, geändert als auch gelöscht werden.

Mit der Umsetzung einer separaten Fehlerbehandlungsseite wird auch die Anforderung nach **Fehlertoleranz** erfüllt. So werden sowohl Laufzeitfehler als auch anderweitige Fehlersituationen, die bspw. auftreten, wenn der Zugriff auf eine Webseite bzw. Ressource nicht möglich ist (z. B. wegen fehlender Rechte des Benutzers bzw. Ausfall des DB-Servers) oder Benutzereingaben unzulässig sind (z. B. beim Anlegen eines bereits existierenden Kunden), abgefangen und in einer zentralen Fehlerbehandlungsroutine ausgewertet. Letztendlich erhöht dies ebenfalls die Benutzerfreundlichkeit, da einem Benutzer anstelle einer sonst üblichen Standardfehlermeldung eine passende Fehlermeldung präsentiert wird, welche in den meisten Fällen eine aussagekräftige Erläuterung zur Fehlerbehebung enthält. Zudem werden in ISDOS wie bereits erwähnt fehlende sowie ungültige Benutzereingaben (z. B. fehlerhafte E-Mail-Adresse, falscher Datentyp), welche zu Laufzeitfehlern führen können, in der Regel schon clientseitig mittels JavaScript abgefangen und behandelt.

Zur Gewährleistung einer einfachen **Bedienbarkeit** und **Erlernbarkeit** von ISDOS wurde vieles realisiert. So ist z. B. die Benutzeroberfläche des Websystems klar in einen Kopf-, Navigations-, Inhalts- und Fußbereich unterteilt, was dem Benutzer hilft, sich

schnell zurechtzufinden. Der Navigationsbereich stellt darüber hinaus ein übersichtliches Pull-Down-Menü zur Verfügung, welches sich den Wahlmöglichkeiten eines Benutzers anpasst, indem nur die Aktionen auswählbar sind, die er auch ausführen darf. Weiterhin sind die Ein- und Ausgabemasken des Inhaltsbereichs klar strukturiert und verfügen jeweils über einen Schriftzug, der einem Benutzer Auskunft darüber gibt, wo er sich gerade befindet und was er dort tun kann. Die vollständige Umsetzung der **Mehrsprachigkeit** ist ein weiterer Punkt, der den Bedienkomfort von ISDOS erhöht, da so die Benutzeroberfläche in der jeweiligen Landessprache eines Benutzers angezeigt werden kann.⁹⁰ Grundsätzlich wurde bei der Gestaltung der Benutzeroberfläche auf ein ansprechendes und einheitliches Layout sowie auf die Einhaltung der RB-internen Vorgaben für Farbgebung, Schriftarten etc. geachtet. Der Vorteil der Realisierung des Layouts mit Hilfe einer ASP.NET-Masterpage und zwei CSS-Dateien besteht darin, dass mögliche Anpassungen an eben jenem Layout an zentraler Stelle durchgeführt werden können.

Darüber hinaus gewährleistet die strikte Trennung von Layout und Programmierlogik sowie die ausführliche Dokumentation der Entwicklung und des Programmcodes eine leichte **Erweiterbarkeit** und Änderbarkeit des bestehenden Websystems, so dass z. B. jederzeit flexibel weitere Funktionen und Sprachen hinzugefügt werden können. Soll ISDOS bspw. um eine komplett neue Funktionalität erweitert werden, ist es sinnvoll ein neues mit der Masterpage verbundenes Webform zu implementieren, von welchem selbstverständlich auf die realisierten Hilfsklassen zugegriffen werden kann. Letztendlich muss die Masterpage um einen Eintrag ergänzt werden, damit die neue Funktion auch über die Menüleiste auswählbar ist. Dagegen können für eine neue Sprache lokalisierte Zeichenfolgen jederzeit der Webanwendung, ohne diese anzupassen, hinzugefügt werden, indem die neuen XML-basierten Ressourcendateien einfach auf den Webserver kopiert und die notwendigen Datenbanktabellen um weitere Einträge ergänzt werden.

Damit eine **Übertragbarkeit** der Webanwendung von einer Computerplattform auf eine andere ohne hohen Änderungsaufwand möglich ist, wurden weder die Datenbankverbindungszeichenfolge noch die Einstellungen zum Versenden von E-Mails fest im Programmcode, sondern zentral in einer XML-basierten Webkonfigurationsdatei hinterlegt. Zudem wurde die Verwendung von absoluten Pfaden z. B. zu URL-Ressourcen und Dateien im Code vermieden. Dies hat den Vorteil, dass bei der Portierung von ISDOS auf eine neue Serverplattform lediglich die E-Mail-Einstellungen und ggf. die Verbindungszeichenfolge zur Datenbank in der Konfigurationsdatei angepasst werden müssen. Grundsätzlich gilt es hierbei zu beachten, dass mit ASP.NET 2.0 entwickelte Webanwendungen aktuell nur auf Windows-Plattformen lauffähig sind. Da jedoch mit

⁹⁰ Wie in den Anforderungen festgelegt, werden aktuell die Sprachen Deutsch und Englisch unterstützt.

ASP.NET 1.1 realisierte Anwendungen z. B. auch auf einem Apache-Webserver betrieben werden können, ist anzunehmen, dass dies auch bald für ASP.NET 2.0 Webanwendungen gilt.

Die aus der Anforderungsliste bisher noch nicht aufgeführten zwei Punkte **Datenzugriff** und **Hilfefunktion** sind noch nicht vollständig umgesetzt. So wurde bei der Realisierung der Datenbank zwar eine Sicht angelegt, mit welcher der Generator der QMM1 sowie die Erstmuster-Datenbank der MSB/H auf einen bestimmten Bereich der Datenbank zugreifen können, jedoch ist in den benannten Programmen die Funktionalität, welche den Datenimport einer ausgewählten EM-Bestellung ermöglichen soll, noch nicht umgesetzt. Ebenso ist die Hilfefunktion bisher nur teilweise realisiert. So sind zwar wie im Systementwurf vorgestellt im Inhaltsbereich der Benutzeroberflächen die Symbole mit dem Fragezeichen zu finden, nur wird ein Benutzer beim Anklicken bisher auf eine Webseite weitergeleitet, welche ihn darüber informiert, dass diese Funktionalität noch nicht zur Verfügung steht. Die Vervollständigung der Hilfefunktion soll in einem weiteren Entwicklungsprojekt erfolgen. Das Ziel dieses Projekts besteht darin eine Online-Hilfe zu implementieren, welche beim Anklicken eines „?“-Symbols in einem neuen Browser-Fenster geladen wird. Diese soll nicht nur wie im Systementwurf vorgestellt eine kurze kontextbezogene Hilfe, sondern auch einen eigenen Navigationsbereich mit einem Menü in Form einer Baumstruktur enthalten, mit dem ein Benutzer durch die verschiedenen Hilfethemen navigieren kann. Zudem existiert die Idee in der Online-Hilfe eine eigenständige Suchfunktion zur Verfügung zu stellen.

Da sich das webbasierte Informationssystem zum Zeitpunkt der Abgabe der Diplomarbeit noch nicht im Realbetrieb befindet, können keine endgültigen Aussagen zur **Zuverlässigkeit** gemacht werden. Bisher hat sich ISDOS jedoch in diversen Tests als sehr stabil und zuverlässig erwiesen.

8 Zusammenfassung und Ausblick

Im Rahmen dieser Diplomarbeit wurde unter der Bezeichnung ISDOS (Initial Sample Documentation Order System) ein webbasiertes Informationssystem entwickelt, welches zur Erfassung und Verwaltung aller für die Bemusterung benötigten Informationen dienen soll.

Zu Beginn der vorliegenden Arbeit sind dazu die bestehenden Strukturen und Abläufe der Bemusterung in der RB Elektronik GmbH analysiert worden. Der dokumentierte Ist-Zustand und dadurch aufgezeigte Schwachstellen bildeten schließlich die Planungsgrundlage für die Erarbeitung des fachlichen Soll-Konzepts. Hierbei wurden in Abstimmung mit Auftraggeber und künftigen Nutzern die Systemanforderungen festgelegt und der Soll-Zustand für den Bemusterungsablauf vorgestellt. Nach der Konzeption und dem Entwurf folgte die praktische Realisierung des Websystems. Im vorherigen Kapitel konnte letztendlich gezeigt werden, dass die entwickelte Lösung nahezu alle gestellten Anforderungen erfüllt, womit der Einführung von ISDOS nichts im Wege steht.

Mit der Nutzung des Websystems sind eindeutige Verbesserungen gegenüber der alten Lösung hervorzuheben. So können mit ISDOS alle relevanten Informationen zur Bemusterung in einheitlicher Form über ein klar strukturiertes Bestellformular erfasst werden. Dieses orientiert sich vom Inhalt und Aufbau nicht nur an den Forderungen und Empfehlungen der Systembenutzer, sondern auch an denen des VDA Bd. 2 und des Referenzhandbuchs PPAP. Zudem werden die erfassten Informationen einer EM-Bestellung zentral in einer Datenbank gespeichert und stehen anschließend über einen Webbrowser jedem berechtigten Systembenutzer online zur Verfügung. Eine weitere Verbesserung besteht darin, dass ISDOS eine Suchfunktion zur Verfügung stellt, mit welcher schnell und auf einfache Weise auf gespeicherte EM-Bestellungen zugegriffen werden kann. Mit ISDOS werden letztlich Zeitverzögerungen und potentielle Fehlerquellen während des Bemusterungsablaufs minimiert sowie Medienbrüche und Mehrfacherfassungen vermieden.

Nach der Beendigung der Testphase wird ISDOS vorerst in der RB Elektronik GmbH am Standort Salzgitter eingeführt. Ist das Websystem installiert, müssen in einem nächsten Schritt die Benutzer bezüglich der Nutzung des Systems geschult werden. Hierfür ist eine multilinguale Bedienungseinleitung zu erstellen, welche auf der Startseite von ISDOS bereitgestellt wird.

Da bei der Entwicklung des Websystems von Anfang an Wert auf eine leichte Übertragbarkeit und Anpassbarkeit gelegt wurde, weiterhin sowohl die Ein- und Ausgabeformulare als auch die Datenbank so allgemein wie möglich realisiert wurden, besteht

die Möglichkeit das System nicht nur am Standort Salzgitter, sondern falls dieses sich bewährt auch an anderen RB-internen Fertigungsstandorten zur Erfassung und Verwaltung der für die Bemusterung benötigten Informationen einzusetzen.

Auch wenn das realisierte Websystem im beschriebenen Umfang bereits praktisch einsetzbar ist, können sinnvolle Weiterentwicklungen, welche u. a. durch die ausführliche Dokumentation des Programmcodes und nicht zuletzt durch die vorliegende Arbeit gut vorbereitet sein dürften, den Nutzen der vorgestellten Lösung weiter erhöhen. Eine Ausbaumöglichkeit bestünde darin, den Benutzern interessante Statistiken in Form von grafischen Aufbereitungen ausgewählter Daten zur Verfügung zu stellen. Mit diesen könnte bspw. dargestellt werden, wie viele EM-Bestellungen in einem Jahr, verteilt auf die einzelnen Monate, am Fertigungsstandort eingegangen sind bzw. wie viele Bestellungen von einem bestimmten Vertriebsmitarbeiter, aufgeschlüsselt nach demselben Muster, erfasst wurden. Zudem könnte der Bedienkomfort von ISDOS weiter erhöht werden, indem zusätzliche Funktionalität mittels JavaScript hinzugefügt wird. So können die Benutzer z. B. dahingehend unterstützt werden, dass beim Auswählen der Vorlagestufe zum Bemusterungsverfahren alle von der gültigen Norm geforderten Prüfungen und Dokumente automatisch vorselektiert werden.

Zusammenfassend lässt sich sagen, dass der produktive Einsatz des entwickelten Websystems den Bemusterungsablauf verbessert, was letztendlich allen beteiligten Abteilungen und Stellen viel Mühe, Zeit und Kosten einspart.

Literaturverzeichnis

- AIAG – Automotive Industry Action Group (Ed.) (2006): Production Part Approval Process (PPAP). 4th ed., Southfield.
- Alpar, P.; Grob, H. L.; Weimann, P.; Winter, R. (2005): Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen. 4. Aufl., Wiesbaden.
- Alvestrand, H. (1995): Tags for the Identification of Languages.
<http://www.faqs.org/rfcs/rfc1766.html>. (2006-09-16).
- Böhm, R.; Fuchs, E. (2002): System-Entwicklung in der Wirtschaftsinformatik. 5. Aufl., Zürich.
- Burghardt, M. (2002): Einführung in Projektmanagement: Definition, Planung, Kontrolle, Abschluss. 4. Aufl., Erlangen.
- Bustamante, M. L. (2005): ASP.NET 2.0-Lokalisierungsfeatures: Ein neuer Ansatz für die Lokalisierung von Webanwendungen.
<http://www.microsoft.com/germany/msdn/library/net/aspnet/ASPNET20LokalisierungfeaturesEinNeuerAnsatzFuerDieLokalisierungVonWebanwendungen.aspx>. (2006-06-09).
- DGQ – Deutsche Gesellschaft für Qualität e. V. (Hrsg.) (2000): Organisatorische Gestaltungsmöglichkeiten für Qualitätsmanagementsysteme. 3. Aufl., Berlin/Wien/Zürich.
- DIN – Deutsches Institut für Normung e. V. (Hrsg.) (1996): DIN EN ISO 9241-10:1996-07, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 10: Grundsätze der Dialoggestaltung. Berlin/Wien/Zürich.
- DIN – Deutsches Institut für Normung e. V. (Hrsg.) (1999): DIN EN ISO 9241-11:1999-01, Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten – Teil 11: Anforderungen an die Gebrauchstauglichkeit – Leitsätze. Berlin/Wien/Zürich.
- DIN – Deutsches Institut für Normung e. V. (Hrsg.) (2000): DIN EN ISO 9000:2000, Qualitätsmanagementsysteme – Grundlagen und Begriffe. Berlin/Wien/Zürich.
- DIN – Deutsches Institut für Normung e. V. (Hrsg.) (2002): Technische Spezifikation ISO/TS 16949: Qualitätsmanagementsysteme: Besondere Anforderungen bei Anwendung von ISO 9001:2000 für die Serien- und Ersatzteil-Produktion in der Automobilindustrie. 2. Aufl., Berlin/Wien/Zürich.
- DQS GmbH (Hrsg.) (2006): QS-9000 Quality System Requirements: Produktinformation. <http://www.dqs.de/servlet/ContentServer/DQS/Page/DQSUnterrubrik2/c-Page/cid-1025005511833.html>. (2006-11-04).
- Dumke, R.; Lothar, M.; Wille, C.; Zbrog, F. (2003): Web Engineering. München.
- EDS Deutschland GmbH (Hrsg.) (2006): Material Daten System.
http://www.mdsystem.com/html/de/home_de.htm. (2006-11-16).
- Fowler, M. (2004): UML konzentriert: Eine kompakte Einführung in die Standard-Objektmodellierungssprache. 3. Aufl., München.

- Geiger, W.; Kotte, W. (2005): Handbuch Qualität : Grundlagen und Elemente des Qualitätsmanagements : Systeme - Perspektiven. 4. Aufl., Wiesbaden.
- Häuslein, A. (2004): Systemanalyse: Grundlagen, Techniken, Notierungen. Berlin/Offenbach.
- Heuer, A.; Saake, G. (2000): Datenbanken: Konzepte und Sprachen. 2. Aufl., Bonn.
- Ishida, R.; Miller, S. K. (2005): Localization vs. Internationalization.
<http://www.w3.org/International/questions/qa-i18n.de.html>. (2006-06-09).
- Kamiske, G. F.; Brauer J.-P. (2006): Qualitätsmanagement von A bis Z : Erläuterungen moderner Begriffe des Qualitätsmanagements. 5. Aufl., München/Wien.
- Kappel, G.; Pröll, S.; Reich, S.; Retschitzegger, W. (2004): Web Engineering: Systematische Entwicklung von Web-Anwendungen. Heidelberg.
- Kruschinski, V. (1999): Layoutgestaltung grafischer Benutzungsoberflächen: Generierung aus OOA-Modellen. Heidelberg/Berlin.
- Lubkowitz, M. (2005): Webseiten programmieren und gestalten: HTML, CSS, XML, PHP, JavaScript, Perl, MySQL, SVG, RSS, Weblogs. 2. Aufl., Bonn.
- Mertens, P. (Hrsg.) (2001): Lexikon der Wirtschaftsinformatik. 4. Aufl., Berlin/Heidelberg/New York.
- Merz, M. (2002): E-Commerce und E-Business: Marktmodelle, Anwendungen und Technologien. 2. Aufl., Heidelberg.
- Microsoft Deutschland GmbH (Hrsg.) (2005a): .NET Framework-Entwicklerhandbuch: Empfehlungen für die Entwicklung weltweit einsatzfähiger Anwendungen.
<http://msdn2.microsoft.com/de-de/library/w7x1y988.aspx>. (2006-06-23).
- Microsoft Deutschland GmbH (Hrsg.) (2005b): ASP.NET: Übersicht über die ASP.NET-Seitenklasse. <http://msdn2.microsoft.com/de-de/library/ms178138.aspx>. (2006-10-02).
- Oracle Corporation (Ed.) (2006): Oracle SQL Developer.
http://www.oracle.com/technology/products/database/sql_developer/index.html. (2006-11-16).
- Pattison, T. (2006): Ressourcen und Lokalisierung in ASP.NET 2.0.
<http://www.microsoft.com/germany/msdn/library/net/aspnet/RessourcenUndLokalisierungInASPNET20.aspx?mfr=true>. (2007-01-02).
- Pfeifer, T. (2001): Qualitätsmanagement : Strategien, Methoden, Techniken. 3. Aufl., München/Wien.
- Rätzmann, M. (2004): Software-Testing & Internationalisierung: Agiles Qualitätsmanagement, Die Praxis der Internationalisierung, Methodische Testfallermittlung. 2. Aufl., Bonn.
- Robert Bosch GmbH (Hrsg.) (2006): Bosch heute. Stuttgart.
- Scheer, A.-W. (2002): ARIS - Vom Geschäftsprozess zum Anwendungssystem. 4. Aufl., Berlin/Heidelberg/New York.
- Schwichtenberg, H. (2006): Microsoft ASP.NET 2.0 mit Visual Basic 2005. Unterschleißheim.
- Seidlmeier, H. (2002): Prozessmodellierung mit ARIS: Eine beispielorientierte Einführung für Studium und Praxis. Braunschweig/Wiesbaden.

- VDA – Verband der Automobilindustrie e. V. (Hrsg.) (2004): VDA-Schriftenreihe Qualitätsmanagement in der Automobilindustrie, Band 2: Sicherung der Qualität von Lieferungen: Lieferantenauswahl, Qualitätssicherungsvereinbarungen, Produktionsprozess- und Produktfreigabe, Qualitätsleistung in der Serie, Deklaration von Inhaltsstoffen. 4. Aufl., Frankfurt am Main.
- Wenz, C.; Hauser, T.; Kordwig, A.; Trennhaus, C. (2002): ASP.NET: Leistungsfähige Webapplikationen mit Visual Basic.NET und C#. München.

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, 17. Januar 2007

.....

(Unterschrift)